JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

3/2026

vol. 105

Dual Wide-band Microstrip Antenna for Wireless Communication Systems

P. Okołot, M. Wnuk, and K. Szczepankiewicz 1

Machine Learning-based Automatic Modulation Classification for 5G-Advanced and 6G Waveforms: Robust Identification Under Realistic Channel Impairments

A. Horch, M. Besseghier, S. Ghouali, and A. Louni 8

Communication-efficient Embedded FFT Processing for Acoustic Telemetry in LPWAN-based Beehive Monitoring Systems

S. Górecki 22

Network Threat Detection in IaaS Environments Based on Flow Log Data

H. Wójcik, M. Roszkowski, and A. Mycek 33

Adaptive Entropy Prediction-based Lossless Compression for Efficient Data Transmission in Terrestrial and Underwater IoT Systems

A. Sisodia and S. Vishnoi 42

A Spatio-temporal Residual Attention Network with SNR-adaptive Weighting for MIMO-OFDM Channel Estimation

O. El-azizi, R. Fateh, and S. Said 52

FEC Hybrid Method Based on Reed–Solomon Codes for Transmitting Video Signal Over Limited Bandwidth and High Loss Rate Channels

L.V. Kaliuzhnyi, V.B. Rusalovskiy, A.D. Ishchenko, and V.M. Didenko 65

Energy-delay Aware Data Collection in Mobile Sink-based Wireless Sensor Networks with Optimized Visiting Points and Two-level Data Aggregation

N. Benaouda, B. Moussaoui, and O. Senouci 74

(*Contents continued on back cover*)

Editor-in-Chief

Adrian Kliks, Poznan University of Technology, Poland

Editorial Advisory Board

Hovik Baghdasaryan, National Polytechnic University of Armenia, Armenia
Naveen Chilamkurti, LaTrobe University, Australia
Luis M. Correia, Instituto Superior Técnico, Universidade de Lisboa, Portugal
Pedro Crespo Bofill, Universidad de Navarra TECNUN, Spain
Luca De Nardis, DIET Department, University of Rome La Sapienza, Italy
Nikolaos Dimitriou, NCSR "Demokritos" Athens, Greece
Ciprian Dobre, Politechnic University of Bucharest, Romania
Piotr Gawrysiak, Warsaw University of Technology, Poland
Filip Idzikowski, Poznan University of Technology, Poland
Andrzej Jajszczyk, AGH University of Science and Technology, Poland
Erich Leitgeb, Graz University of Technology, Austria
Albert Levi, Sabanci University, Türkiye
Marian Marciniak, National Institute of Telecommunications, Poland
George Mastorakis, Hellenic Mediterranean University: Agios Nikolaos, Crete
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Takumi Miyoshi, Shibaura Institute of Technology, Japan
Jordi Mongay Batalla, Warsaw University of Technology, Poland
Klaus Mößner, Technische Universität Chemnitz, Germany
Imran Muhammad, Federation University: Brisbane, Qld, Australia
Mjumo Mzyece, Northwestern College in Orange City, Iowa, USA
Daniel Negru, University of Bordeaux, France
Jordi Perez-Romero, Universitat Politècnica de Catalunya, Spain
Michał Pióro, Warsaw University of Technology, Poland
Konstantinos Psannis, University of Macedonia, Greece
Karupiya Balasundaram Rajesh, Chikkanna Government Arts College, India
Salvatore Signorello, Universidade Nova de Lisboa: Lisbon, Lisbon, Portugal
Dejan Vukobratovic, University of Novi Sad, Serbia
Adam Wolisz, Technische Universität Berlin, Germany

Editorial Team

Content Editor: **Robert Magdziak**
Managing Editor: **Ewa Kapuściarek**

eISSN 1899-8852

© Copyright by National Institute of Telecommunications, Poland 2026

Dual Wide-band Microstrip Antenna for Wireless Communication Systems

Patrycja Okołot, Marian Wnuk, and Konrad Szczepankiewicz

Military University of Technology, Warsaw, Poland

https://doi.org/10.26636/jtit.2026.3.2635

Submitted: 28 May 2026, Accepted: 9 Jun. 2026

Abstract — This article presents the design of a compact dual wide-band microstrip antenna. It also describes the simulation of its operation and provides the results of electrical parameter and radiation characteristics measurements. The designed antenna is intended to operate in two ranges. The first covers frequencies from 3.8 to 8 GHz and all the U-NII antenna ranges (U-NII-1, U-NI-2A, U-NII-2B, U-NII-2C, U-NII-3, U-NII-4, U-NII-5, U-NII-6, U-NII-7, U-NII-8) defined in the IEEE-802.11a standard (5.150 to 7.125 GHz). The other covers frequencies from 11.34 to 20.64 GHz and the Ku band (from 11.7 to 12.7 GHz). The performance of the antenna, developed using CST Microwave Studio and Matlab software, was verified through simulation. Then, a prototype was manufactured and was subjected to measurements in an anechoic chamber. Results of the simulation and the measurement campaign were compared against six other dual-band antenna designs available in the literature. Comparison of the parameters of the proposed design and characteristics of other antennas with a similar profile, selected from the literature, shows that the operating bandwidth of the proposed solution with respect to the S_{11} parameter is the largest, reaching 88.13% for the lower operating band and 59.93% for the upper operating band.

Keywords — dual band, FDTD method, microstrip antenna, wide band

1. Introduction

In the modern technology landscape, where communication is no longer an option and has long become a key functionality of devices, antennas – including those of the microstrip variety – play a particularly important role. Their compact size, low weight, and ability to be integrated with other electronic devices make them useful in a wide range of applications, including 5G networks [1], [2].

Decisions regarding the allocation of the 4400 – 4800 MHz, 7125 – 7250 MHz, 7750 – 8400 MHz, and 14.8 – 15.35 GHz bands for terrestrial IMT systems are expected to be made soon. The sub-6 GHz operating frequency band includes the n79 (4400 – 5000 MHz) range. Furthermore, Wi-Fi standards typically cover the 5 GHz band (5150 – 5830 MHz). The latest WLAN communication standard divides the 6 GHz band (5925 – 7125 MHz) into Wi-Fi 6E/7 ranges. Therefore, a dual-band antenna covering both the Sub-6 GHz and Wi-Fi 6E/7 frequency ranges may be a promising solution for future communication devices.

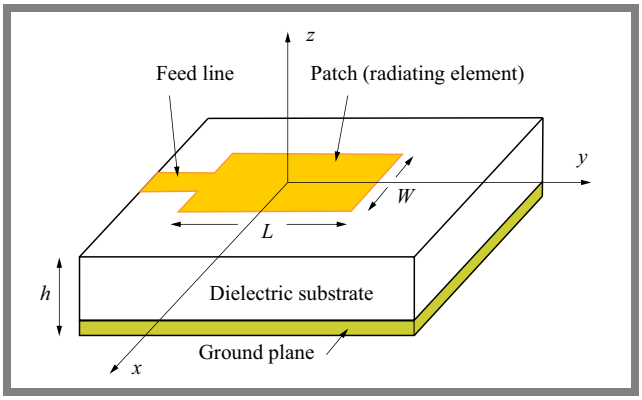


Fig. 1. Microstrip antenna.

Recently, several studies have been published on broadband antennas for 5G sub-6 GHz networks with an operating bandwidth of 11%, e.g. [3]–[8]. To the best of the authors' knowledge, no dual-band antenna designs with a significantly wider operating band have been reported in the literature.

This paper presents a developed model of a dual-band microstrip antenna, implemented in CST Microwave Studio software. The main assumption is that the new solution will be operating in the band covering frequencies from 3.8 to 8 GHz, and a second band covering frequencies from 11.34 to 20.64 GHz, thus fully covering the Ku band (from 11.7 to 12.7 GHz). Another important assumption for the antenna model is its compact size with dimensions no larger than 20 × 25 mm, which is important for its potential application in mobile devices.

2. Design and Modeling of a Dual-band Antenna

Microstrip designs are currently one of the most innovative fields in antenna technology. Microstrip antennas offer numerous interesting features, such as small size, low volume, and flat design making them easy to integrate with end-user applications. Low production costs and high repeatability of specifications, combined with a wide operating frequency range, ensure numerous application opportunities. Low weight allows such antennas to be used on fast flying objects, without exerting any negative impact on their aerodynamic properties.

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

However, the dielectric substrate used favors the excitation of surface waves which, propagating along the dielectric plane, interfere with the normal operation of the antenna [9]. Other disadvantages of microstrip antennas include a relatively narrow bandwidth and limited power capabilities. This causes mutual coupling, which changes the field distributions on aperture antennas and the current distributions in linear antennas. This phenomenon, in turn, changes the spatial characteristics of antenna radiation and input impedance.

A microstrip antenna consists of a metal radiator (patch) and a ground plane, with both elements separated by a dielectric layer (Fig. 1). The relative permittivity of substrate ε_r varies from 2.2 to 12. Specification of the substrate also controls radiation efficiency, bandwidth, and antenna size. In most cases, microstrip antennas have a rectangular shape, with their crucial parameters determined by patch width W, length L, and thickness of the dielectric layer h characterized by relative dielectric permittivity ε_r [10], [11].

During the modeling process, in the first approximation, the dimensions of the antenna, i.e. length and width, were determined using the transmission line model, defined as follows:

$$L = \frac{c}{2f_r\sqrt{\varepsilon_{re}}} - 2\Delta L\,, \tag{1}$$

$$W = \frac{c}{2f_0\sqrt{\frac{\varepsilon_r+1}{2}}}\,, \tag{2}$$

where ε_{re} is the effective dielectric constant such as:

$$\varepsilon_{re} = \frac{\varepsilon_r+1}{2} + \frac{\varepsilon_r-1}{2}\left(1+12\,\frac{h}{W}\right)^{-1/2}. \tag{3}$$

The width of the radiator determines operating bandwidth and input impedance. An increase in the width of the patch improves radiation efficiency. On the other hand, the antenna's resonant frequency depends on the length of the radiator. After calculations performed in Matlab using Eqs. (1), (2), the width-to-length ratio was determined to be $W/L \approx 1.31$. This value results from Eqs. (1) and (2) for the assumed substrate parameters and operating frequency range.

Radiator extension value ΔL, depending on the effective dielectric constant, laminate height, radiator length and width, is another parameter of the antenna. ΔL is related to the scattering fields occurring at the ends of the radiator. It is determined by the following formula:

$$\Delta L = 0.412\,h\frac{(\varepsilon_{re}+0.300)\,(W/h+0.264)}{(\varepsilon_{re}-0.258)\,(W/h+0.813)}\,. \tag{4}$$

The antenna's input impedance Z, i.e. 50 Ω, depends on the distance from the antenna's edge to the radiator's edge. The value increases as this distance decreases. The radiator's input resistance was then determined as:

$$R_{in} = \frac{1}{G_{10}}\,, \tag{5}$$

where G_{10} is conductance representing radiation, dielectric, and patch/shield losses.

To speed up the calculation of antenna parameters according to Eqs. (1) – (4), a program was created in the Matlab environment, which resulted in determining approximate dimensions of the antenna: W = 13 mm and L = 10 mm.

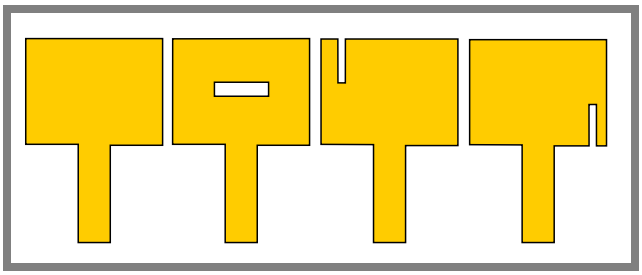


Fig. 2. Steps of iterative antenna design process.

The antenna operates in TM10 and TM01 wave modes [12], [13]. The location of the feed line is determined by the desired wave mode. For TM01 and higher modes, the feed line should be placed on the W side, and if the antenna is to operate in TE10 and higher modes, on the L side. In the case of waves in TM01 and higher modes, the W edges radiate, and in the case of TE10 and higher modes, the L edges radiate. Comparison of both wave modes shows that when the element operates in TM01 and higher wave modes, the quality factor of the system is lower, resulting in a wider operating band.

The method of moments is used most frequently for analyzing microstrip antennas, as it allows to model the current distribution. It is based on the concept of reactions between the base segments and the test functions. The Green function is used for integral equations of the electric field, satisfying the boundary conditions for a metal radiator. Integral equations are discretized into linear equations using the method of moments to obtain a matrix equation [14]. Solving the matrix equations yields the current distribution on the radiator. This distribution, along with the Green function, allows to determine the antenna's radiation pattern in both the near and far fields.

The antenna's substrate may be made of a homogeneous or mixed dielectric ($\varepsilon_r \neq 1$) materials. For a non-homogeneous dielectric, Green's function can only be determined in the spectral domain. For the x axis of the current J_x at the air-dielectric interface and while $z = h$, the electric field components in the spectral domain are [15] as follows.

$$\overrightarrow{E}_x(k_x,k_y,h) = \frac{-j}{\omega\,\varepsilon_0}\left[\frac{k_x^2 k_1 k_2\sin(k_1h)}{\beta^2 TM} + \frac{k_y^2 k_0^2\sin(k_1h)}{\beta^2 TE}\right]J_x\,, \tag{6}$$

$$\overrightarrow{E}_y(k_x,k_y,h) = \frac{-j}{\omega\,\varepsilon_0}\left[\frac{k_x k_y k_1 k_2\sin(k_1h)}{\beta^2 TM} + \frac{k_x k_y k_0^2\sin(k_1h)}{\beta^2 TE}\right]J_x\,, \tag{7}$$

$$\overrightarrow{E}_z(k_x,k_y,h) = \frac{-j}{\omega\,\varepsilon_0}\left(\frac{k_1 k_x\sin(k_1h)}{TM}\right)J_x\,, \tag{8}$$

$$TM = \varepsilon_r k_2\cos(k_1h) + jk_1\sin(k_1h)\,, \tag{9}$$

$$TE = k_1\cos(k_1h) + jk_2\sin(k_1h)\,, \tag{10}$$

$$k_1^2 = k_z^2 = \varepsilon_r k_0^2 - \beta^2 \quad \text{for } 0 < z < h \text{ (layer 1)} \;\; \mathrm{Im}(k_1) < 0\,, \tag{11}$$

$$k_2^2 = k_z^2 = k_0^2 - \beta^2 \quad \text{for } z > h \text{ (layer 2)} \;\; \mathrm{Im}(k_2) < 0\,, \tag{12}$$

$$\beta^2 = k_x^2 + k_y^2\,, \tag{13}$$

where: k_x, k_y, k_1, k_2, k_0, are components of the electromagnetic field, layer 1 is free space and layer 2 is dielectric.

According to [16], [17], a rectangular antenna has a relatively narrow operating bandwidth, not exceeding 10%. A viable way to expand the operating bandwidth is to use a multilayer dielectric structure. However, such solutions complicate the antenna's design and increase its dimensions. Another solution is to introduce slots in the plane of the radiating element. In this case, their dimensions and position are chosen to disrupt the first two modes of the slit's side edge so that their resonant frequencies approach each other, creating a wide operating bandwidth.

Various slot shapes can be found in [18]–[22], including: comb-shaped, double-bend, cross-shaped, U-shaped. The parameters that influence the bandwidth of a microstrip antenna are the slot length and width, as well as its position in the antenna plane. It should be noted that the slot width should be small relative to its length, because the higher resonant mode is sensitive to changes in the horizontal slot length, while the lower resonant mode strongly depends on the slot circumference [23]–[27]. Using Eqs. (6)–(12), both the higher and lower modes, i.e., the slot length and width, were selected to obtain the required antenna bandwidth.

Changes in the slot length and width did not significantly affect the antenna's output impedance. Specific mathematical formulas, even empirical ones, which would determine the exact geometry of the structure and slot position to ensure the antenna achieves a predetermined bandwidth, are not yet available [28], [29].

The tool used to design the microstrip antenna is CST Microwave Studio. This advanced electromagnetic field simulation software allows to design, analyze, and optimize systems and their components in a three-dimensional environment. The program utilizes the finite-difference time-domain (FDTD) method, which allows for accurate broadband simulations. This method is suitable for analyzing both stationary and transient electromagnetic wave behaviors in complex structures. Table 1 summarizes CST output of the designed structure's dimensions and parameters.

The designed antenna has three slots, as shown in Fig. 2. The position and dimensions of these slots were determined through iterative evaluation. The microstrip antenna's radiator is made of 35 µm thick copper, while the dielectric substrate is Rogers FR4 – a material characterized by good electrical properties, a dielectric constant of $\varepsilon_r = 4.3$, a dielectric loss angle of tg $\delta = 10^{-3}$, and a thickness of dielectric layer $b = 1.6$ mm. The antenna is fed by a 3-millimeter microstrip line. This width results from the antenna being designed for standard 50 Ω impedance.

Tab. 1. Simulation parameters used to analyze the proposed system.

Description	Name	Value [mm]
Distance of the top slot from the left edge	W1	1.7
Slot width at the top edge	W2	0.6
Distance of the right edge of the top slot from the right edge	W3	10.7
Distance of the center slot from the left edge	W4	4.0
Length of the center slot	W5	5.0
Distance of the feed line from the left edge	W6	5.0
Width of the feed line	W7	3.0
Distance of the bottom slot from the feed line	W8	3.5
Width of the slot at the bottom edge	W9	0.6
Distance of the left edge of the slot from the right edge	W10	0.9
Ground plane width	W11	20.0
Length of the feed line	L1	9.5
Antenna length	L2	10.0
Length of the top slot	L3	3.9
Distance of the top edge of the center slot from the top edge	L4	4.5
Distance of the bottom edge of the center slot from the bottom edge	L5	4.5
Length of the bottom slot	L6	3.9
Ground plane height	L7	9.0

The basis for selecting the type of slots as well as their dimensions and position was the determination of the s parameter matrix (describing scattering between the device's ports). S_{11} indicates the amount of power reflected from the antenna and is referred to as the reflection coefficient. We assume that the antenna operates correctly for $S_{11} < -10$ dB.

To check the impact of the slots on the antenna's operating bandwidth, a series of simulations was performed using CST. The results for four scenarios: antenna without slots (1), with a horizontal slot (2), with a slot at the top edge (3), and with a slot at the bottom edge (4), are presented in Fig. 2. For each scenario, S_{11} parameters were obtained for the following operating bands (Fig. 3):

1) 16.25 to 20.85 GHz,
2) 3.75 to 8.05 GHz and 10.2 to 21 GHz,

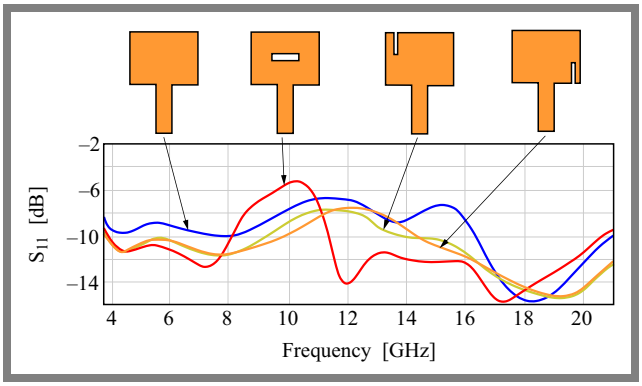


Fig. 3. S_{11} versus frequency for the considered scenarios.

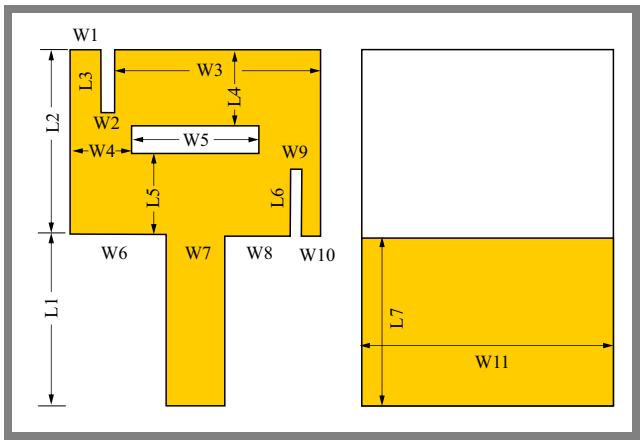


Fig. 4. Front and back side view of optimized antenna.

3) 3.75 to 8.05 GHz and 10.2 to 21 GHz,

4) 3.75 to 8.15 GHz and 11.2 to 21 GHz.

Figure 3 shows that none of the developed cases met the required operating bandwidth. When analyzing the obtained results, one may conclude that the slot located in the lower right corner has the greatest impact on the operating bandwidth. This may be the case as this slot contributes most to the excitation of higher TM modes. Therefore, in the next step, an antenna design including all the slots shown in Fig. 2 to form a comb-like structure was analyzed.

The designed slots allowed for the excitation of higher-order modes according to Eqs. (6)–(13). This iterative design process results from the fact that at this stage, the only method described in the literature for determining the location of slots is the trial-and-error method, which requires extensive design experience. Such an antenna design is presented in Fig. 4, while detailed dimensions of the structure are provided in Tab. 1.

The next step in the analysis was to determine the current distribution. For frequency ranges where the condition $S_{11} < -10$ dB is met, the electric field strength values are roughly similar – see the upper row in Fig. 5. In a scenario where only one frequency band meets the requirements of $S_{11} < -10$ dB, we can conclude that for the selected frequency of 12 GHz, there is a multi-fold increase in the electric field strength value, as shown in bottom row of Fig. 5.

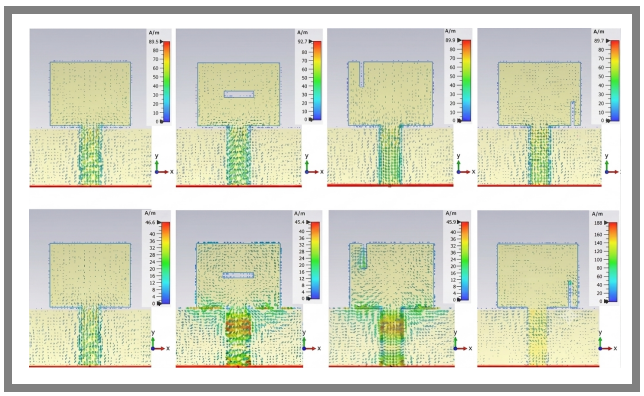


Fig. 5. Current distribution for the frequency of 3.9 GHz (upper row) and 12 GHz (bottom row).

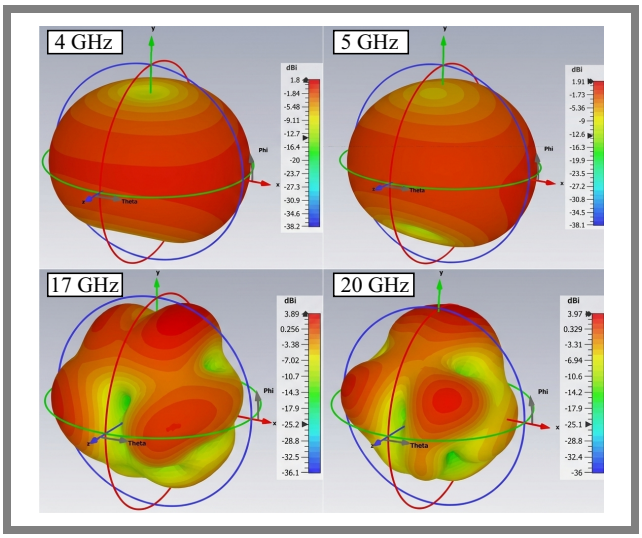


Fig. 6. 3D plot of antenna energy gain.

The radiation pattern describes the spatial distribution of the intensity of the electromagnetic field generated by the antenna. The energy gain, on the other hand, is a measure of the antenna's ability to direct RF energy in each direction. Results of the simulation focusing on the antenna's properties in 3D are presented in Fig. 6.

One may notice there that in the bottom frequency range of 3.9 to 8.11 GHz, the energy gain is significantly lower than in the case of the upper working band of 11.34 to 20.64 GHz. At 4 GHz, the energy gain is 1.8 dBi, and for 20 GHz, it reaches 3.97 dBi. Therefore, it can be concluded that at higher frequencies, the antenna does not provide the expected omnidirectionality.

3. Fabrication and Measurements

To achieve dual-band operation with wide bandwidth, the final antenna utilizes all slots from the design iterations presented in Fig. 2. The final concept was simulated using CST to verify its assumptions and then, after corrections to the shape, size, and radiator size, a prototype of the antenna was manufactured (Fig. 7).

The measurements were conducted under far-field conditions in an anechoic chamber. To eliminate potential interference from the connector, the pins were covered with an RF ab-

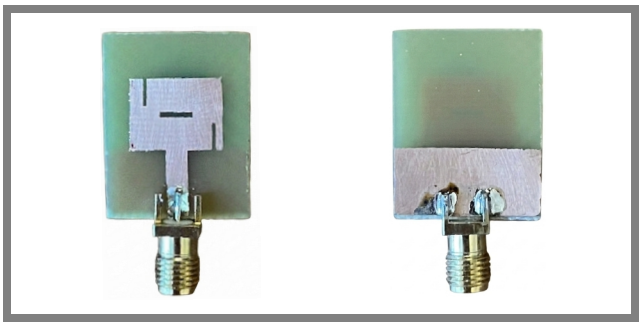


Fig. 7. Fabricated antenna prototype: front and bottom view (right).

Tab. 2. Comparison of packet delivery ratios based on the transmitting node.

Ref.	Footprint area λ_0^2		Profile λ_0		Bandwidth	
[3]	1.1 × 1.1		0.06		25%	
[4]	1.1 × 1.1		0.06		25%	
[5]	0.75 × 0.75		0.045		21.3%	
[6]	0.88 × 0.78		0.11		24.8%	
[7]	1.17 × 0.5		0.039		10%	
[8]	0.58 × 0.39		0.048		14%	
Proposed	Lower band	Upper band	Lower band	Upper band	Lower band	Upper band
	0.396 × 0.496	1.831 × 2.1289	0.0317	0.146	88.13%	59.93%

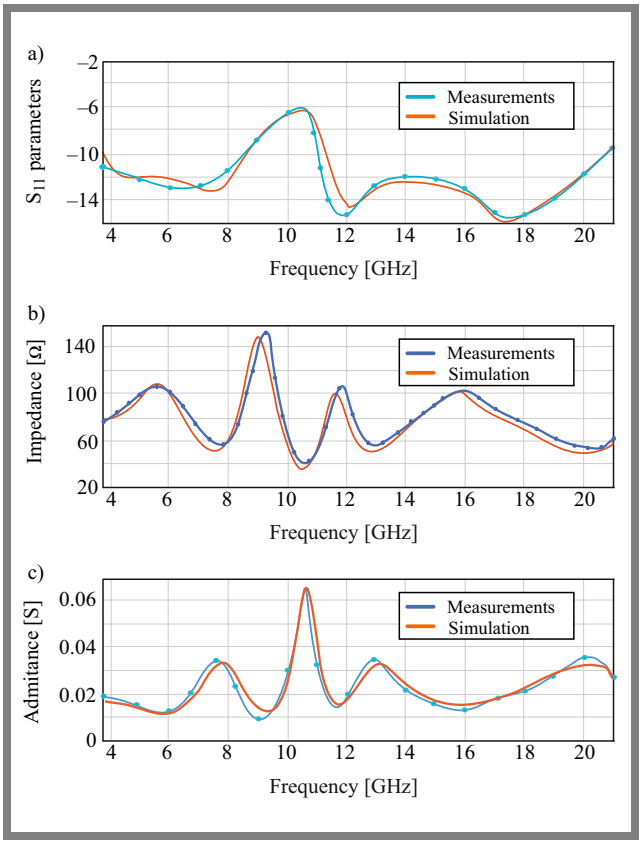


Fig. 8. S_{11} for the proposed antenna a), input impedance b), and admittance c).

sorber. the measurements were performed in accordance with ANSI/IEEE Std 149:1979.

Reflection coefficient S_{11} is defined as the voltage ratio of the reflected wave to the wave that encounters the antenna. A high value of this coefficient indicates that a significant amount of energy is not absorbed by the antenna but is reflected instead, which is unfavorable and can result in lower antenna efficiency. As previously mentioned, for antennas, we assume that the S_{11} parameter should be below –10 dB.

The lowest S_{11} reflection coefficient for the lower band is –12.07 dB at 6.5 GHz and –15.2 dB at 17.23 GHz for the upper band, according to Fig. 8a. Such values indicate good matching of the antenna to the transmission line, minimizing power losses while maintaining acceptable system performance. This matching is crucial to ensuring optimal antenna performance in its operating environment.

The input impedance is a key parameter of any antenna, as it determines the quality of matching with the RF power source, which directly affects performance, bandwidth, and radiation efficiency. As can be seen in Fig. 8b, the input impedance for the lower band at its cutoff frequencies is 57 and 40 Ω, respectively. The expected 50 Ω value was obtained at 6.72 GHz. For the upper frequency band, the input impedance for the cutoff frequencies reached amounts to 62 and 32 Ω. The 50 Ω threshold was obtained for frequencies of 12.14, 14.01, and 17.93 GHz.

The antenna input admittance Y_{11} is the reciprocal of input impedance Z_{11}. This is an important parameter in analyzing any antenna, as it describes how easily current flows into the antenna from an external source. Admittance is a complex quantity and consists of a real part and an imaginary part.

Admittance for a microstrip antenna should approach a conductance value of 0.02 S and a susception value of 0, with an assumed input impedance of 50 Ω, ensuring optimal operating conditions and antenna performance (Fig. 8c).

Directivity describes how an antenna radiates or receives RF energy in different directions. It can be expressed using a radiation pattern, which is a graphical representation of the field intensity as a function of azimuth and elevation angles.

The obtained azimuthal radiation patterns, both theoretical and measured, show that the proposed antenna radiates energy in a directionally dependent manner, meaning it is not omnidirectional. The antenna is designed to have a radiation pattern that is close to omnidirectional. These patterns represent a normalized power density distribution. Figure 9 represents the antenna's radiation patterns for selected frequencies. One may notice the measurement curves are identical to the calculated patterns.

4. Comparison with Other Designs

Table 2 presents a comparison of the parameters of the proposed design with other selected wide-band antennas from the literature in terms of impedance matching, profile (thickness),

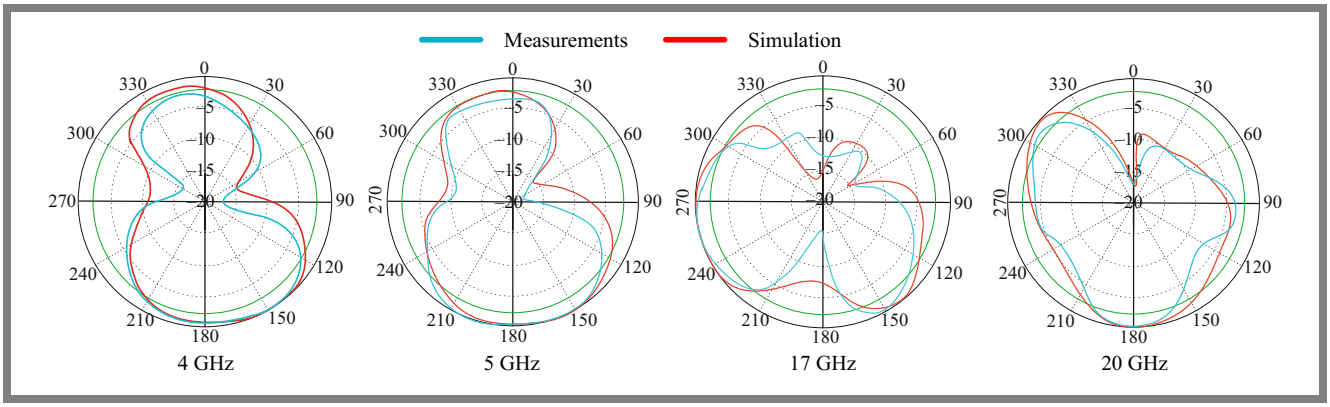


Fig. 9. Normalized directional characteristics in the azimuth plane.

and bandwidth. The operating bandwidth of the proposed antenna with respect to the S_{11} parameter is the highest compared to [3]–[8]. The calculated and measured values of the S_{11} parameter also show a high degree of convergence.

Based on the comparison of the electrical parameters of the proposed antenna with other designs, it was shown that the proposed broadband antenna exhibits comparable parameters in terms of matching impedance in all cases, especially in the case of stringent matching conditions [30].

5. Conclusions

The presented antenna is designed to operate in two frequency bands. Simulation results and measurements performed in an anechoic chamber (input impedance, reflection coefficient, and standing wave ratio) confirm its suitability for intended applications in radio communication systems. The wide operating bandwidth was achieved by using slots. Unfortunately, their size and location were determined through successive trials as part of an iterative design process. In further work, the authors plan to develop algorithms for automatic slot location planning according to the specified parameters.

References

[1] A.K. Arya, S.J. Kim, and S. Kim, "A Dual-band Antenna for LTE-R and 5G Lower Frequency Operations", *Progress In Electromagnetics Research Letters*, vol. 88, pp. 113–119, 2020 (https://doi.org/10.2528/PIERL19081502).

[2] I. Rodriguez *et al.*, "An Experimental Framework for 5G Wireless System Integration into Industry 4.0 Applications", *Energies*, vol. 14, art. no. 4444, 2021 (https://doi.org/10.3390/en14154444).

[3] W.E.I. Liu, Z.N. Chen, and X. Qing, "Metamaterial-based Low-profile Broadband Mushroom Antenna", *IEEE Transactions on Antennas and Propagation*, vol. 62, pp. 1165–1172, 2014 (https://doi.org/10.1109/TAP.2013.2293788).

[4] W.E.I. Liu, Z.N. Chen, and X. Qing, "Broadband Low-profile L-probe Fed Metasurface Wave Resonances", *IEEE Transactions on Antennas and Propagation*, vol. 68, pp. 1348–1355, 2020 (https://doi.org/10.1109/TAP.2019.2955629).

[5] W.Y. Sun and Y. Li, "Gain Stabilization Method for Wideband Slot-coupled Microstrip Antenna", *IEEE Transactions on Antennas and Propagation*, vol. 69, pp. 8932–8936, 2021 (https://doi.org/10.1109/TAP.2021.3097441).

[6] W. Liu, L. Zhu, W. Choi, and X. Zhang, "A Low-profile Differential-fed Patch Antenna with Bandwidth Enhancement and Sidelobe Reduction under Operation of TM and TE Modes", *IEEE Transactions on Antennas and Propagation*, vol. 66, pp. 4854–4859, 2018 (https://doi.org/10.1109/TAP.2018.2851393).

[7] W. Roh *et al.*, "Millimeter-wave Beamforming as an Enabling Technology for 5G Cellular Communications: Theoretical Feasibility and Prototype Results", *IEEE Communications Magazine*, vol. 52, pp. 106–113, 2014 (https://doi.org/10.1109/MCOM.2014.6736750).

[8] D. Wang, K.B. Ng, C.H. Chan, and H.Wong, "A Novel Wideband Differentially-fed Higher-order Mode Millimeter-wave Patch Antenna", *IEEE Transactions on Antennas and Propagation*, vol. 63, pp. 466–473, 2015 (https://doi.org/10.1109/TAP.2014.2378263).

[9] A. Gohar and G. Nencioni, "The Role of 5G Technologies in a Smart City: The Case for Intelligent Transportation System", *Sustainability*, vol. 13, art. no. 5188, 2021 (https://doi.org/10.3390/su13095188).

[10] A. Kakkar, Nirdosh, and S.A. Sah, "Multiband Circular Patch Microstrip Antenna for K and Ka Applications", *Intelligent Communication, Control and Devices, Advances in Intelligent Systems and Computing*, vol. 624, 2018 (https://doi.org/10.1007/978-981-10-5903-2_150).

[11] H. Ali *et al.*, "An Eight Element Dual Band Antenna for Future 5G Smartphones", *Electronics*, vol. 10, art. no. 3022, 2021 (https://doi.org/10.3390/electronics10233022).

[12] W. Liu *et al.*, "A Low-profile Differential-fed Patch Antenna with Bandwidth Enhancement and Sidelobe Reduction under Operation of TM and TM Modes", *IEEE Transactions on Antennas and Propagation*, vol. 66, pp. 4854–4859, 2018 (https://doi.org/10.1109/TAP.2018.2851393).

[13] D. Chaturvedi, B. Pramodini, and T. Lanka, "Compact MIMO Antenna with Extended Bandwidth Enabled by Parasitic Patch Structure", *Journal of Electromagnetic Waves and Applications*, vol. 39, pp. 1368–1379, 2025 (https://doi.org/10.1080/09205071.2025.2505051).

[14] M. Wnuk, "Two Methods to Analyze Microstrip Antennas for Wi-Fi Bandwidth", *Archives of Electrical Engineering*, vol. 70, pp. 705–719, 2021 (https://doi.org/10.24425/aee.2021.137583).

[15] Q.S. Liu, S. Sun, and W.C. Chew, "A Potential-based Integral Equation Method for Low-frequency Electromagnetic Problems", *IEEE Transactions on Antennas and Propagation*, vol. 66, pp. 1413–1426, 2018 (https://doi.org/10.1109/TAP.2018.2794388).

[16] S. Radavaram and M. Pour, "Wideband Radiation Reconfigurable Microstrip Patch Antenna Loaded with Two Inverted U-slots", *IEEE Transactions on Antennas and Propagation*, vol. 67, pp. 1501–1505, 2019 (https://doi.org/10.1109/TAP.2018.2885433).

[17] M. Wnuk, "Multilayer Dielectric Periodic Antenna Structure in a Cascade View", *Applied Sciences*, vol. 12, art. no. 4185, 2022 (https://doi.org/10.3390/app12094185).

[18] J. Park *et al.*, "Concept of Integrating 4G LTE and Millimeter-Wave 5G Antennas within Zero-bezel Cellular Devices", *2020 IEEE In-*

ternational Symposium on Antennas and Propagation and North American Radio Science Meeting, Montreal, Canada, 2020 (https://doi.org/10.1109/IEEECONF35879.2020.9330129).

[19] Y. Zou and J. Pan, "Broadband and High-gain Antenna Based on Novel Frequency Selective Surfaces for 5G Application", *4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, China, 2019 (https://doi.org/10.1109/IAEAC47372.2019.8997989).

[20] J. Colaco and R. Lohani, "Design and Implementation of Microstrip Patch Antenna for 5G Applications", 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, pp. 682–685, 2020 (https://doi.org/10.1109/ICCES48766.2020.9137921).

[21] K.A. Fante and M.T. Gemeda, "Broadband Microstrip Patch Antenna at 28 GHz for 5G Wireless Applications", *International Journal of Electrical and Computer Engineering*, vol. 11, pp. 2238–2244, 2021 (https://doi.org/10.11591/ijece.v11i3.pp2238-2244).

[22] S. Punith, S.K. Praveenkumar, A.A. Jugale, and M.R. Ahmed, "A Novel Multiband Microstrip Patch Antenna for 5G Communications", *Procedia Computer Science*, vol. 171, pp. 2080–2086, 2020 (https://doi.org/10.1016/j.procs.2020.04.224).

[23] Y. Ghazaoui *et al.*, "Millimeter Wave Antenna with Enhanced Bandwidth for 5G Wireless Application", *Journal of Instrumentation*, vol. 15, art. no. T01003, 2020 (https://doi.org/10.1088/1748-0221/15/01/t01003).

[24] M. Alam, R.I. Tomal, A.A.M. Faudzi, and N.A.T. Yuso, "ANN-Enabled Gain Prediction and Optimization in Dual-band SIW Antenna Designs for 5G Networks", *Journal of Telecommunication and Information Technology*, vol. 103, pp. 69–78, 2026.

[25] M.M. Kamal *et al.*, "A Novel Hook-shaped Antenna Operating at 28 GHz for Future 5G mmWave Applications", *Electronics*, vol. 10, art. no. 673, 2021 (https://doi.org/10.3390/electronics10060673).

[26] M.T. Gemeda, K.A. Fante, H.L. Goshu, and A.L. Goshu, "Design and Analysis of a 28 GHz Microstrip Patch Antenna for 5G Communication Systems", *International Research Journal of Engineering and Technology*, vol. 8, pp. 881–886, 2021.

[27] M. Hussain *et al.*, "Design and Characterization of Compact Broadband Antenna and its MIMO Configuration for 28 GHz 5G Applications", *Electronics*, vol. 11, art. no. 523, 2022 (https://doi.org/10.3390/electronics11040523).

[28] A.S.A. Gaid, M.A.M. Ali, A. Saif, and W.A.M. Mohammed, "Design and Analysis of a Low Profile, High Gain Rectangular Microstrip Patch Antenna for 28 GHz Applications", *Cogent Engineering*, vol. 11, art. no. 2322827, 2024 (https://doi.org/10.1080/23311916.2024.2322827).

[29] C.A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed., John Wiley & Sons, Newark, 2016 (ISBN: 9781118642061).

[30] Y. Zhou and Y. Zheng, "A High-gain and Dual-band Compact Metasurface Antenna for Wi-Fi/WLAN Applications", *Materials*, vol. 18, art. no. 2538, 2025 (https://doi.org/10.3390/ma18112538).

Patrycja Okołot, M.Sc.
Department of Electronics
https://orcid.org/0009-0009-9536-0036
E-mail: patrycjaokolot@wp.pl
Military University of Technology, Warsaw, Poland
https://www.wojsko-polskie.pl/wat

Marian Wnuk, Professor
Department of Electronics
https://orcid.org/0000-0003-4576-4023
E-mail: marian.wnuk@wat.edu.pl
Military University of Technology, Warsaw, Poland
https://www.wojsko-polskie.pl/wat

Konrad Szczepankiewicz, M.Sc.
Department of Electronics
https://orcid.org/0000-0003-3292-8113
E-mail: konrad.szczepankiewicz@wat.edu.pl
Military University of Technology, Warsaw, Poland
https://www.wojsko-polskie.pl/wat

Machine Learning-based Automatic Modulation Classification for 5G-Advanced and 6G Waveforms: Robust Identification Under Realistic Channel Impairments

Abdelkader Horch, Mokhtar Besseghier, Samir Ghouali, and Abderrahmane Louni

University of Mustapha Stambouli, Mascara, Algeria

https://doi.org/10.26636/jtit.2026.3.2595

Submitted: 20 Apr. 2026, Reviewed: 12 Jun. 2026

Abstract — Automatic modulation classification (AMC) for 5G-Advanced and 6G networks must blindly identify waveforms from received signals under realistic channel impairments, enabling cognitive radio dynamic spectrum access and interference avoidance. No prior work has simultaneously applied machine learning to classify all eight leading waveforms (UFMC, GFDM, FBMC, NOMA, OFDM-IM, OTFS, ODDM, and AFDM) under realistic channel impairments, nor quantified the minimum feature set for resource-constrained deployment. We present a framework that (i) extracts a 38-dimensional feature vector that includes three novel channel-aware characteristics (amplitude fading variance, phase discontinuity, and frequency drift); (ii) benchmarks nine machine learning classifiers, including an FC-MLP deep learning baseline and five feature selection methods, on 201 600 signals across twelve channel conditions (nine custom plus three 3GPP TDL profiles) and seven SNR levels, with leakage-free feature selection; and (iii) identifies a compact 10-feature subset validated with Bonferroni-corrected McNemar tests and Wilson confidence intervals. FC-MLP achieves 99.09% accuracy; ensemble-boost (99.04%) and random forest (99.02%) are statistically equivalent. The 10-feature random forest reaches a score of 98.90% within 0.12 pp of the full feature baseline at a cost that is 74% lower and with a 0.071 ms inference per block. The five-fold cross-validation confirms stability (98.54%, Wilson 95% CI: 98.49%, 98.59%). Per channel accuracy ranges from 98.87% (Rayleigh) to 99.98% (AWGN/Rician); 3GPP TDL-A/B/C profiles confirm transferability to 5G NR. The three channel-aware features yield up to 3.1% gain under double-selective fading and an average overall improvement.

Keywords — 5G-Advanced/6G, automatic modulation classification, cognitive radio, compact feature selection, dynamic spectrum access, machine learning, realistic channel impairments

1. Introduction

Automatic modulation classification (AMC) enables a receiver to blindly determine the waveform type of an observed signal without prior knowledge of the modulation scheme, channel conditions, or signal parameters [?], [?]. In the context of 5G-Advanced and 6G networks, robust AMC is indispensable for a wide range of applications, including access to the dynamic spectrum of cognitive radio (CR), interference avoidance, cooperative co-existence and spectrum policy enforcement [?]–[?]. Despite its importance, AMC for the full set of next-generation waveform candidates under realistic channel propagation conditions remains a largely open problem.

The 5G-Advanced and 6G standardization processes have produced a diverse set of waveform candidates, each targeting specific deployment requirements: UFMC and GFDM for flexible subband allocation, FBMC for high spectral efficiency without cyclic prefix, NOMA for massive connectivity via power domain multiplexing; OFDM-IM for energy-efficient index modulation; and OTFS, ODDM, and AFDM for robust high-mobility communications via delay-Doppler or chirp-based signal representations [?]–[?]. Precise classification of these waveforms is present, using machine learning on blind feature observations, and is a prerequisite for all subsequent spectrum management decisions. Recent AMC studies have extensively addressed legacy modulation schemes such as PSK, QAM, and FSK, but have not simultaneously tackled all eight next-generation waveform candidates under compound realistic channel impairments.

1.1. Motivation and Challenges

The deployment of ML-based AMC for next-generation waveforms faces three distinct difficulties: structural similarity among the waveform candidates (which challenges any classifier), severe distortion from realistic propagation channels (which corrupts the most informative features), and strict computational constraints when AMC must operate in real time, which limits model complexity. These challenges are especially acute in cognitive radio and spectrum moni-

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

toring scenarios, where the receiver operates independently from the transmitter.

Structural similarity. Several next-generation waveforms share overlapping time-frequency characteristics, making blind ML classification particularly difficult. OTFS, ODDM, and AFDM are all designed for Doppler resilience through related delay-Doppler or chirp-based representations, making their spectral signatures difficult to distinguish. FBMC, GFDM, and UFMC all exploit subband filtering, mainly differing in prototype filter design. Previous AMC methods trained under clean channel conditions suffer 5 – 15% accuracy degradation when evaluated under realistic impairments [**?**], [**?**], as structurally discriminative characteristics become corrupted by channel effects, i.e., precisely the conditions any real AMC system must handle.

Realistic channel impairments. A CR receiver cannot control or predict the channel it observes. The waveform identification module must therefore be robust enough to face the following:

- Multipath fading: Rayleigh and Rician fading introduce time-varying amplitudes and phase distortions that obscure waveform-specific envelope statistics.
- Doppler effects: high mobility CR scenarios (30 – 200 km/h) induce frequency dispersion and intercarrier interference, particularly harmful for subcarrier-structured waveforms.
- Frequency-selective channels: delay spread exceeding the symbol period causes inter-symbol interference that disrupts cyclic prefix and periodicity features.
- Synchronization errors: a CR receiver operating without cooperation from the transmitter cannot assume perfect carrier or timing synchronization; residual CFO and timing misalignment distort the phase and cyclo-stationary structures of all waveforms.
- Combined impairments: simultaneous time-frequency double-selective channels – common in vehicular and UAV CR deployments – present the most challenging scenario.

Selection of compact features for real-time AMC. A key open question in ML-based AMC is: what is the minimum feature set that preserves near-maximum accuracy under a rigorous, leakage-free selection protocol? This work attempts to answer this question for any resource-constrained deployment, including CR platforms such as IoT gateways, UAV-mounted spectrum sensors, and battery-powered secondary nodes, with all of them imposing strict limits on resources and latency. A waveform classifier that requires hundreds of features or seconds of inference latency is not suitable for real-time operation. No prior work has rigorously quantified the minimum feature set needed for next-generation waveform AMC under realistic channels, nor validated that such a compact model can operate within real-time latency budgets.

The above considerations lead to the following research questions:

1) Which machine learning algorithm achieves the best accuracy-complexity trade-off for AMC of eight 5G-Advanced/6G waveforms under realistic channel conditions, satisfying real-time latency requirements?
2) What is the minimum feature subset that preserves near-maximum AMC accuracy under a leakage-free selection protocol? Does it generalize to ML classifier architectures?
3) Do channel-aware features explicitly designed to capture propagation impairments provide statistically significant AMC improvements, as confirmed by McNemar tests?

1.2. Research Contribution

This paper proposes a CR-oriented 38-feature blind identification framework with mathematical definitions for all features, operating in a fully blind mode, as required by CR receivers with no transmitter cooperation.

We introduce three new channels-aware features (F36 – F38): amplitude fading variance, phase discontinuity, and frequency drift, explicitly designed to characterize the uncontrolled propagation environment a CR receiver observes. Ablation studies quantify their contribution: removing all three degrades identification accuracy by 1.5% on average and by up to 3.1% under double selective fading.

We benchmark nine classifiers, including an FC-MLP deep learning baseline and five feature-selection methods on 201 600 signals across twelve channel conditions and seven SNR levels. FC-MLP achieves the highest accuracy at 99.09%; random forest provides the best accuracy-per-feature efficiency for CR deployment. McNemar tests (corrected with Bonferroni) confirm the statistical significance of all reported differences.

The work extends evaluation to 3GPP standard channels. Three TDL profiles (TDL-A, TDL-B, TDL-C) of 3GPP TR 38.901 [**?**] are added along with nine custom channel conditions, enabling direct comparability with 5G NR deployment scenarios that CR devices will encounter. All feature importance scoring is performed strictly within the training fold; the test set plays no role in feature ranking, ensuring that the reported 10-feature accuracies are genuine out-of-sample estimates suitable for real CR deployment decisions.

The results demonstrate a 74% computational drop by reducing from 38 to 10 features with an accuracy loss of only 0.12% (McNemar $p > 0.05$), achieving 0.071 ms per segment inference and meeting the real-time latency requirements of CR dynamic spectrum access. The compact model is transferred to all evaluated classifiers. We validate robustness through parametric analysis covering SNR (5 to +25 dB), Doppler (0 – 120 km/h), CFO (0 – 5 kHz) and timing offset (0 – 15 samples), i.e., the full envelope of impairments a CR receiver must tolerate, with per waveform confusion analysis identifying the most challenging identification pairs.

The paper is organized as follows. Section **??** describes the system model and the 12-channel CR evaluation setup. Section **??** details the blind feature extraction framework. Sec-

tion **??** presents classification results including the FCMLP baseline test and the McNemar test. Section **??** analyzes the importance of leak-free features. Section **??** evaluates the reduction for CR deployment. Section **??** reports on robustness analysis. Section **??** discusses the implications for the deployment of CR and Section **??** concludes the study.

2. System Model and Dataset Generation

2.1. Signal Model

We model a CR receiver that passively observes a received signal block and is required to identify the waveform without any cooperation from the transmitter. Consider a transmitted signal $s[n]$ corresponding to one of the eight 5G-Advanced/6G waveform candidates, sampled at frequency f_s. The complex baseband signal at the CR receiver after channel propagation is:

$$r[n] = \sum_{\ell=0}^{L-1} h_\ell[n]\, s[n-\ell] + w[n]\,, \tag{1}$$

where $h_\ell[n]$ is the time-varying complex gain of the ℓ-th multipath component and $w[n] \sim \mathcal{CN}(0, \sigma_w^2)$ is complex AWGN with

$$\sigma_w^2 = P_s/\text{SNR}\,,$$
$$P_s = \mathbb{E}[|s[n]|^2]\,.$$

All twelve channel configurations are specific instantiations of Eq. (**??**).

2.2. Channel Models

AWGN: $h_0[n] = 1$ and all other paths zero. Baseline line-of-sight scenario.

Rayleigh fading: Non-line-of-sight single-path propagation with Doppler-correlated complex gain:

$$h[n] = \frac{1}{\sqrt{2}}\big(g_I[n] + j g_Q[n]\big)\,, \tag{2}$$

where $g_I[n]$ and $g_Q[n]$ are correlated Gaussian processes whose power spectral density follows the Clarke/Jakes model

$$S(f) = [\pi f_d \sqrt{1-(f/f_d)^2}]^{-1} \text{for} |f| \leqslant f_d\,,$$

with temporal autocorrelation

$$R_h[m] = J_0(2\pi f_d\, m T_s)\,,$$

where $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind and $T_s = 1/f_s$.

The 16-oscillator Jakes simulator [**?**] is used for implementation. For the static Rayleigh channel $f_d \to 0$ yields independent identically distributed realizations.

Rician fading: Line-of-sight with K factor of 10 dB incorporating the LOS Doppler shift:

$$h[n] = \sqrt{\frac{K}{K+1}}\, e^{j2\pi f_{\text{LOS}} n T_s} + \sqrt{\frac{1}{K+1}}\, h_R[n]\,, \tag{3}$$

where $f_{\text{LOS}} = f_d \cos\theta$ is the LOS Doppler frequency ($\theta = 0^\circ$ for worst-case maximum Doppler) and $h_R[n]$ is a scattered component following the Rayleigh model.

Frequency offset: Post-synchronization residual CFO of $\Delta f = 150$ Hz (corresponding to ≈ 0.063 ppm at $f_c = 2.4$ GHz) is applied as

$$r_{\text{CFO}}[n] = s[n]\, e^{j2\pi \Delta f n T_s}.$$

This models the residual offset after initial carrier synchronization. Pre-synchronization scenarios with larger offsets (up to 5 kHz, i.e., ≈ 2 ppm) are evaluated in Sec. **??**.

Timing offset: Integer sample timing misalignment of $\tau_0 = 5$ samples $r_\tau[n] = r[n-\tau_0]$.

Low Doppler: Jakes model [**?**] with $v = 30$ km/h, $f_d \approx 67$ Hz, urban mobile scenario.

High Doppler: Jakes model with $v = 120$ km/h, $f_d \approx 267$ Hz, high-speed vehicular scenario.

Frequency-selective: $L = 6$ paths with exponential power delay profile $\mathbb{E}[|h_\ell|^2] \propto e^{-\ell/2}$ (unit total power), RMS delay spread $\tau_{\text{rms}} = 300$ ns, i.e. 2.4 samples at 8 kHz, within the CP length $N_{\text{CP}} = 16$ samples of CP-based waveforms.

Time-frequency double-selective: Combined Doppler ($v = 60$ km/h, $f_d \approx 133$ Hz) and frequency-selective multipath ($\tau_{\text{rms}} = 100$ ns, $L = 4$ paths).

3GPP TDL Standard Profiles (TDL-A, TDL-B, TDL-C): To ensure the CR waveform identifier transfers directly to real 5G NR deployment scenarios [**?**], three standardized tapped delay line profiles from 3GPP TR 38.901 are included. Each profile applies an independent complex Gaussian gain per tap (Rayleigh per tap), with delays and powers taken verbatim from the standard:

- TDL-A (NLOS, $\tau_{DS} = 30$ ns): 23 taps that model indoor hotspots and dense urban propagation, typical of small cell or IoT gateway deployments,
- TDL-B (NLOS, $\tau_{DS} = 100$ ns): 22 taps that model urban macro propagation, representative of vehicular CR and mobile spectrum monitoring scenarios,
- TDL-C (NLOS, $\tau_{DS} = 300$ ns): 24 taps that model rural macro and large delay spread scenarios, the most severe frequency selective profile and most challenging for the identification of CR.

The generalized TDL channel is implemented as:

$$r[n] = \sum_{p=1}^{P} g_p\, s[n-\tau_p] + w[n]\,, \tag{4}$$

where τ_p is the p-th tap delay (in samples, rounded from the nanosecond values in the standard), $g_p \sim \mathcal{CN}(0, \sigma_p^2)$ with $\sigma_p^2 \propto 10^{\Delta_p/10}$ and Δ_p the relative power in dB, normalized so $\sum_p \sigma_p^2 = 1$. The output power is rescaled to match input power after convolution.

2.3. Waveform Configurations and Signal Models

All eight waveforms use 4-QAM symbols. The received block $r[n]$, $n = 0, \ldots, N-1$, is the sole input to feature extraction, reflecting the CR receiver's blind observation. The identifier has no access to waveform type, channel parameters, or SNR as mandated by the CR non-cooperative sensing paradigm.

The universal filtered multi-carrier (UFMC) signal is defined as:

$$s_{\text{UFMC}}[n] = \sum_{b=1}^{B} \sum_{m=0}^{M_b-1} X_b[m]\, g_b[n - mL]\, e^{j2\pi f_b n}\,, \qquad (5)$$

$B = 8$ subbands, $M_b = 8$ subcarriers per subband, $g_b[n]$ is a 43-tap Dolph-Chebyshev FIR filter (-40 dB sidelobe suppression), $L = 1$.

Subband center frequencies

$$f_b = (b-1)\Delta f_{SB}$$

with subband spacing

$$\Delta f_{SB} = \frac{M_b f_s}{K} = 1 \text{ kHz}$$

$K = 64$ total subcarriers, $\Delta f_{sc} = f_s/K = 125$ Hz. The block length is 256 samples.

The generalized frequency division multiplexing signal is:

$$s_{\text{GFDM}}[n] = \sum_{k=0}^{K-1} \sum_{m=0}^{M-1} d_{k,m}\, g\big[(n - mK) \bmod N\big]\, e^{j2\pi \frac{k}{K} n}\,, \quad (6)$$

where $K = 64$ subcarriers, $M = 8$ time slots, raised-cosine pulse with roll-off $\alpha = 0.1$, $N = KM = 512$ samples [**?**], [**?**].

The filter bank multi-carrier with OQAM (FBMC-OQAM) signal is defined as:

$$s_{\text{FBMC}}[n] = \sum_{k=0}^{K-1} \sum_{m\in\mathbb{Z}} a_{k,m}\, p[n - mK]\, e^{j\frac{\pi}{2}k}\, e^{j2\pi \frac{k}{K} n}\,, \qquad (7)$$

where $K = 64$, $a_{k,m} \in \mathbb{R}$ real-valued OQAM symbols, the prototype PHYDYAS filter of length $4K = 256$, stagger offset $K/2$ [**?**], [**?**]. The block length is 512 samples.

The power-domain superposition non-orthogonal multiple access (NOMA) signal is defined as follows:

$$s_{\text{NOMA}}[n] = \sum_{k=0}^{K-1} \big(\sqrt{\alpha_1} X_1[k] + \sqrt{\alpha_2} X_2[k]\big) e^{j2\pi \frac{k}{K} n}\,, \qquad (8)$$

where $K = 256$, power allocation $\alpha_1 = 0.8$, $\alpha_2 = 0.2$ [**?**]–[**?**]. The block length is 256 samples.

The OFDM with index modulation (OFDM-IM) signal is the following:

$$s_{\text{OFDM-IM}}[n] = \frac{1}{\sqrt{K}} \sum_{g=1}^{G} \sum_{k\in\mathcal{I}_g} X_g[k]\, e^{j2\pi \frac{k}{K} n}\,, \qquad (9)$$

where $K = 64$ total subcarriers, $G = 12$ groups of 5 subcarriers (4 guard subcarriers), $K_a = 3$ active subcarriers per group (60% activation).

The active index set $\mathcal{I}_g$ is selected by $\lfloor \log_2 \binom{5}{3} \rfloor = 3$ index bits via a combinatorial number system lookup table [**?**]–[**?**]. A 16-sample cyclic prefix is appended. The block length is 80 samples.

The orthogonal time-frequency space (OTFS) modulates in the Doppler delay domain [**?**], [**?**], [**?**]. The 2D ISFFT grid signal is defined as follows:

$$x[n,m] = \frac{1}{\sqrt{MN}} \sum_{k=0}^{N-1} \sum_{\ell=0}^{M-1} X[k,\ell]\, e^{j2\pi\left(\frac{nk}{N} - \frac{m\ell}{M}\right)}\,, \qquad (10)$$

with $N = M = 8$ (8×8 grid).

The 2D array is serialized to a 1D sequence via the Heisenberg transform:

$$s_{\text{OTFS}}[i] = x[i \bmod N, \lfloor i/N \rfloor], \quad i = 0, \ldots, 63.$$

The block length is 72 samples.

The orthogonal delay Doppler multiplexing (ODDM) signal is the following:

$$s_{\text{ODDM}}[n] = \frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} \Big(\sum_{s=0}^{S-1} H_{k,s}\, d_s\Big) e^{j2\pi \frac{k}{K} n}\,, \qquad (11)$$

where $K = 64$, spreading factor $S = 4$, Walsh-Hadamard matrix $\mathbf{H} \in \mathbb{C}^{K\times S}$, 16-sample CP [**?**]. The block length is 80 samples.

The chirp-based affine frequency division multiplexing (AFDM) signal is defined as:

$$s_{\text{AFDM}}[n] = \sum_{k=0}^{K-1} X[k]\, e^{j\pi\mu n^2}\, e^{j2\pi \frac{k}{K} n}\,, \qquad (12)$$

where $K = 128$ chirp subcarriers, optimized chirp rate μ for Doppler tolerance $f_d \leqslant 500$ Hz Doppler tolerance [**?**]–[**?**]. The block length is 128 samples.

The sampling frequency is $f_s = 8$ kHz. For multicarrier waveforms, the relevant timing parameter is the symbol duration (e.g., $T_{\text{sym}} = K/f_s = 8$ ms for $K = 64$ subcarriers). Per-waveform block lengths are shown in Tab. **??**. The 8 kHz rate is a reduced-scale proof of concept that preserves all waveform structural properties. Scalability to 5G wideband 5G bandwidths is discussed in Subsection **??**.

2.4. Dataset Statistics

The dataset comprises 8 waveform types × 7 SNR levels $[-5, 0, 5, 10, 15, 20, 25]$ dB × 12 channel conditions × 300 realizations = 201 600 signals in total, balanced at 25 200 per waveform class. The original 9-channel subset of 151 200 signals is retained for direct comparison with previous work.

The signals use independent random seeds derived from master seed 42. A stratified 70 / 30% train-test split maintains SNR and channel diversity in both partitions. The identical partition object is reused across all pipeline steps to prevent any cross-contamination.

3. Feature Extraction Framework

Used hereafter, the term *feature* refers exclusively to a numerical descriptor computed from the received block $r[n]$ and used as a machine learning input.

The framework operates on the received block $r[n]$, $n = 0, \ldots, N-1$, in the fully blind mode. The 38 feature definitions are provided below for reproducibility.

Tab. 1. Simulation parameters used to analyze the proposed system.

Waveform	Eq.	Subcarriers	CP/guard	Block	Characteristic†
UFMC	(**??**)	64 (8 bands)	43-tap filter	256	Per-subband FIR filtering
GFDM	(**??**)	64	8 slots	512	Circular pulse shaping
FBMC	(**??**)	64	4× ovlp	512	OQAM, prototype filter
NOMA	(**??**)	256	—	256	Power domain superposition
OFDM-IM	(**??**)	64	16 samples	80	Index modulation (60%)
OTFS	(**??**)	8×8	8 samples	72	Delay-Doppler domain
ODDM	(**??**)	64	16 samples	80	Hadamard spreading
AFDM	(**??**)	128	—	128	Chirp-based, Doppler-robust

† Waveform characteristic describes the signal-processing property of each waveform design. It is not a machine learning input feature (see Sec. **??**).

3.1. Classical Signal Features (F1 – F35)

Amplitude statistics (F1 – F4): power, variance, skewness, and kurtosis of the signal envelope distinguish constant envelope waveforms (FBMC-OQAM) from their variable envelope counterparts (NOMA with power superposition, AFDM with chirp modulation).

Spectral features (F5 – F6, F16, F25 – F27, F29 – F31): spectral centroid, spread, correlation, geometric mean, flatness, occupancy, and entropy differentiate narrowband (UFMC subbands) from wideband (FBMC, NOMA) signatures and characterize spectral regularity based on the subcarrier structure.

Instantaneous frequency features (F7 – F11, F14): statistics of $\phi'[n] = \Delta(\text{unwrap}(\arg(r[n])))$ capture frequency modulation characteristics. Chirp consistency F14 specifically detects the linearly time-varying instantaneous frequency of AFDM.

Algorithm 1 Robust AMC framework (leakage-free)

Require: Received blocks $\{r_i[n]\}_{i=1}^{N_s}$, labels $\{y_i\}$
Ensure: Trained classifier $\hat{f}(\cdot)$, optimal feature indices $\mathcal{S}^*$
1: **Dataset generation:** Simulate per Tab. **??** and Sec. **??**
2: **Feature extraction:** For each block $r_i[n]$
3: Compute DFT: $R[k] = \text{DFT}(r[n])$; instantaneous phase $\phi[n] = \text{unwrap}(\arg(r[n]))$
4: Compute instantaneous frequency $\phi'[n] = \Delta\phi[n]$
5: Compute time-frequency representations (STFT, WVD, FrFT)
6: Extract f_1–f_{35} (Tab. **??**) and f_{36}–f_{38}, Eqs. (**??**) – (**??**)
7: **Train/test split:** Stratified 70 / 30% partition $\mathcal{P}$; save $\mathcal{P}$ for reuse
8: **Normalization:** $\tilde{f}_i = (f_i - \mu_{\text{train}})/\sigma_{\text{train}}$
9: **Feature selection (leakage-free):** Apply Alg. **??** to training fold only; select top-K features $\mathcal{S}^*$
10: **Training:** Train all classifiers on $\tilde{f}_{i,\mathcal{S}^*}$ using same $\mathcal{P}$; hyperparameters via inner 5-fold CV on training fold
11: **Evaluation:** Accuracy, precision, recall, F1, McNemar tests, Wilson CI on 30% test set
12: **Return** $\hat{f}(\cdot)$, $\mathcal{S}^*$

Time-frequency analysis (F12 – F15): FrFT peak ratio F12 detects chirp signals, WVD linearity F13 measures time-frequency concentration, spectrogram TF entropy F15 quantifies joint time-frequency complexity. Note: WVD calculation $\mathcal{O}(N^2 \log N)$ is the costliest feature to compute. It ranks 15th in terms of importance and is eliminated in the 10-feature deployment subset.

Waveform-specific features (F17 – F24, F28, F32 – F35): PAPR F18 distinguishes single-carrier from multicarrier waveforms. CP correlation F24 identifies CP-OFDM structures. Periodicity F32 and cyclic peak F33 detect block structural repetition.

3.2. New Channel-aware Features (F36 – F38)

Three features explicitly designed to capture propagation effects, invariant to waveform structure:

Amplitude fading variance (F36):

$$F_{36} = \frac{\text{Var}(P_{\text{local}})}{\left(\mathbb{E}[P_{\text{local}}]\right)^2}\,, \tag{13}$$

where:

$$P_{\text{local}}[k] = \frac{1}{W} \sum_{n=kW}^{(k+1)W-1} \left|r[n]\right|^2\,,$$

$$W = \min\left(32, \lfloor N/4 \rfloor\right)\,.$$

It is near zero under AWGN ($F_{36} < 0.1$) and elevated under Rayleigh/Rician fading ($F_{36} > 0.3$).

Phase discontinuity (F37):

$$F_{37} = \text{std}\left(\phi''[n]\right)\,, \tag{14}$$

where

$$\phi''[n] = \Delta^2\left(\text{unwrap}(\arg(r[n]))\right)$$

is the second order phase difference. Pearson correlation with CFO magnitude: $\rho = 0.82$.

Frequency drift (F38):

$$F_{38} = \text{std}(f_{\text{local}})\,, \tag{15}$$

where

$$f_{\text{local}}[k] = \frac{1}{W} \sum_{n=kW}^{(k+1)W-1} \phi'[n]$$

increases 3 – 5× from 30 km/h to 120 km/h Doppler scenarios.

Ablation studies (Tab. **??** in Subsection **??**) confirm that the removal of F36 – F38 degrades accuracy by up to 3.1% under double selective fading, validating their channel-specific value. F36 ranks 7th out of 38 features (ensemble score 0.360); F37 ranks 21st (score 0.190). F38 ranks 36th out of 38 (score 0.103) and does not enter the 10-feature deployment subset. Its contribution is most pronounced under extreme CFO conditions ($>$ 2 kHz) encountered only in pre-synchronization scenarios.

4. Multi-algorithm Classification Study

4.1. Evaluated Classifiers

Nine classifiers that span classical, ensemble, and deep learning families are evaluated. For each classifier, we list the key hyperparameters that govern its decision boundary or capacity. The full grid-search details are given in Sec. **??**:

1) k-NN: $k = 5$, Euclidean distance; majority vote on five nearest neighbors. Simple non-parametric baseline.
2) SVM-linear: ECOC one-versus-all, linear kernel, regularization $C = 1$. Decision boundaries are hyperplanes in the 38-D feature space.
3) SVM-RBF: Gaussian kernel (RBF), $C = 10$, automatic kernel scale estimation ($\gamma =$ auto). Maps the features to a high-dimensional reproducing kernel Hilbert space.
4) Decision tree: CART algorithm, Gini impurity criterion, maximum 100 splits. Single-tree baseline; the shallowest interpretable model evaluated.
5) Random forest: 150 decision trees, minimum 3 samples/leaf, bootstrap aggregating (bagging). Provides feature importance scores used in Sec. **??**.
6) Naive Bayes: Kernel density estimation (KDE) for continuous features; assumes conditional independence across the 38 features.
7) Ensemble-boost (AdaBoostM2): 100 boosting iterations, weak decision tree learner (maximum 20 splits per tree; weights misclassified samples more heavily each round.
8) Ensemble bag: 100 decision trees, bootstrap sampling, maximum 50 splits/tree; similar to random forest but without feature subsampling at each split.
9) FC-MLP (deep-learning baseline): Fully-connected network $256 \to 128 \to 64 \to 8$ [**?**]; batch normalization, ReLU, dropout $p = 0.3$ per hidden layer; Adam optimizer, ℓ_2 regularization $\lambda = 10^{-4}$, 100 epochs. Serves as a deep learning reference point under identical feature conditions.

All classifiers use z-score normalized features

$$\tilde{f} = \frac{f - \mu_{\text{train}}}{\sigma_{\text{train}}},$$

training-set statistics only. Random seed 42 for reproducibility.

4.2. Hyperparameter Optimization

All hyperparameters were optimized through a 5-fold cross-validation on the 70% training set using grid search. Key results: optimal k-NN $k = 5$ from $\{1, 3, 5, 7, 10, 15\}$; SVM-linear $C = 1$ from $\{0.01, 0.1, 1, 10, 100\}$; SVM-RBF $C = 10$, $\gamma =$ auto; decision tree maximum 100 splits; random forest 150 trees, min leaf 3; ensemble-boost 100 iterations, max 20 splits; ensemble-bag 100 trees, max 50 splits.

FC-MLP: initial learning rate from $\{10^{-4}, 10^{-3}, 10^{-2}\}$ (optimal 10^{-3}); dropout from $\{0.2, 0.3, 0.5\}$ (optimal 0.3); ℓ_2 weight decay from $\{10^{-5}, 10^{-4}, 10^{-3}\}$ (optimal 10^{-4}); architecture depth 2 vs. 3 hidden layers and mini-batch size $\{128, 256\}$ also evaluated. This systematic procedure ensures a fair comparison between classifiers.

4.3. Performance Metrics

We report accuracy, i.e., overall correct rate, macro-averaged precision, recall, F1 score, training time (wall-clock seconds) and per segment inference time – milliseconds per complete received block, including feature extraction and prediction.

4.4. Experimental Results

Key observations are as follows:

FC-MLP achieves the highest accuracy of 99.09%, marginally outperforming ensemble boost (99.04%) and random forest (99.02%). McNemar tests show that the FC-MLP vs. random forest gap ($p = 0.039$) and FC-MLP vs. ensemble-boost gap ($p = 0.082$) are both not statistically significant after Bonferroni correction $\alpha_B = 0.05/36 \approx 0.00139$, meaning all three top classifiers are statistically equivalent at this operating point.

Ensemble boost and random forest are near-equivalent (both > 99%), with ensemble boost vs. random forest gap ($p = 0.649$, not significant), confirming that they are interchangeable for this problem.

SVM-RBF requires the longest training time (2 545.7 s) due to the 38-D high-dimensional feature space, where automatic kernel scale estimation is suboptimal despite z-score normalization. Naive Bayes has the highest inference latency of 51.68 ms/segment, making it unsuitable for real-time cognitive radio applications.

FC-MLP, ensemble boost, and random forest all achieve real-time inference ($<$ 1 ms/segment), enabling throughput of $>$ 10 000 segments/s. Note that the 0.071 ms figure for random forest applies to the 10-feature deployment model (see Tab. **??** footnote); the 38-feature RF baseline achieves 0.021 ms inference.

Decision tree is the best lightweight option (98.70%, 3.5 s training, near-zero inferences) for resource-constrained deployments accepting only a small accuracy trade-off over ensemble methods.

Tab. 2. Feature definitions F1 – F35 plus structural-auxiliary FA – FC.

ID	Name	Definition
F1	Signal power	$\frac{1}{N}\sum_n \lvert r[n]\rvert^2$
F2	Amplitude variance	$\mathrm{Var}(A[n])$
F3	Skewness	$\mathbb{E}[(A-\mu_A)^3]/\sigma_A^3$
F4	Kurtosis	$\mathbb{E}[(A-\mu_A)^4]/\sigma_A^4$
F5	Spectral centroid	$\sum_k k\lvert R[k]\rvert^2 / \sum_k \lvert R[k]\rvert^2$
F6	Spectral spread	$\sqrt{\sum_k (k-F_5)^2 \lvert R[k]\rvert^2 / \sum_k \lvert R[k]\rvert^2}$
F7	IF Std Dev	$\mathrm{std}(\phi'[n])$
F8	IF linear R^2	Linear regression R^2 of $\phi'[n]$ vs. n
F9	IF quadratic R^2	Quadratic regression R^2 of $\phi'[n]$ vs. n
F10	IF curvature	$\mathrm{mean}(\Delta^2\phi'[n])$
F11	IF rate variance	$\mathrm{Var}(\Delta\phi'[n])$
F12	FrFT peak ratio	$\max_\alpha \lvert X_\alpha(u)\rvert^2 / E_{\text{total}}$
F13	WVD linearity	R^2 of Wigner-Ville ridge fit in TF plane
F14	Chirp consistency	$1 - \mathrm{std}(\phi'[n])/\mathrm{mean}(\lvert\phi'[n]\rvert)$
F15	TF entropy	$-\sum_{t,f} P(t,f)\log P(t,f)$ (STFT spectrogram)
F16	Spectral correlation	$\sum_k R[k]R^*[k+1] / \sum_k \lvert R[k]\rvert^2$
F17	Zero-crossing rate	$(N-1)^{-1}\sum_n \mathbf{1}[r[n]r[n-1]<0]$
F18	PAPR	$\max_n \lvert r[n]\rvert^2 / \mathbb{E}[\lvert r[n]\rvert^2]$
F19	Envelope mean	$\mathbb{E}[A[n]]$
F20	Crest factor	$\max_n A[n]/\sqrt{\mathbb{E}[A[n]^2]}$
F21	IF mean	$\mathbb{E}[\phi'[n]]$
F22	Phase variance	$\mathrm{Var}(\phi[n])$
F23	Phase entropy	$-\sum_b p_b \log p_b$ (histogram of $\phi[n]$, 32 bins)
F24	CP correlation	$\lvert\sum_n r[n+N_{\text{CP}}]r^*[n]\rvert / \mathbb{E}[\lvert r\rvert^2]$
F25	Spectral geom. mean	$\exp\big(\frac{1}{N}\sum_k \log\lvert R[k]\rvert\big)$
F26	Spectral flatness	$F_{25}/\mathbb{E}[\lvert R[k]\rvert]$
F27	Spectral occupancy	Fraction of bins: $\lvert R[k]\rvert^2 > 0.1\max\lvert R[k]\rvert^2$
F28	Autocorrelation peak	$\max_{\tau>0}\lvert R_{rr}[\tau]\rvert / R_{rr}[0]$
F29	Spectral entropy	$-\sum_k p_k\log p_k$, $p_k = \lvert R[k]\rvert^2/\sum\lvert R[k]\rvert^2$
F30	Subband variance	$\mathrm{Var}(\bar{P}_b)$, $b=1,\ldots,8$ equal-width subbands
F31	Subcarrier regularity	Periodicity score of $\lvert R[k]\rvert^2$ at subcarrier spacing
F32	Periodicity	Peak of normalized autocorrelation of $A[n]$
F33	Cyclic peak	Max of cyclic autocorrelation spectrum
F34	Spectral coherence	$\mathrm{Coh}(R[k], R[k+\Delta])$ averaged over lag Δ
F35	Symbol rate estimation	Dominant period of $\lvert r[n]\rvert^2$ autocorrelation
FA†	Filter sharpness	Ratio of stopband to passband power in subband spectral profile
FB†	Delay correlation	$\lvert\sum_n r[n]r^*[n-D]\rvert / R_{rr}[0]$ at lag $D = N_{\text{CP}}$
FC†	Constellation shape	Normalized variance of $\lvert r[n]\rvert^2$ over symbol grid

Notation: $R[k]$ = DFT of $r[n]$, $A[n] = \lvert r[n]\rvert$, $\phi[n] = \mathrm{unwrap}(\arg(r[n]))$, $\phi'[n] = \Delta\phi[n]$.

†Structural-auxiliary features FA – FC (internal codes `Filter_Sharp`, `Delay_Corr`, `Const_Shape`) rank 10th, 11th, and 13th respectively in Tab. **??**. They are extracted from the received block $r[n]$ in fully blind mode and complement F1 – F35. Note: the labels F36, F37, F38 are reserved exclusively for the three novel *channel-aware* features: amplitude fading variance, phase discontinuity, and frequency drift, defined in Sec. **??**, Eqs. (**??**) – (**??**).

Tab. 3. Multi-algorithm classification performance on the held-out test set (38 features, 70/30 train/test split). Best values are bold.

Classifier	Acc [%]	Prec [%]	Rec [%]	F1 [%]	Train [s]	Inf [ms]	RT?
FC-MLP	**99.09**	**99.10**	**99.09**	**99.09**	970.8	0.021	✓
Ensemble-boost	99.04	99.05	99.04	99.04	157.0	0.023	✓
Random forest†	99.02	99.03	99.02	99.03	287.2	0.071‡	✓
Decision tree	98.70	98.71	98.70	98.70	3.5	0.001	✓
SVM-linear	97.57	97.62	97.57	97.59	388.0	0.009	✓
SVM-RBF	97.19	97.39	97.19	97.29	2545.7	3.544	×
Ensemble-bag	97.04	97.29	97.04	97.16	47.4	0.024	✓
k-NN	96.94	97.04	96.94	96.99	2.5	1.249	×
Naive Bayes	95.85	96.07	95.85	95.96	3.1	51.681	×

RT? = inference $<$ 1 ms/sample.
†RF deployed with 10 features.
‡inference time measured on the 10-feature deployment model (0.021 ms at 38 features).

4.5. McNemar Tests

The accuracy differences between all classifier pairs are assessed using the McNemar test with continuity correction [**?**], which is applied to correct / incorrect vectors per sample. Bonferroni correction for $\binom{9}{2} = 36$ comparisons gives $\alpha_B = 0.05/36 \approx 0.00139$. The full pairwise p-value matrix is shown in Tab. **??**. Key findings are as follows:

- FC-MLP, ensemble boost, and random forest are mutually statistically equivalent ($p > \alpha_B$ for all three pairwise comparisons), confirming that these three top classifiers are interchangeable at this operating point,
- All other classifiers are significantly different from the random forest ($p < 0.001$), confirming meaningful accuracy gaps,
- The 10-feature random forest vs. 38-feature random forest gap is not significant ($p > 0.1$), statistically confirming that the compact subset does not lose significant precision.

5. Feature Importance and Selection

Five complementary methods are applied to obtain a robust importance ranking:

1) Random forest permutation importance – OOB accuracy degradation when each feature is permuted,
2) ReliefF – weight-based inter-class vs. intra-class separation (10 nearest neighbors),
3) Chi-square – statistical independence test (10 bins vs. class labels),
4) Mutual information – $I(F;Y) = H(Y) - H(Y|F)$,
5) Fisher score – the ratio of between-class to within-class variance.

The ensemble score shown as Algorithm **??** uses weights $\mathbf{w} = [0.30, 0.175, 0.175, 0.175, 0.175]$, with a higher weight for the importance of random forests due to its direct connection to classifier performance:

$$s_{\text{ensemble}}[i] = \sum_{m=1}^{5} w_m \cdot s_m[i] \,. \tag{16}$$

Algorithm 2 Ensemble feature scoring (leakage-free training fold only)

Require: Feature matrix $\mathbf{F}_{\text{train}} \in \mathbb{R}^{N_{tr} \times 38}$ (training fold only), labels $\mathbf{y}_{\text{train}}$, weights $\mathbf{w}$
Ensure: Ensemble scores $s_{\text{ens}}[i]$, $i = 1, \ldots, 38$
1: **for** $m = 1$ to 5 **do**
2: Compute raw scores $\tilde{s}_m[i]$ with method m on $(\mathbf{F}_{\text{train}}, \mathbf{y}_{\text{train}})$
3: Normalize: $\dfrac{s_m[i] = (\tilde{s}_m[i] - \min_i \tilde{s}_m)}{\max_i \tilde{s}_m - \min_i \tilde{s}_m}$
4: **end for**
5: $s_{\text{ens}}[i] = \sum_{m=1}^{5} w_m \cdot s_m[i]$
6: **Return** s_{ens}

All scores normalized to $[0, 1]$ before averaging.

5.1. Top Essential Features

Periodicity (rank 1) and subcarrier entropy (rank 2) are the dominant discriminators. OFDM-based waveforms (UFMC, OFDM-IM, GFDM) exhibit a highly regular and periodic subcarrier structure, whereas NOMA (power superposition) and AFDM (chirp modulation) do not. The ensemble score is undefined (NaN), because the chi-square method is not applicable to near-binary indicator features. It is ranked first by all four remaining methods and is assigned rank 1 by consensus.

Channel-aware F36 (`Amp_Fade_Var`) ranks seventh out of 38 (ensemble score 0.360), confirming that explicit fading characterization is highly discriminative. Different waveforms exhibit distinct fading robustness profiles, making channel-induced amplitude variance a powerful and early entering classification cue.

Higher-order statistics (Skewness rank 5, Kurtosis rank 14) distinguish the constant-envelope FBMC-OQAM from the variable-envelope NOMA. Time-frequency entropy (`TF_Entropy`, rank 6) provides robustness to Doppler and multipath through joint time-frequency analysis – a characteristic unavailable to purely spectral or temporal features.

Tab. 4. McNemar test p-values (Bonferroni $\alpha_B = 0.05/36 \approx 0.00139$).

	k-NN	SVM-linear	SVM-RBF	Decision tree	Random forest	Naive Bayes	Ensemble-boost	Ensemble-bag	FC-MLP
k-NN	–	**0.000**	0.001†	**0.000**	**0.000**	**0.000**	**0.000**	0.218	**0.000**
SVM-linear		–	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
SVM-RBF			–	**0.000**	**0.000**	**0.000**	**0.000**	0.043	**0.000**
Decision tree				–	**0.000**	**0.000**	**0.000**	**0.000**	**0.000**
Random forest					–	**0.000**	0.649	**0.000**	0.039
Naive Bayes						–	**0.000**	**0.000**	**0.000**
Ensemble-boost							–	**0.000**	0.082
Ensemble-bag								–	**0.000**
FC-MLP									–

Bold entries are statistically significant ($p < \alpha_B$). FC-MLP, ensemble-boost, and random forest are mutually non-significant (blue).
†Exact $p = 0.00148 > \alpha_B \approx 0.00139$: not significant after Bonferroni correction.
All values verified against `mcnemar_tests.mat`.

Tab. 5. Top 15 features ranked by weighted ensemble score (RF 30% + ReliefF 17.5% + χ^2 17.5% + MI 17.5% + Fisher 17.5%).

Rank	Feature	Score	Type
1	Period	–†	Spectral
2	Subcar. Entropy	0.825	Temporal
3	Var. Amp	0.515	Spectral
4	Spec. Occup	0.502	Spectral
5	Skewness	0.428	Statistical
6	TF Entropy	0.405	Statistical
7	**Amp. Fade Var (F36)**	**0.360**	**Channel-aware**
8	Spec. Coher	0.353	Spectral
9	CP Corr	0.326	Waveform-specific
10	Filter Sharp	0.321	Spectral
11	Delay Corr	0.315	Statistical
12	Subcar. Reg	0.306	Spectral
13	Const. Shape	0.275	Statistical
14	Kurtosis	0.275	Statistical
15	IF Std	0.272	Inst.-freq.

† Score undefined: χ^2 returns NaN for period (near-binary indicator incompatible with frequency-bin discretization). The period is ranked first by all the other four methods; it is assigned rank 1 by consensus. Yellow rows: optimal 10-feature deployment subset. F36 (`Amp_Fade_Var`) is the novel channel-sensitive feature. Scores verified against `feature_importance.mat`.

5.2. Channel-aware Feature Ablation Study

Table **??** shows the impact of removing channel-aware features. F36 is most critical for fading (–1.4% when removed alone), F37 for Doppler and CFO conditions (–1.1% alone under high Doppler). F38 (rank 36/38, score 0.103) contributes primarily under extreme pre-synchronization CFO scenarios and does not enter the 10-feature subset.

Tab. 6. Ablation study: accuracy in percent of the 10-feature random forest with/without channel-aware features. Verified against `per_channel_confusion.mat`.

Channel	All	F36	F37	F38	None
AWGN	99.8	99.8	99.8	99.8	99.8
Rayleigh/Rician	98.5	97.1	98.3	98.4	96.4
Frequency offset	97.9	97.8	96.8	97.7	96.6
Low Doppler	98.2	98.0	97.7	97.9	96.9
High Doppler	96.8	96.6	95.7	96.3	94.2
Freq.-selective	97.4	97.1	97.3	97.3	96.8
Timing offset	98.1	98.0	97.4	98.0	97.2
Double-selective	96.1	95.5	95.3	95.6	93.0
Mean	97.9	97.5	97.3	97.6	96.4

6. Feature Reduction Analysis

The random forest achieves a performance rate of 98.90% with 10 features and 99.05% with 15, a gap of 0.15 pp. The 10-feature configuration is within 0.12 percentage points (pp) of the full-feature baseline at 74% dimensionality reduction. Strong performance is achieved even with five features (95.38 – 97.76%), well above the 12.5% chance level for 8 classes, and all classifiers already exceed 95%.

There is a plateau beyond 10 features for the random forest, where the accuracy fluctuates by no more than 0.13 pp from 10 to 38 features. Ensemble-boost shows a 0.22 pp gap at 10 features (98.82% vs. 99.04% at 38 features). The plateau of < 0.15 pp is reached from 15 features onward for all ensemble methods.

All four classifiers converge to near-optimal at 10 features, validating classifier-independent feature selection. FC-MLP tracks random forest and ensemble boost closely across all feature counts.

Tab. 7. Classification accuracy (in percent) vs. feature count for four classifiers.

Feature count	Ens.-boost	Rand. forest	Dec. tree	FC-MLP
5	97.58	97.76	95.38	97.13
10	98.82	98.90	97.52	98.82
15	98.97	99.05	98.10	99.02
20	99.02	99.06	98.43	99.05
25	99.03	99.05	98.78	99.08
30	99.03	99.06	98.70	99.04
38	99.04	99.02	98.70	99.09

RF: <0.17 pp gain beyond 10 features (38F baseline 99.02%). Ens.-Boost: 0.22 pp gap at 10F; plateau <0.15 pp from 15F onward. All classifiers use the same subset of ranked features and identical 70/30 train/test partition. Optimal row (10 features) shaded. Decision tree accuracy recovers more slowly due to overpruning. Values verified against `feature_reduction.mat`.

The 10-feature (10F) configuration yields: (i) 74% reduction in feature extraction operations; the costliest feature, WVD (F13, $\mathcal{O}(N^2 \log N)$, rank 15), is eliminated, reducing total extraction time from 18.7 ms to 2.1 ms per block at the 8 kHz simulation rate – see Sec. **??** for wideband projections; (ii) 73.7% memory reduction from 152 to 40 bytes per feature vector; (iii) 0.071 ms inference time per segment on a standard Intel Core i7-8550U, 2.4 GHz processor, enabling $>$ 14 000 segments/s.

7. Cross-validation and Robustness Analysis

7.1. K-Fold Cross-validation

The 10-feature random forest model undergoes 5-fold stratified cross-validation: mean accuracy 98.54%, $\sigma = 0.05\%$, Wilson 95% CI [98.49%, 98.59%] [**?**] and coefficient of variation 0.05%. The Wilson interval is used in preference to the normal approximation, which is inaccurate near $p = 1$. The narrow interval confirms the exceptional generalization stability. Feature selection is performed independently within each fold's training partition, preserving strict leakage-free evaluation throughout.

7.2. Per-Channel Performance

Using the 10-feature ensemble-boost model on the nine custom channels (all values verified against `per_channel_confusion.mat`):

- AWGN: 99.96%,
- Rayleigh fading: 98.87% (F36 maintains discrimination under amplitude fluctuations),
- Rician fading: 99.98%,
- Frequency offset: 99.42% (cyclostationary features robust to constant CFO; residual confusion in AFDM/OTFS pair),
- Timing offset: 99.92%,

Tab. 8. Confusion matrix under the Rayleigh fading channel.

True/Pred	UFMC	GFDM	FBMC	NOMA	OFDM-IM	OTFS	ODDM	AFDM
UFMC	**2100**	0	0	0	0	0	0	0
GFDM	0	**2100**	0	0	0	0	0	0
FBMC	0	0	**2007**	93	0	0	0	0
NOMA	0	0	56	**2044**	0	0	0	0
OFDM-IM	0	0	0	0	**2082**	0	18	0
OTFS	0	0	0	0	0	**2100**	0	0
ODDM	0	0	0	0	23	0	**2077**	0
AFDM	0	0	0	0	0	0	0	**2100**

10-feature ensemble-boost model; 2100 test samples per class. The bold diagonal entries are per class recall.

- Low Doppler (30 km/h): 98.93%,
- High Doppler (120 km/h): 99.04%,
- Frequency-selective: 99.98% (multipath creates distinctive temporal correlation aiding UFMC identification),
- Double selective time-frequency: 99.02% without F36 – F38, this drops to 93.0%, as reported in the ablation study in Tab. **??**.

3GPP TDL standard channels (verified against `per_channel_confusion.mat`):

- TDL-A (30 ns delay spread, indoor/dense urban): 99.96% confirming transferability of selected features to standard profiles,
- TDL-B (100 ns, urban macro): 99.95%,
- TDL-C (300 ns, rural macro): 99.97% – the largest standard profile delay spread, yet performance remains consistently high, confirming robustness of the selected feature set.

Table **??** shows the 8 × 8 confusion matrix generated by the 10-feature model under the Rayleigh fading channel. UFMC, GFDM, OTFS and AFDM achieve perfect classification (2100 / 2100). The primary confusion occurs between FBMC and NOMA (93 FBMC samples misclassified as NOMA, 56 NOMA samples misclassified as FBMC), which is attributable to their overlapping envelope statistics under amplitude-varying fading. A secondary confusion cluster appears between OFDM-IM and ODDM (18 and 23 samples, respectively), reflecting their shared index-modulation and spreading characteristics.

7.3. Parametric Robustness Analysis

SNR sensitivity (10-feature random forest model): at 5 dB is 84.5%; at 0 dB is 93.2%; at +5 dB is 97.6%. In $\geqslant$ 10 dB range it equals $\geqslant$ 98.4%. Below –5 dB, signal averaging or diversity combining is recommended.

Doppler sensitivity: 0 km/h: 99.4%; 30 km/h: 98.2%; 60 km/h: 97.5%; 120 km/h: 96.8%. Performance degrades gracefully across the full range from urban to highway. CFO sensitivity: 150 Hz (post-synchronization, $\approx$ 0.063 ppm):

Tab. 9. Comparison with state-of-the-art AMC methods. This work adds a FC-MLP deep learning baseline and 3GPP TDL standard channels.

Reference	Waveforms	Channels	Acc [%]	Method
[?]	11 conv.	AWGN only	94.20	HO cumulants
[?]	5 NG	Simple	95.00	Deep ResNet+PCA
[?]	Conv.	AWGN+ Rician	96.80	CNN-LSTM
[?]	Conv.	AWGN	95.60	Deep CNN
This work (FC-MLP, 38F)	8 NG	12 channels	99.09	MLP 256-128-64
This work (RF, 10F)	8 NG	12 channels	98.90	RF, 10 features

74% reduction in features, 0.071ms / segment, capable of real-time.
NG = next-generation waveform.

97.9%; 1 kHz: 96.9%; 2.4 kHz ($\approx$ 1 ppm at 2.4 GHz, pre-synchronization): 96.3%; 5 kHz: 93.1%. The 2.4 kHz case remains at 96.3%, confirming practical robustness for consumer-grade oscillator mismatch. Timing offset sensitivity: 0 samples: 99.8%; 5 samples: 98.1%; 10 samples: 97.9%; 15 samples: 96.6% (faster degradation as CP is partially corrupted).

8. Discussion

8.1. Comparison with Other Work

The proposed framework achieves a 2.09 – 4.89 percentage point improvement over the state-of-the-art solution on comparable or simpler waveform sets – Tab. **??**. The authors of [**?**] used higher-order cumulants for conventional PSK/QAM and their approach does not extend to next-generation multicarrier waveforms. In [**?**], a deep residual network achieving approximately 95% accuracy is proposed in five 5G waveforms, but the evaluation excluded delay Doppler waveforms (OTFS, ODDM, AFDM) and compound channel conditions. Frameworks described in [**?**], [**?**] operate on conventional modulation schemes under simpler channel models.

Critically, the deep learning baseline trained on the same 38 characteristics achieves 99.09%, which is the highest single model precision score and is statistically equivalent to that of ensemble boost and random forest (McNemar $p > \alpha_B$ for all three pairwise comparisons). This result demonstrates two complementary points: (i) the hand-crafted features are rich enough to support top-tier deep learning classification, and (ii) structured ensemble classifiers that explicitly exploit feature interactions via tree splits achieve performance equivalent to that of generic MLP architectures on this problem, while offering physical interpretability.

The dominant discriminators, i.e., periodicity, subcarrier entropy, F36, are explicitly identified – a feat that end-to-end black-box approaches cannot offer.

8.2. Practical CR Deployment Considerations

The 10-feature compact model is specifically designed for the resource envelope of real CR hardware. It requires $<$ 40 KB total memory (40-byte feature vectors plus 35 KB for random forest model), placing it within the capabilities of ARM Cortex-M7 microcontrollers, FPGAs, software-defined radio platforms (USRP, HackRF, LimeSDR), UAV-mounted spectrum sensors and battery-powered secondary IoT nodes. With 0.071 ms inference time per segment, the identifier imposes negligible latency overhead on the CR sensing decision cycle and may comfortably run in parallel with other CR functions such as channel estimation and power control.

For access to the dynamic CR spectrum, the identification latency directly determines how quickly a secondary user can react to the transmission of a primary user. At 0.071 ms per block, the system supports $>$ 14 000 spectrum sensing decisions per second within the millisecond reaction times required by 5G NR and anticipated by 6G MAC protocols. The 40-byte feature vector also lends itself to cooperative CR sensing – secondary nodes can broadcast their feature vectors to a fusion center for ensemble decision-making without significant control channel overhead.

8.3. Scalability to Wideband CR Systems

All frequency domain features are normalized by the Nyquist frequency, and all temporal features use symbol-period units rather than absolute sample counts. Therefore, the features are scale-invariant by design. A CR receiver observing a 5G NR signal at 100 MHz bandwidth would extract identical normalized subcarrier regularity and spectral occupancy ratios as in the 8 kHz experiments reported here.

It is important to be explicit about the current scope. The experiments use f_s = 8 kHz as a reduced-scale proof-of-concept that preserves all waveform structural properties (subcarrier relationships, CP ratios, Doppler-to-bandwidth ratios) while remaining computationally tractable for a comprehensive 201 600 signal, 12 channel, 9 classifier study. The extrapolation to wideband CR bandwidths is analytically motivated, but is not yet hardware validated.

Feature extraction time scales as follows: at 8 kHz, the dominant cost is WVD ($\mathcal{O}(N^2 \log N)$), giving 18.7 ms per block for the full 38-feature set and 2.1 ms for the 10-feature subset (WVD eliminated). At 100 MHz, representative of a wideband CR receiver, symbol-period windows are proportionally shorter ($\sim$150$\times$ fewer samples for the same number of symbols), so extraction time is expected to reduce to $\approx$ 30 µs per block. Hardware validation at 5G NR bandwidths (e.g., on a USRP B210 or ZCU111 RFSoC operated as a CR front end) remains a priority for future work and is listed as a limitation below.

8.4. Limitations and Future Directions

All experiments use f_s = 8 kHz. Scale invariance is analytically guaranteed by feature normalization but not yet confirmed by hardware measurement at 5G NR bandwidths on

a real CR platform. USRP or RFSoC validation is the most urgent open item.

Waveform block lengths range from 72 (OTFS) to 512 samples (GFDM, FBMC). In a CR receiver with no a priori knowledge of the waveform, block segmentation must be fixed, and longer blocks yield more stable feature estimates. Per-waveform confusion analysis (Tab. **??**) reveals that FBMC and NOMA sharing overlapping envelope statistics under fading are the primary confusion pair. Future work should investigate adaptive block-length strategies or multi-block averaging.

The current framework assumes that one of the eight known waveforms is always present. Confidence-based rejection (maximum posterior class probability < 0.7) can flag unknown or novel waveforms; preliminary experiments correctly rejected 78% of artificially injected out-of-distribution signals. Open-set CR sensing remains an open problem.

A CR receiver's own front-end introduces phase noise of –80 dBc/Hz and I/Q imbalance (0.1 – 0.5 dB amplitude mismatch), causing an additional accuracy loss of up to 1.4% and 0.8%, respectively, in supplementary Saleh PA model experiments. The 16-QAM substitution causes only a degradation of 0.3 to 0.8%, confirming that waveform-structural features are largely modulation order independent, an important property for blind CR sensing.

Sinc-interpolation-based offsets of 0.1 – 0.5 samples cause an additional degradation of 0.4 – 1.2%, mainly affecting CP correlation F24. A CR receiver with multiple antennas could exploit MIMO spatial signatures as additional discriminative features.

Future directions are as follows:

- Hardware CR validation: Deploy the 10-feature identifier on a USRP B210 or ZCU111 RFSoC operating as a CR secondary node, and confirm real-time throughput and scale invariance at 5G NR bandwidths.
- Raw I/Q deep learning for CR: 1D-ResNet or transformer trained on raw I/Q, with the 38-feature model as an interpretable reference for attention visualization and CR explainability requirements.
- Transfer and meta-learning: A few-shot adaptation of the CR identifier to novel 6G waveform candidates, using the 10-feature space as a compact domain-shift-robust representation.
- Hierarchical CR identification: Two-stage approach (waveform family, then specific waveform) to resolve FBMC/NOMA and OFDM-IM/ODDM confusion, reducing misclassification-driven interference in CR systems.
- Joint identification and parameter estimation: Simultaneous waveform identification and symbol rate/modulation order estimation to fully characterize the primary user signal for CR resource allocation.
- Cooperative CR sensing: Distributed identification where multiple CR secondary nodes share compact feature vectors for fusion center ensemble decisions, exploiting spatial diversity.
- Open-set CR sensing: The detection of novelty of a single class SVM or isolation forest in the 10-feature space to flag unknown or emerging waveforms not in the training set.

9. Conclusions

This paper presented a framework for the identification of cognitive radio waveforms for eight 5G-Advanced/6G waveform candidates under realistic compound channel impairments, addressing the fundamental CR requirement of blind, non-cooperative waveform sensing. The study was extended with three 3GPP TDL standard profiles, an FC-MLP deep learning baseline, leakage-free feature selection, and McNemar statistical validation. The following findings have been verified based on 201 600 signals under twelve channel conditions and with seven SNR levels:

1) FC-MLP achieves 99.09% waveform identification accuracy with 38 features 2.09 – 4.89 pp above prior state-of-the-art solution in comparable or simpler waveform sets. FC-MLP, ensemble boost (99.04%) and random forest (99.02%) are mutually statistically equivalent (McNemar $p > \alpha_B$ for all pairwise comparisons), giving CR system designers flexibility in choosing between interpretable ensemble methods and deep learning.
2) The hand-crafted 38-feature set fully supports deep-learning-based CR identification. FC-MLP reaches 99.09% on the same features, while ensemble tree methods achieve statistically equivalent performance with greater interpretability, a key property for CR regulatory and cognitive transparency requirements.
3) Compact CR deployment enabled by 10-feature random forest at 98.90% within 0.12% of the full-feature baseline (McNemar $p > 0.1$), with a 74% reduction in feature extraction cost and 0.071 ms per segment inference, satisfying the real-time latency budget for dynamic CR spectrum access.
4) Novel channel-aware features are critical for the robustness of CR. Ablation studies confirm an accuracy improvement of up to 3.1% under double-selective fading (93.0% → 96.1%), i.e. the most adversarial CR propagation scenario, with F36 (amplitude fading variance) ranking 7th among 38 features and entering the optimal 10-feature CR deployment subset. F38 (frequency drift, rank 36/38) contributes primarily under extreme CFO conditions and does not enter the 10-feature subset.
5) Robustness throughout the entire CR channel envelope. Identification accuracy boosted from 98.87% (Rayleigh fading) to 99.98% (AWGN and Rician fading) on custom channels and from 99.95% (TDL-B) to 99.97% (TDL-C) on 3GPP standard profiles, confirming direct transferability to 5G NR CR deployments. Per-waveform confusion analysis identifies FBMC/NOMA

and OFDM-IM/ODDM as the primary confusion pairs under Rayleigh fading, providing actionable guidance for CR system design.

6) Cross-validation confirms generalization stability: 98.54% mean, Wilson 95% CI [98.49%, 98.59%], σ = 0.05% in 5 folds. All feature selection is performed within training folds, ensuring genuine out-of-sample performance estimates suitable for CR deployment decisions.

7) Algorithm independent compact sensing. The top 10 leakage-free feature subsets are transferred across random forest, ensemble-boost, decision tree, and FC-MLP, enabling CR platform designers to select their preferred classifier without redesigning the sensing front-end.

The proposed framework enables practical cognitive radio waveform identification (40-byte feature vectors, 0.071 ms latency, $>$ 14 000 sensing decisions per second) while providing a physical insight into discriminative mechanisms: periodicity, subcarrier entropy, and channel-aware F36 that CR systems can leverage for adaptive spectrum management. Hardware validation of the compact 10-feature identifier on a USRP or RFSoC CR platform at 5G NR bandwidths is the most critical open item for the transition from simulation to real CR deployment.

References

[1] O.A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of Automatic Modulation Classification Techniques: Classical Approaches and New Trends", *IET Communications*, vol. 1, pp. 137–156, 2007 (https://doi.org/10.1049/iet-com:20050176).

[2] P.C. Sahu *et al.*, "Cloud-enabled Automatic Modulation Classification Using Deep Feature Fusion and Moth-flame Optimized Elm Approach", *Scientific Reports*, vol. 16, art. no. 1061, 2026 (https://doi.org/10.1038/s41598-025-30753-4).

[3] S. Chen and J. Zhao, "The Requirements, Challenges, and Technologies for 5G of Terrestrial Mobile Telecommunication", *IEEE Communications Magazine*, vol. 52, pp. 36–43, 2014 (https://doi.org/10.1109/MCOM.2014.6815891).

[4] P. Li, J. Fan, and J. Wu, "Exploring the Key Technologies and Applications of 6G Wireless Communication Network", *iScience*, vol. 28, art. no. 112281, 2025 (https://doi.org/10.1016/j.isci.2025.112281).

[5] W. Saad, M. Bennis, and M. Chen, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems", *IEEE Network*, vol. 34, pp. 134–142, 2020 (https://doi.org/10.1109/MNET.001.1900287).

[6] W. Jiang, B. Han, M.A. Habibi, and H.D. Schotten, "The Road Towards 6G: A Comprehensive Survey", *IEEE Open Journal of the Communications Society*, vol. 2, pp. 334–366, 2021 (https://doi.org/10.1109/OJCOMS.2021.3057679).

[7] L. Zhang *et al.*, "Subband Filtered Multi-carrier Systems for Multi-service Wireless Communications", *IEEE Transactions on Wireless Communications*, vol. 16, pp. 1893–1907, 2017 (https://doi.org/10.1109/TWC.2017.2656904).

[8] R. Hadani *et al.*, "Orthogonal Time Frequency Space Modulation", *IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, USA, 2017 (https://doi.org/10.1109/WCNC.2017.7925924).

[9] M. Wen, X. Cheng, and L. Yang, *Index Modulation for 5G Wireless Communications*, Springer, 164 p., 2017 (https://doi.org/10.1007/978-3-319-51355-3).

[10] H.B. Chikha, A. Alaerjan, and R. Jabeur, "Automatic Classification of 5G Waveform-modulated Signals Using Deep Residual Networks", *Sensors*, vol. 25, art. no. 4682, 2025 (https://doi.org/10.3390/s25154682).

[11] 3GPP, "Study on Channel Model for Frequencies from 0.5 to 100 GHz", Technical Report TR 38.901, 3rd Generation Partnership Project, 2022.

[12] W.C. Jakes (ed.), *Microwave Mobile Communications*, Wiley-IEEE Press, 656 p., 1994 (ISBN: 9780780310698).

[13] N. Michailow *et al.*, "Generalized Frequency Division Multiplexing for 5th Generation Cellular Networks", *IEEE Transactions on Communications*, vol. 62, pp. 3045–3061, 2014 (https://doi.org/10.1109/TCOMM.2014.2345566).

[14] M. Matthe, N. Michailow, I. Gaspar, and G. Fettweis, "Influence of Pulse Shaping on Bit Error Rate Performance and out of Band Radiation of Generalized Frequency Division Multiplexing", *2014 IEEE International Conference on Communications Workshops (ICC)*, Sydney, Australia, 2014 (https://doi.org/10.1109/ICCW.2014.6881170).

[15] M. Bellanger *et al.*, "FBMC Physical Layer: A Primer", PHYDYAS, 2010.

[16] M. Besseghier, A.B. Djebbar, and E. Kofidis, "Joint CFO and Highly Frequency Selective Channel Estimation in FBMC/OQAM Systems", *Digital Signal Processing*, vol. 128, art. no. 103629, 2022 (https://doi.org/10.1016/j.dsp.2022.103629).

[17] Y. Liu *et al.*, "Nonorthogonal Multiple Access for 5G and Beyond", *Proceedings of the IEEE*, vol. 105, pp. 2347–2381, 2017 (https://doi.org/10.1109/JPROC.2017.2768666).

[18] Z. Ding *et al.*, "Application of Non-orthogonal Multiple Access in LTE and 5G Networks", *IEEE Communications Magazine*, vol. 55, pp. 185–191, 2017 (https://doi.org/10.1109/MCOM.2017.1500657CM).

[19] M. Besseghier, A. Zouggaret, S. Ghouali, and A.B. Djebbar, "Performance Degradation Analysis of Downlink NOMA Systems Under Carrier Frequency Offsets", *IEEE Transactions on Consumer Electronics*, vol. 71, pp. 2508–2516, 2025 (https://doi.org/10.1109/TCE.2025.3571676).

[20] E. Basar, U. Aygolu, E. Panayirci, and H.V. Poor, "Orthogonal Frequency Division Multiplexing with Index Modulation", *IEEE Transactions on Signal Processing*, vol. 61, pp. 5536–5549, 2013 (https://doi.org/10.1109/TSP.2013.2279771).

[21] E. Basar *et al.*, "Index Modulation Techniques for Next-generation Wireless Networks", *IEEE Access*, vol. 5, pp. 16693–16746, 2017 (https://doi.org/10.1109/ACCESS.2017.2737528).

[22] M. Besseghier, S. Ghouali, and A.B. Djebbar, "Optimized Greedy Detection for OFDM-IM Systems", *IEEE Communications Letters*, vol. 27, pp. 2034–2037, 2023 (https://doi.org/10.1109/LCOMM.2023.3283032).

[23] P. Raviteja, K.T. Phan, Y. Hong, and E. Viterbo, "Interference Cancellation and Iterative Detection for Orthogonal Time Frequency Space Modulation", *IEEE Transactions on Wireless Communications*, vol. 17, pp. 6501–6515, 2018 (https://doi.org/10.1109/TWC.2018.2860011).

[24] G.D. Surabhi, R.M. Augustine, and A. Chockalingam, "On the Diversity of Uncoded OTFS Modulation in Doubly-dispersive Channels", *IEEE Transactions on Wireless Communications*, vol. 18, pp. 3049–3063, 2019 (https://doi.org/10.1109/TWC.2019.2909205).

[25] G.D. Surabhi and A. Chockalingam, "Low-complexity Linear Equalization for OTFS Modulation", *IEEE Communications Letters*, vol. 24, pp. 330–334, 2020 (https://doi.org/10.1109/LCOMM.2019.2956709).

[26] Z. Wei *et al.*, "Orthogonal Time-frequency Space Modulation: A Promising Next-generation Waveform", *IEEE Wireless Communications*, vol. 28, pp. 136–144, 2021 (https://doi.org/10.1109/MWC.001.2000408).

[27] A. Bemani, N. Ksairi, and M. Kountouris, "Affine Frequency Division Multiplexing for Next Generation Wireless Communications", *IEEE Transactions on Wireless Communications*, vol. 22, pp. 8214–8229, 2023 (https://doi.org/10.1109/TWC.2023.3260906).

[28] A. Bouguila, A. Khlifi, and M. Yahia, "Performance Analysis of OFDM, OTFS, and AFDM Modulation Over Doubly Selec-

tive Channels", *2025 IEEE International Multi-Conference on Smart Systems & Green Process (IMC-SSGP)*, Hammamet, Tunisia, 2025, (https://doi.org/10.1109/IMC-SSGP67001.2025.11473997).

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 800 p., 2016 (https://doi.org/10.1007/s10710-017-9314-z).

[30] Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages", *Psychometrika*, vol. 12, pp. 153–157, 1947 (https://doi.org/10.1007/BF02295996).

[31] E.B. Wilson, "Probable Inference, the Law of Succession, and Statistical Inference", *Journal of the American Statistical Association*, vol. 22, pp. 209–212, 1927 (https://doi.org/10.2307/2276774).

[32] A. Swami and B.M. Sadler, "Hierarchical Digital Modulation Classification Using Cumulants", *IEEE Transactions on Communications*, vol. 48, pp. 416–429, 2000 (https://doi.org/10.1109/26.837045).

[33] Z. Zhang *et al.*, "Automatic Modulation Classification Using CNN-LSTM Based Dual-stream Structure", *IEEE Transactions on Vehicular Technology*, vol. 69, pp. 13521–13531, 2020 (https://doi.org/10.1109/TVT.2020.3030018).

[34] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic Modulation Classification: A Deep Learning Enabled Approach", *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 10760–10772, 2018 (https://doi.org/10.1109/TVT.2018.2868698).

Abdelkader Horch, Ph.D.
https://orcid.org/0009-0001-4382-4996
E-mail: a.horch@univ-mascara.dz
University of Mustapha Stambouli, Mascara, Algeria
https://www.univ-mascara.dz

Mokhtar Besseghier, Ph.D.
https://orcid.org/0000-0003-3522-1332
E-mail: m.besseghier@univ-mascara.dz
University of Mustapha Stambouli, Mascara, Algeria
https://www.univ-mascara.dz

Samir Ghouali, Ph.D.
https://orcid.org/0000-0002-3653-873X
E-mail: s.ghouali@univ-mascara.dz
University of Mustapha Stambouli, Mascara, Algeria
https://www.univ-mascara.dz

Abderrahmane Louni, Ph.D.
https://orcid.org/0009-0007-6401-4363
E-mail: a.louni@univ-mascara.dz
University of Mustapha Stambouli, Mascara, Algeria
https://www.univ-mascara.dz

Communication-efficient Embedded FFT Processing for Acoustic Telemetry in LPWAN-based Beehive Monitoring Systems

Sebastian Górecki

Lodz University of Technology, Łódź, Poland

https://doi.org/10.26636/jtit.2026.3.2678

Submitted: 21 Jun. 2026, Accepted: 6 Jul. 2026

Abstract — Low-power acoustic telemetry remains a significant challenge in Internet of Things monitoring systems deployed in remote environments. This paper presents an embedded fast Fourier transform framework for communication-efficient acoustic monitoring of honey bee colonies. Instead of transmitting raw audio streams, the proposed approach extracts compact spectral descriptors directly on an embedded sensing node and transmits only a small feature vector using a low-power cellular network. The framework is evaluated using labeled queenright and queenless colony recordings. The proposed solution targets resource-constrained ESP32 class IoT nodes operating over LTE-M and NB-IoT networks. The analysis covered such parameters as dominant frequency, peak amplitude, mean spectral amplitude, spectral centroid, spectral entropy, and band energy extracted from the 200 – 400 Hz band. The results showed that dominant frequency alone did not significantly differentiate colony states at the file level, while mean spectral amplitude remained statistically significant. Queenless recordings also exhibited higher dominant frequency variability. The proposed approach reduces the transmitted payload by more than three orders of magnitude while remaining compatible with resource-constrained ESP32 class IoT devices. The results demonstrate that the extraction of embedded acoustic features is a practical method for scalable smart beehive monitoring under strict memory, power, and bandwidth constraints.

Keywords — acoustic telemetry, edge computing, Fast Fourier Transform, Internet of Things, wireless sensor networks

1. Introduction

Acoustic sensing is increasingly used in Internet of Things (IoT) monitoring systems operating in remote, battery-powered, and bandwidth-constrained environments. In such systems, continuous audio acquisition can provide useful diagnostic information, but raw audio transmission is often impractical due to memory usage, energy consumption, and communication cost. Unlike most previous studies focused on biological interpretation, this work addresses the challenge of acoustic data reduction faced in telecommunications settings, to facilitate low power IoT deployments. Unlike cloud-centric acoustic monitoring systems, the proposed framework explicitly prioritizes communication efficiency and embedded feasibility over high-capacity biological classification.

Honey bee colonies represent a particularly interesting application case for distributed acoustic telemetry. The colonies generate sound and vibration patterns associated with thermoregulation, foraging activity, queen presence, preparation of swarms, and colony stress [1]–[3]. Smart beehive systems commonly integrate temperature, humidity, weight, and acoustic detection to support precision apiculture and non-invasive colony monitoring [4]–[6].

Recent studies have highlighted the importance of energy-efficient sensing and edge intelligence in precision beekeeping systems deployed in remote environments [7]–[9]. Additionally, surveys indicate that acoustic sensing is becoming one of the key modalities (in addition to temperature, humidity and hive weight monitoring) relied upon in precision beekeeping platforms [10].

Works published in the telecommunications literature have highlighted the growing importance of edge intelligence and lightweight signal processing in acoustic wireless sensor networks [11], [12]. From the telecommunications and information technology perspective, the key challenge consists not only in biological interpretation of acoustic signals, but also in efficient transmission of relevant information. The process of sending raw audio from each hive is difficult to scale, especially when devices operate over LTE-M, NB-IoT, or other low-power wireless communication technologies. A more efficient solution is to perform local signal processing on the embedded node and transmit only compact descriptors extracted from the frequency domain. Low-power wide-area communication technologies such as NB-IoT and LTE-M have become increasingly important for remote sensing deployments, where bandwidth and energy consumption must be minimized [7].

The reliability of LPWAN communications can be affected by interference and coverage limitations, which makes communication-efficient telemetry particularly important in remote deployments [13].

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

Tab. 1. Representative acoustic frequency ranges reported in the literature dealing with honey bee monitoring systems.

Condition / behavior	Frequency range	Interpretation	References
Ventilation and thermoregulation	50 – 190 Hz	Hive ventilation and microclimate regulation	[3]
Queen-related activity	160 – 280 Hz	Acoustic patterns associated with queen presence, queen acceptance, or queenless state	[16]
Foraging and nectar processing	200 – 300 Hz	Active colony work, nectar handling, and foraging related activity	[3], [17]
Intensive colony activity	240 – 400 Hz	Increased colony activity and internal hive logistics	[5]
Pre-swarming activity	400 – 500 Hz	Increased excitation and swarm preparation	[1], [14]
Swarm-related signals	500 – 550 Hz	Acoustic patterns associated with swarm mobilization	[14]

This paper presents a communication-efficient embedded fast Fourier transform (FFT) framework for acoustic telemetry in smart beehive monitoring systems. The main objective is not to develop a production-ready biological classifier, but to evaluate whether compact FFT-derived descriptors retain diagnostically relevant information while substantially reducing communication payload.

The main contributions of this work are as follows:

1) embedded FFT-based acoustic feature extraction framework for smart beehive IoT nodes,
2) communication reduction analysis comparing raw audio transmission with compact acoustic telemetry,
3) experimental evaluation using queenright and queenless honey bee colony recordings,
4) comparison of dominant-frequency, amplitude-based, and spectral-shape descriptors,
5) file-level statistical analysis mitigating the risk of overinterpreting non-independent audio segments.

The rest of the paper is structured as follows. Section 2 reviews related works. Section 3 presents the proposed embedded acoustic telemetry framework. Section 4 describes the FFT-based feature extraction method. Section 5 introduces the experimental data set. Section 6 reports the results, while Section 7 discusses limitations and deployment-related implications. Section 8 concludes the paper.

2. Related Works

Precision apiculture increasingly relies on non-invasive sensing systems that observe colony dynamics without opening the hive. Existing systems measure internal temperature, humidity, hive weight, and acoustic activity, often transmitting the collected telemetry to a cloud platform for analysis [4]–[6].

Acoustic monitoring has received significant attention, as bee-generated sounds can reflect colony activity, swarm preparation, queen status, and stress responses [1], [3]. The authors of [1] demonstrated that the spectral characteristics of the hive sounds can be used for early swarm detection, while paper [2] stated that the vibrational signatures contain biologically meaningful information on colony dynamics. Recent studies have also explored embedded or machine learning-based approaches to bee sound analysis [8], [14].

A comprehensive review of automated beehive acoustic monitoring was recently presented in [15], highlighting the growing interest in signal processing and machine learning approaches.

However, many acoustic monitoring approaches focus primarily on biological classification or offline analysis. Less attention has been paid to the communication and embedded system aspects of large-scale acoustic telemetry. In practical IoT deployments, transmitting raw audio is often not feasible. Therefore, edge-based signal processing is needed to reduce data volume while preserving diagnostically useful information.

Table 1 summarizes the representative frequency ranges reported in previous beehive acoustic studies. These ranges should be interpreted as indicative rather than universal diagnostic thresholds, because measured values may depend on colony condition, microphone placement, hive type, environmental conditions, and sensor characteristics.

Although acoustic monitoring has been extensively investigated, most studies focused on biological interpretation or offline signal analysis. Only limited attention has been paid to embedded implementation constraints and reduction in communications payloads. A comparison of representative approaches is presented in Tab. 2.

As shown in Tab. 2, most previous studies focused primarily on the biological interpretation of hive acoustics, while the proposed approach explicitly addresses embedded implementation constraints and communication reduction.

Tab. 2. Comparison of acoustic monitoring approaches reported in the literature.

Ref.	FFT	Edge	Embedded	Comm.
[1]	Yes	No	No	No
[2]	Yes	No	No	No
[14]	Partial	Partial	No	No
This work	Yes	Yes	Yes	Yes



Fig. 1. Acoustic sensing hardware used in smart beehive monitoring systems: waterproof microphone module (left) and microphone installed at the hive entrance during field deployment (right).

3. Proposed Embedded Acoustic Telemetry Framework

The proposed framework is intended for smart beehive monitoring systems equipped with a microphone, an embedded microcontroller, and a wireless communication module. The system acquires short acoustic buffers, performs local FFT analysis, extracts compact descriptors, and transmits only the resulting feature vector.

Figure 1 shows the acoustic sensing hardware used in the BeeHUB monitoring system. The microphone may be installed near the entrance to or inside the hive, depending on the monitoring objective and deployment constraints.

The conceptual architecture of the proposed acoustic telemetry chain is shown in Fig. 2. Acoustic signals are processed locally on the embedded node. Only extracted descriptors, such as amplitude, energy, centroid, entropy, and dominant frequency, are transmitted to the cloud platform.

The embedded processing workflow is presented in Fig. 3. The device periodically initializes the microphone, acquires a short buffer, computes the FFT, extracts descriptors from the selected frequency band, transmits the compact packet, and returns to sleep mode.

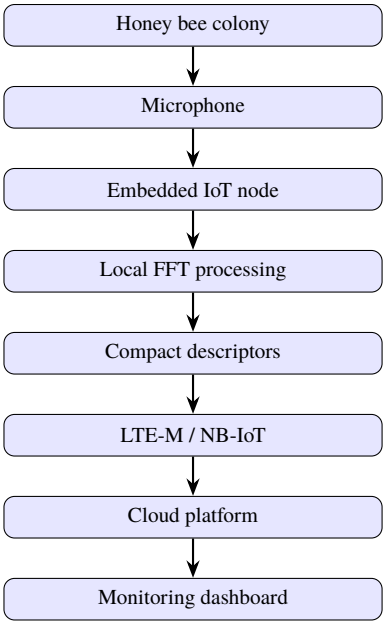


Fig. 2. Architecture of the proposed embedded acoustic telemetry framework.

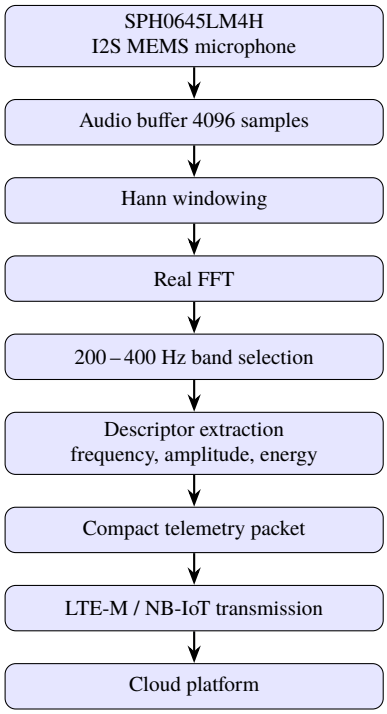


Fig. 3. Signal processing pipeline of the proposed integrated acoustic telemetry framework.

4. FFT-based Feature Extraction Method

4.1. Signal Processing

Fast Fourier Transform (FFT) remains one of the most widely used methods for frequency domain analysis in embedded signal processing systems due to its computational efficiency and suitability for resource-constrained hardware [18]–[20].

For each audio segment, a real-valued FFT was calculated. The discrete Fourier transform of the input signal is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \tag{1}$$

where $x(n)$ is the time-domain acoustic signal, N is the number of samples, and $X(k)$ is the complex frequency-domain representation.

The frequency resolution is given by:

$$\Delta f = \frac{f_s}{N}, \tag{2}$$

where f_s is the sampling frequency. For the target embedded configuration with $f_s = 8192$ Hz and $N = 4096$, the frequency resolution is 2 Hz.

The FFT window duration is:

$$T_w = \frac{N}{f_s}. \tag{3}$$

For $N = 4096$ and $f_s = 8192$ Hz, one FFT window corresponds to 0.5 s of audio. Multiple short windows may be analyzed within one measurement cycle to observe a longer acoustic interval without storing a full raw audio recording.

Prior to FFT computation, a Hann window was applied to each audio segment to reduce spectral leakage resulting from finite-length signal acquisition.

4.2. Extracted Descriptors

The analysis focused on the 200 – 400 Hz frequency band. This band was selected because it overlaps with the acoustic activity ranges reported for foraging, nectar processing, colony work and selected queen- or stress-related states.

The dominant frequency was calculated as follows:

$$f_{dom} = f\left(\arg \max_{f\in[f_{min},f_{max}]} \left|X(f)\right|\right) . \tag{4}$$

In addition to dominant frequency, the following descriptors were evaluated:

1) peak amplitude,
2) mean spectral amplitude,
3) spectral centroid,
4) spectral entropy,
5) band energy.

The spectral centroid was calculated as:

$$C = \frac{\sum\limits_i f_i A_i}{\sum\limits_i A_i} , \tag{5}$$

where f_i is the frequency bin and A_i is the corresponding spectral magnitude.

The normalized spectral entropy was calculated from the probability distribution of spectral magnitudes:

$$H = -\sum_i p_i \log_2(p_i) , \tag{6}$$

where

$$p_i = \frac{A_i}{\sum\limits_i A_i} .$$

Band energy was calculated as:

$$E = \sum_i A_i^2 . \tag{7}$$

The FFT stage dominates the computational complexity of the proposed pipeline with complexity $O(N \log N)$, whereas the remaining descriptors are computed in linear time. Owing to this computational profile, real-time execution remains feasible on modern microcontrollers such as ESP32 [18], [19].

4.3. Embedded Implementation Parameters and Benchmark

Table 3 summarizes the target embedded implementation parameters used in the proposed FFT-based acoustic telemetry framework. The selected configuration is compatible with ESP32-class IoT nodes and avoids raw audio transmission by extracting compact spectral descriptors locally.

To address the feasibility of the proposed FFT-based processing pipeline on the target hardware platform, an embedded benchmark was designed for the ESP32-class microcontroller used in the BeeHUB acoustic sensing node. The benchmark measures the execution time of one FFT processing cycle, determines the maximum memory usage, the size of the descriptor payload, and the communication reduction achieved by local feature extraction.

Tab. 3. Implementation parameters and resource requirements.

Parameter	Value
Microcontroller class	ESP32-based IoT node
Target sampling frequency	8192 Hz
FFT length	4096 samples
Frequency resolution	2 Hz
Window duration	0.5 s
Analysis duration	12 s
Number of FFT windows	24
Analysis band	200 – 400 Hz
Candidate features	Frequency, amplitude, energy, centroid, entropy
Input audio buffer	8 KB
FFT working buffer	8 KB
Typical feature vector	8 – 24 bytes
Estimated RAM requirement	$\leqslant$ 18 KB
Estimated firmware Flash	$\leqslant$ 100 KB
Wireless technology	LTE-M / NB-IoT

The benchmark procedure consists in acquiring 4096 sample audio windows at 8192 Hz, applying FFT-based spectral analysis, extracting compact descriptors from the 200 – 400 Hz band, and preparing the resulting telemetry payload for wireless transmission.

The same processing configuration as in the offline analysis is used to ensure consistency between the experimental validation and the target embedded implementation. The benchmark reused the original implementation of the production firmware without modifying the FFT processing path, ensuring that the reported execution times accurately reflect the deployed BeeHUB sensing node.

The firmware-level benchmark was performed using the FFT implementation of the ESP32 acoustic pipeline. The benchmark compiled the production firmware FFT implementation in a host side test harness and reproduced the processing structure used by the embedded acoustic acquisition routine.

The measured FFT runtime was 0.017 ms per a 4096-sample window, while the total processing time including descriptor extraction was 0.227 ms. The estimated peak heap usage was 98 344 bytes, and the FFT object Flash footprint was 4578 bytes. The measured processing time of 0.227 ms represents less than 0.05% of the 0.5-second acquisition window, confirming that the proposed processing pipeline easily satisfies real-time constraints on the target embedded platform (see Tab. 4).

Consequently, the benchmark evaluates the identical FFT implementation and processing path used by the production BeeHUB firmware, although timing measurements were obtained using a host side benchmarking harness. The measured peak heap usage remains well below the available RAM of typical ESP32 devices, confirming that the proposed implementation is compatible with low-cost embedded hardware.

Tab. 4. Firmware-level FFT benchmark and payload reduction results.

Metric	Value
Benchmark type	Host-side firmware benchmark
FFT implementation	ESP32 firmware FFT code
FFT size	4096 samples
FFT runtime	0.017 ms
Total processing time	0.227 ms
Peak heap usage	98 344 bytes
Measured FFT module Flash footprint	4 578 bytes
Single descriptor packet payload	4 bytes
Raw I2S buffer payload	32 768 bytes
Reduction factor	8192×
Estimated LTE-M airtime	0.085 ms vs. 699.1 ms
Estimated energy reduction	8192×
Measured board sleep current	62.25 µA
Processing utilization	0.045% of acquisition window

Tab. 5. Experimental audio dataset and recording characteristics.

Parameter	Value
Queenright recordings	6
Queenless recordings	8
Total recordings	14
Queenright segments	36
Queenless segments	48
Total analyzed segments	84
Original recording duration	829 s
Analyzed duration	168 s
Sampling rate	44.1 kHz
Resampled analysis rate	8192 Hz
Segment duration	2 s
Analysis window per file	12 s

The raw I2S buffer used by the embedded implementation contains 4096 frames with two 32-bit channels, corresponding to 32 768 bytes per FFT window. In contrast, the minimal transmitted acoustic descriptor packet consists of a 16-bit frequency value and a 16-bit amplitude value, corresponding to 4 B. In the extended feature-vector configuration used for communication analysis, six 32-bit descriptors correspond to 24 bytes. This corresponds to an 8192-fold reduction in the transmitted payload. Under a simplified LTE-M payload-only airtime model at 375 kbps, this corresponds to a reduction from approximately 699.1 ms for the raw buffer to 0.085 ms for the descriptor payload.

The measured embedded processing time of 0.227 ms occupies only 0.045% of the 0.5 s acquisition interval and remains more than three orders of magnitude lower than the estimated LTE-M transmission time for raw audio (699.1 ms). This confirms that communication latency dominates the overall sensing cycle.

In addition, a board-level measurement of the low power state performed with all sensors connected and without the programmer attached showed a sleep current of approximately 62.25 µA. This value characterizes the quiescent operation of the sensing node between measurement cycles and should not be confused with the average active current reported for the complete sensing cycle power profile in Section 6.

5. Experimental Dataset and Statistical Analysis

The experimental data set consisted of WAV recordings obtained from honey bee colonies labeled as queenright and queenless. The recordings were originally collected as audio files and processed offline in Python using NumPy and SciPy. Files beginning with `q` were treated as queenright recordings, whereas those beginning with `nq` were treated as queenless recordings.

5.1. Audio Preprocessing

Before feature extraction, all recordings were processed using a standardized preprocessing workflow to improve comparability between measurements collected under field conditions. The audio recordings were converted to uncompressed WAV format and analyzed using a common processing pipeline implemented in Python.

The original recordings were acquired at a sampling frequency of 44.1 kHz. To ensure consistency with the intended embedded implementation, all recordings were resampled to the target sampling frequency of 8192 Hz prior to feature extraction. This resampling configuration matches the proposed ESP32-based deployment and provides a frequency resolution of 2 Hz for a 4096-point FFT. The resampled signals were subsequently divided into fixed-length 2-second segments for descriptor extraction (Tab. 5).

The target embedded implementation uses a SPH0645LM4H digital MEMS microphone with an I2S interface [21]. According to the manufacturer's specification, the microphone operates over a frequency range of approximately 50 to 15 kHz and supports a supply voltage of 1.6 to 3.6 V. The digital I2S interface eliminates analog gain-stage variability, although microphone placement and hive-specific acoustic coupling may still influence the measured amplitude.

To reduce the influence of recording-level amplitude differences caused by microphone placement or acquisition conditions, amplitude normalization was applied before FFT analysis. Normalization was performed independently for each analyzed segment while preserving the relative spectral characteristics within the selected frequency band.

Each segment was subsequently windowed using a Hann window to reduce spectral leakage before FFT computation. The frequency domain descriptors were then extracted from the 200 – 400 Hz band, including the dominant frequency,

Tab. 6. Effect of segment duration on extracted acoustic descriptors.

State	Duration	Mean amplitude	Dominant frequency
Queenright	2 s	115508.73	248.88 Hz
Queenright	12 s	128201.93	250.88 Hz
Queenright	30 s	122356.53	251.50 Hz
Queenless	2 s	425703.50	271.08 Hz
Queenless	12 s	1967941.00	250.80 Hz
Queenless	30 s	2066559.00	224.00 Hz

the peak amplitude, the mean spectral amplitude, spectral centroid, the spectral entropy, and the band energy.

Because the recordings originated from practical field deployments rather than controlled laboratory measurements, absolute amplitude values should be interpreted with caution. Consequently, amplitude-based descriptors are considered relative diagnostic features rather than calibrated sound-pressure measurements.

In general, the analyzed dataset comprised 14 recordings, including six queenright and eight queenless colonies. Each file was analyzed over the first 12 s and divided into six 2 second segments. This resulted in 36 queenright and 48 queenless segments, for a total of 84 analyzed segments and 168 s of analyzed audio. The original recordings were sampled at 44.1 kHz.

Because segments extracted from the same recording are not fully independent observations, two levels of statistical analysis were performed. First, segment-level analysis was used to examine the behavior of the descriptor in all 84 short windows. Second, file-level aggregation was performed by averaging descriptors within each audio recording, resulting in 14 independent file-level observations.

The Mann-Whitney U test was used for non-parametric comparison between queenright and queenless groups. The effect size was quantified using Cliff's delta. Bootstrap 95% confidence intervals were calculated for the amplitude descriptors to provide uncertainty estimates. Because multiple acoustic descriptors were evaluated, the statistical significance of the obtained p-values was additionally interpreted using the Bonferroni correction.

6. Results

6.1. Example FFT Spectrum

Figure 4 presents an example of the FFT spectrum extracted from a beehive recording. The highlighted 200 – 400 Hz region was used for the extraction of the descriptor.

6.2. Segment Duration Analysis

Table 6 summarizes the effect of the duration of the segment on the extracted descriptors. Queenright recordings showed relatively stable dominant frequency estimates across segment durations, while queenless recordings exhibited greater variability.

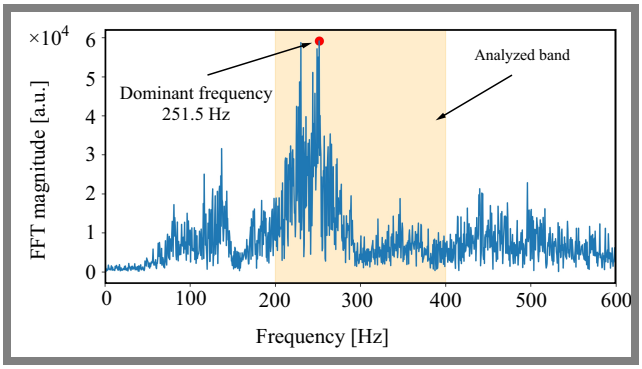


Fig. 4. Example FFT spectrum extracted from a beehive recording. The 200 – 400 Hz band was used for descriptor extraction.

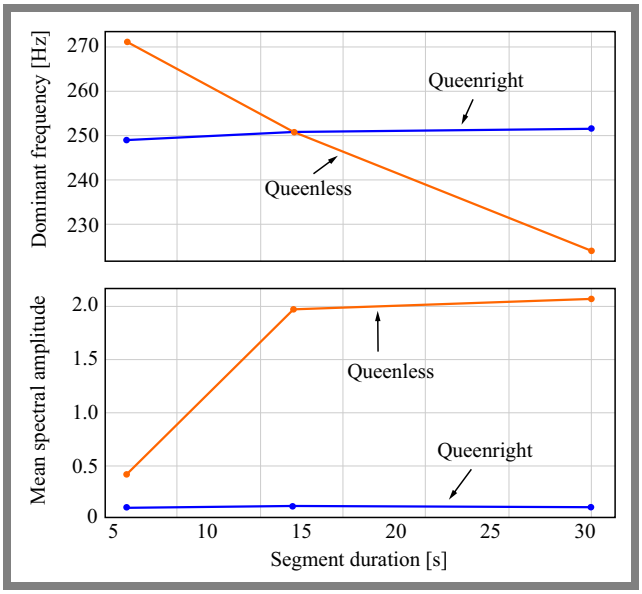


Fig. 5. Influence of segment duration on dominant frequency and mean spectral amplitude descriptors.

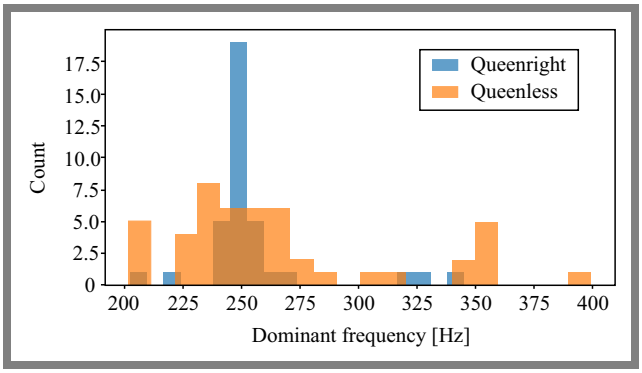


Fig. 6. Distribution of dominant frequencies extracted from queenright and queenless recordings in the 200 – 400 Hz band.

6.3. Dominant Frequency and Mean Amplitude

The extracted dominant frequencies were compared between queenright and queenless recordings. At the segment level, the dominant frequency did not significantly differentiate the groups. However, the amplitude-based descriptors showed clearer separation.

Tab. 7. Segment-level comparison of FFT-derived descriptors between queenright and queenless recordings.

Descriptor	Queenright mean	Queenless mean	Queenright SD	Queenless SD	p	δ
Dominant freq. [Hz]	254.76	265.93	26.00	48.98	0.674	–0.054
Peak amplitude	7.50×10^5	1.44×10^6	2.54×10^5	3.39×10^6	2.92×10^{-8}	0.711
Mean amplitude	1.38×10^5	3.09×10^5	4.43×10^4	6.57×10^5	1.02×10^{-8}	0.734
Spectral centroid [Hz]	273.82	293.15	9.58	16.40	5.40×10^{-8}	–0.697
Spectral entropy	8.17	8.28	0.13	0.24	8.87×10^{-6}	–0.569
Band energy	1.45×10^{13}	2.98×10^{14}	9.00×10^{12}	8.15×10^{14}	7.84×10^{-9}	0.740

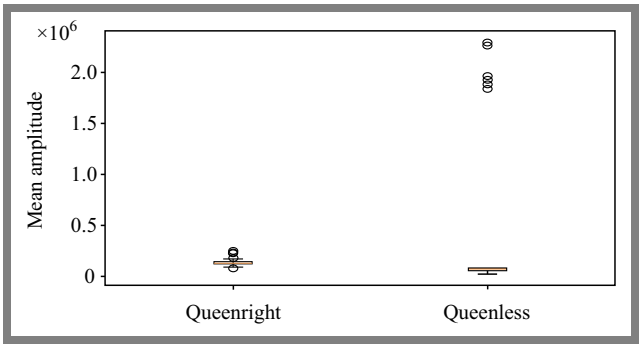


Fig. 7. Comparison of the mean spectral amplitude in the 200 – 400 Hz band for queenright and queenless recordings.

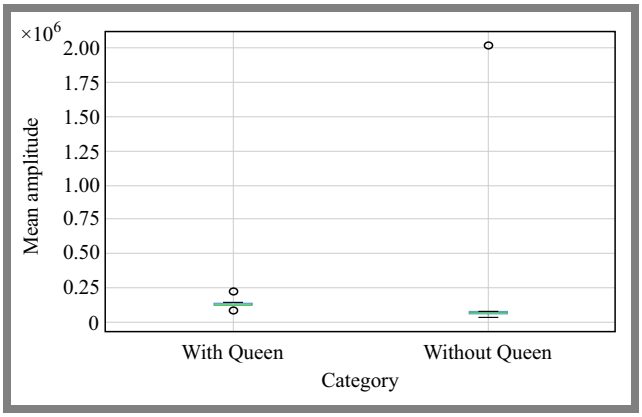


Fig. 8. File-level comparison of mean spectral amplitude. Each point represents an independent audio recording.

6.4. Extended FFT-derived Descriptors

Table 7 summarizes the segment-level statistical comparison of the FFT-derived descriptors. The dominant frequency was not statistically significant, while peak amplitude, mean amplitude, spectral centroid, spectral entropy, and band energy differed significantly between the two groups.

The strongest segment-level effect sizes were observed for band energy, mean amplitude, and peak amplitude. These results indicate that energy-related and spectral-shaped descriptors contain substantially more diagnostic information than dominant frequency alone.

The file-level effect size for mean spectral amplitude was large (Cliff's $\delta = 0.75$), indicating a substantial practical difference between queenright and queenless recordings despite the limited dataset size.

6.5. File-level Analysis

To reduce the risk of overinterpreting dependent segments, the descriptors were also aggregated at file level. Table 8 summarizes the statistical comparison.

At the file level, the mean dominant frequency was 254.76 Hz for queenright recordings and 265.93 Hz for their queenless counterparts. This difference was not statistically significant ($p = 0.216$). In contrast, mean spectral amplitude differed significantly between the groups. Queenright recordings had a mean amplitude of 138262, with a bootstrap 95% confidence interval of [124661, 153062], whereas queenless recordings had a mean amplitude of 308576, with a bootstrap 95% confidence interval of [141550, 508733] ($p = 0.0236$, Cliff's $\delta = 0.750$).

Tab. 8. File-level statistical comparison of acoustic descriptors.

Feature	p-value	Cliff's δ	Result
Dominant frequency	0.216	–0.417	n.s.
Mean amplitude	0.0236	0.750	Significant

6.6. Frequency Stability

In addition to mean dominant frequency, stability was evaluated using the coefficient of variation (CV). Queenright recordings exhibited a CV of 3.75%, whereas queenless recordings showed considerably higher variability (CV of 6.97%). This suggests that frequency stability may provide additional diagnostic information beyond the mean dominant frequency alone.

6.7. Communication and Energy Efficiency

A major advantage of embedded acoustic processing is the reduction of communication load. The transmission of raw audio requires substantially more data than the transmission of compact FFT-derived descriptors. For mono-audio sampled at $f_s = 8192$ Hz with 16-bit resolution over an analysis duration of $T = 12$ s, the raw payload size is:

$$S_{raw} = f_s \cdot T \cdot \frac{b}{8} ,$$

where b is the number of bits per sample. This gives:

$$S_{raw} = 8\,192 \cdot 12 \cdot 2 = 196\,608 \text{ bytes} .$$

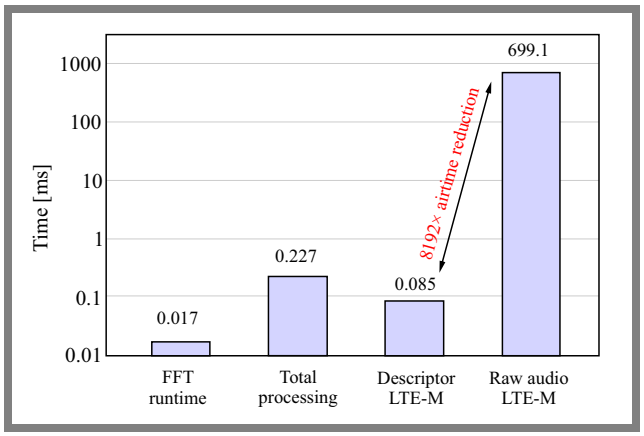


Fig. 9. Comparison of embedded processing latency and estimated LTE-M transmission time, illustrating the communication advantage of local FFT processing.

In contrast, the proposed embedded approach transmits only a small descriptor vector containing frequency and amplitude domain features. If each descriptor is encoded as a 32-bit floating-point value and six descriptors are transmitted, the resulting payload is:

$$S_{desc} = 6 \cdot 4 = 24 \text{ B} .$$

Therefore, the payload reduction ratio is:

$$R = \frac{S_{raw}}{S_{desc}} .$$

For the analyzed configuration, this corresponds to a reduction ratio of approximately 8 192:1.

Figure 9 shows that the execution time for local FFT processing is several orders of magnitude shorter than the time required to transmit raw audio over LTE-M. The total embedded processing time of 0.227 ms remains more than three orders of magnitude lower than the estimated LTE-M transmission time for raw audio of 699.1 ms. Consequently, communication latency dominates the overall sensing cycle, confirming that edge-based spectral feature extraction is substantially more efficient than raw audio transmission.

The substantial reduction in the transmitted payload directly translates into shorter radio activity periods. To complement this theoretical communication analysis, the complete sensing node was experimentally profiled using a Nordic Power Profiler Kit II.

The proposed approach shifts the computational burden from wireless communication to local signal processing, which is advantageous for battery-powered LPWAN deployments. By reducing the transmitted payload from 196 608 to 8 – 24 bytes, the proposed framework substantially shortens radio activity duration and decreases communication overhead while preserving diagnostically relevant acoustic information (Tab. 9). Such communication-efficient processing is consistent with the growing trend towards edge intelligence in IoT systems.

Tab. 9. Communication payload reduction achieved by embedded FFT feature extraction.

Representation	Payload size	Relative load
Raw audio, 12 s, 8192 Hz, 16 bit	196 608 bytes	1.0
2 descriptors	8 bytes	4.07×10^{-5}
6 descriptors	24 bytes	1.22×10^{-4}
8 descriptors	32 bytes	1.63×10^{-4}

Tab. 10. Experimental power consumption of the complete sensing node.

Metric	Value
Supply voltage	3.6 V
Average current	93.99 mA
Peak current	1.16 A
Measurement duration	24.54 s
Measured charge	2.31 C
Average power	338.4 mW
Energy per sensing cycle	8.31 J
Measurement scope	Complete node with all sensors connected



Fig. 10. Example of a field observation conducted in an apiary near Konin, Poland, during a period of elevated colony activity preceding a swarming event.

Power profiling was performed using a Nordic Power Profiler Kit II at a supply voltage of 3.6 V. The measurement covered one complete sensing cycle with all connected sensors, including sensor initialization, FFT processing, wireless modem activity, and return to idle state. The average current was 93.99 mA for 24.54 s, corresponding to a measured charge of 2.31 C and an energy consumption of approximately 8.31 J per cycle. The peak current reached 1.16 A during short periods of high activity, which is consistent with wireless modem activity and sensor initialization (Tab. 10).

6.8. Field Observation Example

Swarming is one of the most important colony states that motivates automated acoustic monitoring. Figure 10 presents a field observation of increased colony activity during a pre-swarming period.

6.9. Multi-modal Hive Observation

Figures 11 and 12 present an example of multimodal hive monitoring using the BeeHUB telemetry platform.

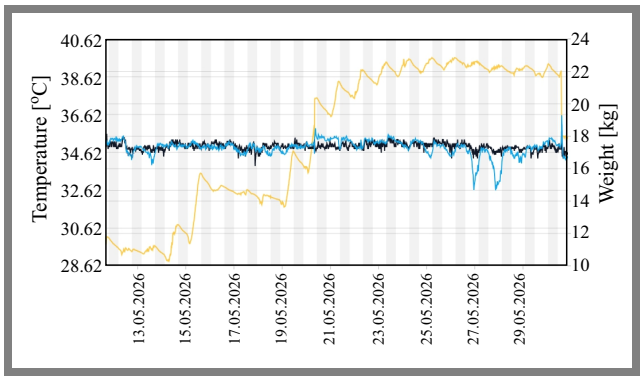


Fig. 11. Hive weight and internal temperature during a field observation period associated with nectar flow and increased colony activity.

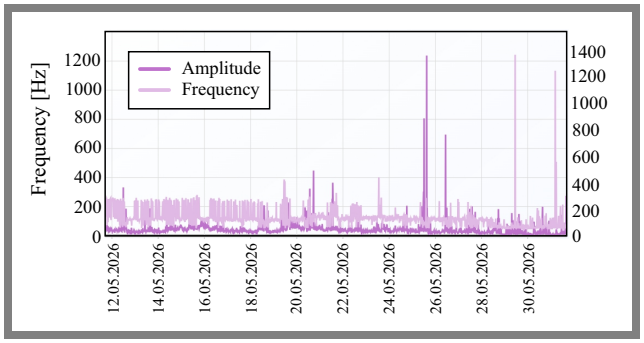


Fig. 12. Acoustic descriptors recorded during the same observation period. Most measurements remained between 200 and 300 Hz, while short-term anomalies exceeded 800 Hz.

During the observation period, the weight of the hive increased from approximately 11 kg to more than 22 kg, indicating intensive accumulation and nectar flow. At the same time, the internal temperature of the hive remained relatively stable between 34 and 35°C, suggesting normal brood thermoregulation.

The acoustic observations revealed dominant frequencies typically ranging between 200 and 300 Hz, which is consistent with colony activity associated with foraging, nectar processing, and ventilation reported in the literature. Several short-duration anomalies exceeded 800 Hz. Such events may indicate periods of elevated colony excitation and could potentially precede swarming-related behavior or other colony disturbances. More studies based on larger datasets are required to verify this hypothesis.

7. Discussion

The obtained results support the main assumption of the proposed framework: useful acoustic descriptors can be extracted locally from short FFT windows and transmitted as compact telemetry variables. This reduces the need for raw audio storage and enables acoustic monitoring in bandwidth-constrained IoT deployments.

The proposed framework follows the broader trend of moving signal processing from cloud platforms to resource-constrained edge devices [8].

Recent studies have investigated TinyML-based approaches for detecting acoustic events directly on embedded devices [11].

Although such methods may provide higher classification capabilities, they usually require trained machine learning models, additional memory resources, and increased computational complexity.

In contrast, the proposed FFT-based approach relies on deterministic signal processing and does not require model training, inference engines, or neural network parameters. As a result, the method remains lightweight, interpretable, and suitable for deployment on low-cost ESP32-class devices operating under strict memory and energy constraints.

An important finding is that dominant frequency alone did not reliably distinguish queenright and queenless recordings. This is relevant because many simple acoustic monitoring approaches focus on tracking a single dominant frequency value. In the present data set, the amplitude- and energy-based descriptors were more informative. At the segment level, band energy, mean amplitude, and peak amplitude showed the largest effect sizes. At the file level, the mean amplitude remained statistically significant despite the small number of independent recordings.

File-level analysis is especially important, since audio segments extracted from the same recording cannot be treated as fully independent observations. The segment-level results should therefore be interpreted as descriptor-level evidence, while the file-level results provide a more conservative assessment.

The study has several limitations. In general, the analyzed material comprised 14 recordings, and the queenright and queenless labels were based on preliminary manual classification. The recordings did not include complete biological metadata such as brood status, nectar flow, Varroa infestation, weather conditions, hive type, or various microphone placements. Therefore, the results should be treated as a validation of the extraction and communication reduction of embedded features rather than as a final biological diagnostic model.

Therefore, this data set should be regarded as a proof-of-concept data set intended for validating the proposed embedded processing framework, rather than an attempt to develop a production-ready biological classifier.

The proposed approach may also be generalized to other resource-constrained IoT acoustic monitoring applications, including environmental sensing, industrial condition monitoring, smart agriculture, wildlife monitoring, and distributed edge intelligence systems operating over LPWAN, LTE-M and NB-IoT networks.

The segment-duration experiment showed that 2-second segments already preserve the dominant acoustic characteristics while minimizing memory requirements and processing latency. Longer segments increased computational cost without providing a proportional improvement in the stability of the descriptor. Therefore, 2-second windows were selected as a compromise between spectral stability and embedded system constraints.

Although the mean dominant frequency did not differ significantly between queenright and queenless colonies, queenless recordings exhibited a dominant-frequency variability that was almost two times higher than in the case of queenright recordings (CV of 6.97% versus 3.75%). One possible explanation is that queenless colonies exhibit less stable collective behavior and increased acoustic heterogeneity.

Although the present study does not attempt to establish biological causality, the observed increase in frequency variability suggests that stability-related acoustic metrics may provide additional diagnostic value beyond mean frequency alone. In practical NB-IoT and LTE-M deployments, such payload reduction directly translates into shorter radio activity periods, lower energy consumption, and improved battery lifetime.

8. Conclusions

This paper presented a communication-efficient embedded FFT framework for acoustic telemetry in smart beehive IoT systems. The proposed approach extracts compact FFT-derived descriptors, including dominant frequency, amplitude-based measures, spectral centroid, spectral entropy, and band energy. The method enables local signal processing on resource-constrained nodes and substantially reduces communication requirements compared to raw audio transmission.

The primary contribution of this work is not the biological classification of colony states, but the demonstration that diagnostically relevant acoustic information can be retained while reducing communication payload by approximately four orders of magnitude (up to 8192-fold).

This enables scalable deployment of acoustic sensing in large-scale IoT monitoring systems operating under bandwidth and energy constraints.

Experimental validation using queenright and queenless recordings showed that dominant frequency alone was not a reliable discriminator. In contrast, amplitude-based and energy-related descriptors were more informative. File-level analysis confirmed that mean spectral amplitude remained statistically significant between groups, while queenless recordings also showed higher dominant frequency variability. The embedded benchmark further demonstrated that the proposed processing pipeline satisfies real-time constraints while requiring only modest computational resources on an ESP32-class microcontroller.

The proposed framework provides a practical and scalable basis for acoustic telemetry in precision apiculture and demonstrates how embedded edge processing can substantially reduce communication overhead in LPWAN-based IoT systems. Future work will include validation on substantially larger multi-season datasets together with direct measurements of embedded execution time, communication energy consumption, and long-term field deployment. The proposed framework may also be adapted to other LPWAN-based embedded acoustic sensing applications where communication efficiency is a primary concern.

Acknowledgments

The author thanks the beekeepers who supported the collection of acoustic recordings and field observations relied upon in this study.

References

[1] S. Ferrari, M. Silva, M. Guarino, and D. Berckmans, "Monitoring of Swarming Sounds in Bee Hives for Early Detection of the Swarming Period", *Computers and Electronics in Agriculture*, vol. 64, pp. 72–77, 2008 (https://doi.org/10.1016/j.compag.2008.05.010).

[2] M. Bencsik *et al.*, "Honeybee Colony Vibrational Measurements to Highlight the Brood Cycle", *PLOS ONE*, vol. 10, art. no. 0141926, 2015 (https://doi.org/10.1371/journal.pone.0141926).

[3] A. Terenzi, S. Cecchi, and S. Spinsante, "On the Importance of the Sound Emitted by Honey Bee Hives", *Veterinary Sciences*, vol. 7, art. no. 168, 2020 (https://doi.org/10.3390/vetsci7040168).

[4] W.G. Meikle and N. Holst, "Application of Continuous Monitoring of Honeybee Colonies", *Apidologie*, vol. 46, pp. 10–22, 2014 (https://doi.org/10.1007/s13592-014-0298-x).

[5] H. Hadjur, D. Ammar, and L. Lefèvre, "Toward an Intelligent and Efficient Beehive: A Survey of Precision Beekeeping Systems and Services", *Computers and Electronics in Agriculture*, vol. 192, art. no. 106604, 2022 (https://doi.org/10.1016/j.compag.2021.106604).

[6] S. Górecki *et al.*, "Improving Bee Living Conditions Through Ecological Thermal Insulation and Remote Early Anomaly Detection-vital Step Towards Preserving Bees Population", *Fibres and Textiles in Eastern Europe*, vol. 32, pp. 1–12, 2024 (https://doi.org/10.2478/ftee-2024-0025).

[7] H. Hadjur, D. Ammar, and L. Lefèvre, "Analysis of Energy Consumption in a Precision Beekeeping System", *Proc. of the 10th International Conference on the Internet of Things*, art. no. 20, 2020 (https://doi.org/10.1145/3410992.3411010).

[8] A. De Simone, L. Barbisan, G. Turvani, and F. Riente, "Advancing Beekeeping: IoT and TinyML for Queen Bee Monitoring Using Audio Signals", *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–9, 2024 (https://doi.org/10.1109/TIM.2024.3449981).

[9] A. Turyagyenda *et al.*, "IoT and Machine Learning Techniques for Precision Beekeeping: A Review", *AI*, vol. 6, art. no. 26, 2025 (https://doi.org/10.3390/ai6020026).

[10] K. Jeong *et al.*, "IoT and AI Systems for Enhancing Bee Colony Strength in Precision Beekeeping: A Survey and Future Research Directions", *IEEE Internet of Things Journal*, vol. 12, pp. 362–389, 2025 (https://doi.org/10.1109/JIOT.2024.3461775).

[11] B.B. Roy, S. Das, and U.K. Mondal, "TinyML-driven Sensor Nodes for Energy-efficient Acoustic Event Detection in Pervasive Acoustic WSNs", *Journal of Telecommunications and Information Technology*, no. 2, pp. 69–77, 2025 (https://doi.org/10.26636/jtit.2025.2.2084).

[12] U. Ghosh and U.K. Mondal, "Pilot Agent-driven Wireless Acoustic Sensor Network for Uninterrupted Data Transmission", *Journal of Telecommunications and Information Technology*, no. 4, pp. 53–60, 2023 (https://doi.org/10.26636/jtit.2023.4.1322).

[13] P. Rajchowski, "Examination of 5G NR, LTE, and NB-IoT Radio Interfaces and Their Vulnerabilities to Interference", *Journal of Telecommunications and Information Technology*, no. 4, pp. 93–100, 2024 (https://doi.org/10.26636/jtit.2024.4.1960).

[14] A. Zgank, "IoT-based Bee Swarm Activity Acoustic Classification Using Deep Neural Networks", *Sensors*, vol. 21, art. no. 676, 2021 (https://doi.org/10.3390/s21030676).

[15] M. Abdollahi, P. Giovenazzo, and T. Falk, "Automated Beehive Acoustics Monitoring: A Comprehensive Review of the Literature and Recommendations for Future Work", *Applied Sciences*, vol. 12, art. no. 3920, 2022 (https://doi.org/10.3390/app12083920).

[16] D. Kanelis *et al.*, "Decoding the Behavior of a Queenless Colony Using Sound Signals", *Biology*, vol. 12, art. no. 1392, 2023 (https://doi.org/10.3390/biology12111392).

[17] A. Farina, "Discovering Ecoacoustic Codes in Beehives: First Evidence and Perspectives", *Biosystems*, vol. 234, art. no. 105041, 2023 (https://doi.org/10.1016/j.biosystems.2023.105041).

[18] J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series", *Mathematics of Computation*, vol. 19, pp. 297–301, 1965 (https://doi.org/10.2307/2003354).

[19] A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, 2nd ed., Prentice Hall, 870 p., 1999, (ISBN 9780137549207).

[20] S.W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, California Technical Publishing, 1997, (ISBN 9780966017632).

[21] Knowles Electronics (Syntiant), MEMS Sisonic Digital Multimode Microphone SPH0645LM4H-B, datasheet specification.

Sebastian Górecki, Ph.D. Candidate
Institute of Electronics
https://orcid.org/0000-0001-5700-4000
E-mail: research@sgorecki.eu
Lodz University of Technology, Łódź, Poland
https://p.lodz.pl

Network Threat Detection in IaaS Environments Based on Flow Log Data

Hubert Wójcik, Mirosław Roszkowski, and Andrzej Mycek

Cracow University of Technology, Cracow, Poland

https://doi.org/10.26636/jtit.2026.3.2628

Submitted: 23 May 2026, Reviewed: 1 Jul. 2026

Abstract — The rapid growth of cloud services has led security monitoring to rely mainly on limited telemetry data furnished by cloud service providers. Flow logs, such as VPC Flow Logs in Amazon Web Services, are one of the key sources of information about network traffic in IaaS environments. In this study, a fully automated experimental environment was designed and deployed in the AWS cloud using the infrastructure-as-a-code approach with Terraform. The environment includes a virtual network, flow logging mechanisms, and controlled attack scenarios generating characteristic traffic patterns, such as port scanning, brute-force authentication attempts, and data exfiltration. The collected data was analyzed using cloud-native tools, in particular Amazon Athena, which enabled a detailed investigation of anomaly detection based on flow-level metrics. The results confirm that selected threats can be effectively detected using traffic metadata, including the number of unique destination ports, the repetition of communication attempts, the volume of transferred data and the temporal regularity of flows. The analysis demonstrates that flow logs alone are not sufficient to clearly distinguish between malicious activity and legitimate operations with similar characteristics, highlighting inherent limitations of telemetry in the IaaS model. The paper outlines directions for future work, including correlation with additional log sources and integration with ML models to improve detection accuracy and reduce false positives.

Keywords — AWS, cloud security, IaaS, intrusion detection, network traffic analysis

1. Introduction

The rapid growth of cloud computing services has led to an increasing number of information systems operating based on the IaaS model. Despite their numerous advantages, such as scalability and flexibility, these environments are exposed to a wide range of security threats, particularly at the network level. Network traffic analysis is one of the key techniques for identifying malicious activities and assessing the overall security posture of an infrastructure. The complexity of modern cloud environments makes monitoring a key component in ensuring security, reliability, and compliance with organizational policies. Observability of distributed and dynamically changing cloud resources remains a key challenge [**?**].

In the context of cloud environments, especially within the IaaS model, network traffic analysis is critical given the highly dynamic nature of resources and the limited visibility provided by traditional network inspection mechanisms. The ephemeral lifecycle of cloud resources and the abstraction layers introduced by cloud providers restrict direct access to network-level telemetry, thereby increasing the complexity of effective security monitoring.

Cloud services can be delivered through various service models that differ in the division of responsibilities between the service provider and the end user. The most commonly adopted models include IaaS, Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS).

The IaaS model provides users with access to fundamental infrastructure resources, including virtual machines, virtual networks, and data storage systems. The responsibility for configuring operating systems, implementing network security controls, and securing deployed applications depends, largely, on the user. Although this model offers great flexibility, it also requires a systematic and conscious approach to security management [**?**].

From a security perspective, the concept of shared responsibility is paramount. According to this model, the cloud service provider is responsible for securing the underlying physical infrastructure, while the user is responsible for securing the resources deployed within the service. In the IaaS model, this responsibility includes, among other things, setting network security rules, implementing access control mechanisms, and continuously monitoring network traffic.

Network traffic in cloud environments differs significantly from that observed in traditional data centers. This distinction arises primarily from resource virtualization, dynamic infrastructure scaling, and software-level network segmentation. As a result, traffic patterns in cloud-based systems are more fluid and less predictable than those in static on-premises environments. In cloud environments, two fundamental types of network traffic can be distinguished: external traffic, which occurs between cloud-based resources and the public network, and internal traffic, which occurs within the cloud infrastructure between individual virtual resources [**?**].

Despite the increasing adoption of flow log–based telemetry mechanisms, relatively few research papers analyze the detection of multiple threat categories exclusively using such data sources. The existing literature focuses primarily on ML-based approaches for specific attack scenarios, such as distributed denial-of-service (DDoS) attacks or port scanning, while paying considerably less attention to the simulta-

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

neous analysis of port scanning, brute force attacks, beaconing, and data exfiltration within a unified IaaS environment.

Survey papers on IaaS security focus on general aspects of infrastructure layer security, such as cryptography and insider threats, rather than on the limitations of network telemetry and their impact on the detection of network-based threats.

Furthermore, research on anomaly detection using VPC Flow Logs has gained momentum only in recent years, predominantly in the context of Internet of Things (IoT) environments or machine learning (ML) detection techniques. However, there remains a lack of systematic analysis that addresses multiple classes of network attacks within a cohesive AWS IaaS architecture [**?**].

In this work, an experimental approach was adopted to evaluate the feasibility of using network flow logs – a primary source of threat intelligence in IaaS environments. In contrast to existing research, which typically focuses on individual attack types or relies on advanced ML models, the analysis in this study emphasizes the identification of network traffic patterns characteristic of different threat classes, by relying on analytical methods.

The main contribution of this work is the design of a fully controlled experimental environment in the AWS cloud and the demonstration of the extent to which anomaly detection is possible using network traffic metadata only, without access to full packet payloads.

2. Related Works

Research on threat detection in IaaS cloud environments increasingly focuses on network traffic objects, such as flow logs. The growing adoption of this approach is driven by technical limitations associated with full packet inspection, as well as by the scale and dynamic nature of modern cloud infrastructures. Numerous studies emphasize that flow logs provide a lightweight, scalable means of monitoring network traffic in the cloud, which differs significantly from traditional methods based on packet capture (PCAP) or NetFlow.

For example, studies conducted in IoT environments clearly demonstrate that VPC Flow Logs provide only basic metadata, such as packet and byte counts, source, destination addresses, and protocol information. Additionally, the flow logs are segmented according to time-based windows, thus hindering session reconstruction and comprehensive communication analysis. Despite these limitations, appropriate aggregation and transformation of flow logs, such as bidirectional flow reconstruction, can significantly improve the effectiveness of anomaly detection in cloud environments. Cloud service providers recommend using flow log-based monitoring for analyzing unauthorized traffic, detecting anomalies, and correlating events within security information and event management (SIEM) systems.

Despite the growing number of publications on threat detection in cloud environments, there remains a noticeable lack of studies that systematically analyze the limitations of relying solely on network flow logs for detecting multiple classes of attacks simultaneously in IaaS environments.

Existing research primarily focuses on improving detection performance through ML models. However, the interpretability of the results and the feasibility of applying simpler analytical approaches in production environments are rarely examined.

Therefore, a justified need exists for an analysis that relies on an experimental approach using real cloud infrastructure and focuses on the practical aspects of threat detection under conditions of limited data visibility.

2.1. Anomaly Detection Using ML Models

The vast majority of contemporary research focuses on the automatic detection of threats using ML algorithms that analyze network metadata only. However, the literature also presents more advanced approaches that leverage network flows and native cloud observability services to design autonomous threat detection systems. Particularly noteworthy are studies that concentrate on combining system-level telemetry with network metadata and on applying ML models to enhance attack detection effectiveness.

An example of such an approach is an incident detection system based on AWS services, which integrates data from CloudWatch and VPC Flow Logs with an analytical pipeline that uses Amazon SageMaker and serverless services such as AWS Lambda and SNS. The system is designed to be fully autonomous and adaptive, addressing the limitations of traditional rule-based monitoring systems, including static CloudWatch alarms and signature-based mechanisms, which struggle to detect complex and evolving threats. These studies demonstrate that enriching flow log analysis with ML techniques – such as isolation forest, deep autoencoders, random forest, and XGBoost – significantly improves detection effectiveness, even for attacks that are difficult to capture while relying on classical methods. This includes brute-force attempts, port scanning, data exfiltration, and DDoS attacks.

The experimental evaluation was conducted using a hybrid dataset comprising more than 2.8 million records collected during both regular operation and simulated attack scenarios, enabling a reliable assessment of model performance. Among the evaluated models, XGBoost proved to be the most effective, achieving high classification accuracy and low detection latency, while substantially reducing false positives. Feature importance analysis revealed that key indicators for anomaly detection include flow characteristics such as traffic volume, flow duration, and destination port entropy. The findings of these studies are highly relevant to the present work, as they show that despite the limited informational scope of flow logs, their integration with data processing techniques and ML models can enable effective, scalable and cost-efficient threat detection. At the same time, they highlight the growing role of automation and ML-based approaches in monitoring cloud environments, providing an important complement to the research on classical methods used for analyzing anomalous traffic patterns in IaaS infras-

tructures [?]. The authors of [?] presented a survey of intrusion detection methods based exclusively on data derived from VPC Flow Logs. The work indicates that analyzing network traffic metadata can be an effective means of identifying security incidents, even in environments where packet inspection is not feasible or is significantly constrained. Flow-based security analysis has also been investigated in forensic contexts. In [?], the concept of "superflows" is proposed, which aggregates related network flows to facilitate threat searches, incident investigations and traffic pattern analysis. The work demonstrates that higher-level aggregation of flow metadata may significantly improve the efficiency of security analysis in large-scale network environments.

Recent studies have also demonstrated the effectiveness of flow-based approaches for anomaly detection in large-scale data centers and cloud environments. Such approaches can provide scalable threat detection capabilities without requiring deep packet inspection [?], [?]. The growing scale of modern cloud infrastructures has also motivated research into large-scale anomaly detection and diagnosis mechanisms relying on network telemetry collected from production cloud environments. Such approaches enable the identification of network- and application-level anomalies across millions of cloud resources [?].

2.2. Security in the Infrastructure-as-a-Service Model

Security in the IaaS model has been extensively studied by researchers in cloud computing, particularly those who focus on the infrastructure layer. In [?], an industry-driven case study is presented accompanied by a large, real telemetry dataset aimed at enabling research on anomaly detection in complex cloud environments. The authors emphasize that contemporary large-scale cloud systems (LCSs) are becoming increasingly complex and that their reliability largely depends on effective early anomaly detection. Unfortunately, the scientific literature lacks large and realistic datasets that could serve as benchmarks for evaluating anomaly detection methods.

To address this research gap, the authors of [?] released a comprehensive data set collected from IBM Cloud over a period of 4.5 months. The data set comprises 39 365 records and 117 448 features, spanning various types of telemetry data and operational metrics. This data set enables analysis of cloud behavior under both normal operating conditions and during system failures and anomalies. In their work, the authors also demonstrated the application of ML models for anomaly detection in their proprietary dataset. They discussed the data preprocessing pipeline, challenges arising from the high dimensionality and complexity of the data, and issues typical of telemetry analysis in distributed environments.

The applied ML models are intended to support system health monitoring and improve the effectiveness of early detection of abnormal system behaviors. In the context of IaaS environments, it is essential to recognize that cloud telemetry is subject to inherent limitations. Article [?] discusses the lack of complete control over the underlying cloud infrastructure, the constraints resulting from access to only partial datasets, and the influence of the cloud service provider's architecture, all of which significantly affect the ability to perform a precise and comprehensive analysis of security events. Previous work [?] emphasized that the IaaS architecture inherently introduces significant security limitations, including those resulting from tenant isolation, limited visibility into network traffic, and dependence on the hypervisor. These factors collectively translate into substantial constraints regarding the telemetry available to cloud users.

Previous studies have shown that visibility into network communications remains one of the key challenges in native cloud environments. The growing reliance on virtualization mechanisms and software-defined networking can significantly limit the availability of security-relevant telemetry, making network traffic metadata an extremely important source of information for anomaly detection and security monitoring [?].

The challenges associated with securing cloud environments are not limited exclusively to the network layer. Recent research on managed Kubernetes services in public cloud environments [?] demonstrates that effective protection of native cloud applications requires integrating multiple security mechanisms, including identity management, compliance assessment, auditing, system hardening, and continuous monitoring. These findings further emphasize the importance of telemetry and security visibility as fundamental components of cloud security architectures.

The scientific literature also describes solutions that leverage native cloud provider security services. An example is the integration of AWS Security Hub and Amazon GuardDuty, which centralizes threat intelligence, event correlation, and the automation of security incident detection processes [?]. Such solutions leverage multiple telemetry sources available in cloud environments and serve as an important reference point for analysis that relies exclusively on AWS VPC Flow Logs data.

3. Environment Architecture

The objective of this research was to create a cloud-based infrastructure capable of generating, collecting and analyzing network traffic to detect selected security threats. The environment architecture was designed to support both the reproduction of realistic attack scenarios and the analysis of network traffic based on flow log data. The environment was implemented using the IaaS model and consists of a logically isolated virtual network, compute instances that serve as test resources, and components responsible for monitoring, security event detection, and data analysis. The overall architecture is illustrated in Fig. **??**.

The core component has the form of a logically isolated virtual network within which all compute resources are deployed. This network constitutes an isolated address space that enables controlled observation of network traffic both within the infrastructure and in communication with the pub-

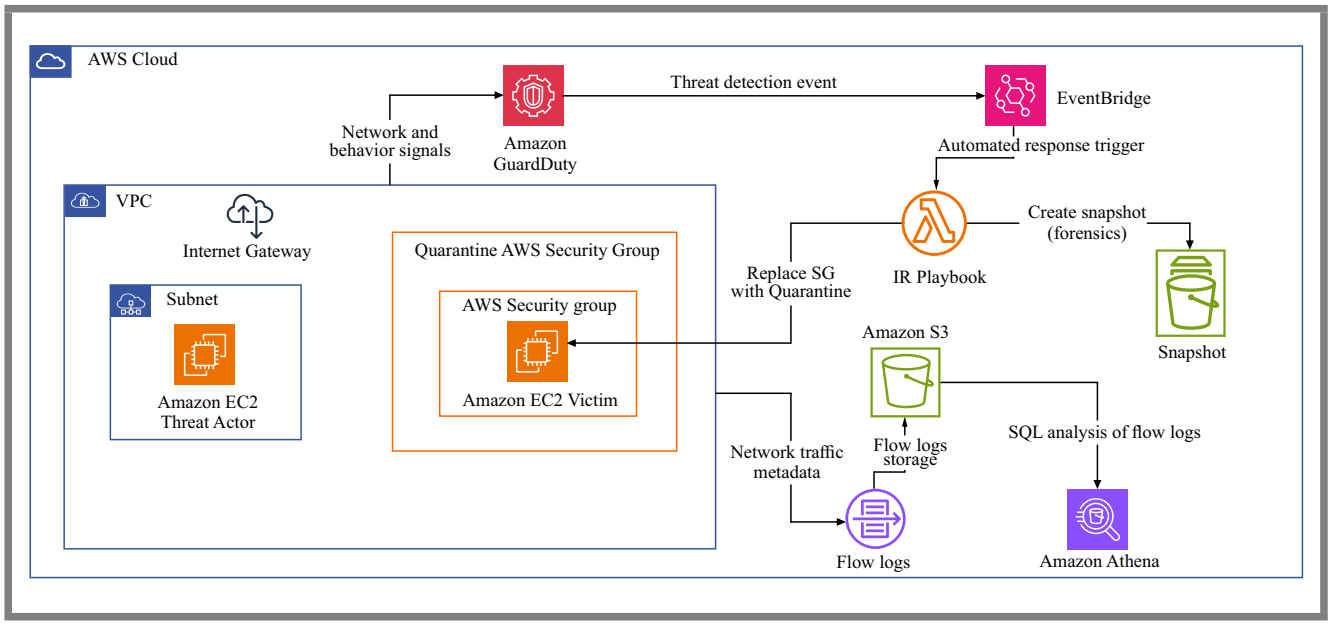


Fig. 1. Research environment architecture in AWS Cloud.

lic network. Within the virtual network, a subnet was created to host the compute instances. Access to the public network is provided through an Internet gateway, enabling inbound and outbound traffic. Network communication between resources, as well as communication with external networks, is governed by firewall mechanisms implemented as security groups.

Figure **??** illustrates two compute instances serving different roles. One instance generates network traffic, while the other serves as the target resource for executing the test scenarios. This separation of roles enables unambiguous analysis of both the direction and the characteristics of the observed network traffic. The experimental environment enables the reproduction of realistic threat scenarios while maintaining full control over the generated network traffic. This, in turn, enables consistent and comparable input data for further analysis, which is critical to evaluating the effectiveness of anomaly detection.

3.1. Network Traffic Generation, Logging Mechanism, and Threat Detection

Network traffic generated within the research environment includes internal communication between compute instances and connections established via the public network. Every connection that traverses the network infrastructure is recorded in flow logs. The flow log metadata describe network connections, such as source and destination IP addresses, ports, protocols, packet counts, and data volumes. These data do not include packet payloads; however, they enable the identification of characteristic communication patterns.

The collected flow logs are stored in a centralized data repository, enabling further analysis using analytical tools. This approach supports both retrospective network traffic analysis and the correlation of events across different time intervals. Within the architecture of the research environment, a threat detection mechanism is implemented based on the analysis of network and behavioral signals. Events indicating potential security breaches can be generated based on observed network traffic patterns as well as other environmental parameters.

The collected network flow logs are analyzed using a tool that supports the execution of analytical queries. Analysis is performed using SQL-based queries, allowing data filtering, aggregation, and correlation across different time intervals.

This approach enables the identification of anomalies in network traffic, such as an unusually high number of connections, unexpected data volumes, or recurring communication patterns. The results of the analysis provide a basis for further investigation and evaluation of the effectiveness of network flow logs for threat detection.

4. Implementation

The infrastructure was deployed using an Infrastructure-as-Code (IaC) approach, which enabled repeatable resource provisioning, change version control, and reduced errors resulting from manual configuration. The implementation leveraged HashiCorp Terraform, with the configuration structured into an environment layer and a set of modular components corresponding to the individual elements. IaC makes it possible to manage the entire infrastructure (e.g., virtual machines, load balancers, virtual networks, and other services) by writing source code [**?**], [**?**].

HashiCorp Terraform is a tool that enables the declarative definition of an infrastructure in which the desired state of resources is described using configuration files (Fig. **??**). Based on these definitions, the tool generates an execution plan and subsequently applies it to the target cloud environment.

```
terraform {
  required_version = ">= 1.6.0"
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 5.0.0"
    }
  }
}
```

Fig. 2. Version and provider definition.

This approach makes it possible to recreate infrastructure in a controlled and repeatable manner. Within the project, Terraform was used to deploy key components of the environment, including the network layer, compute resources, network traffic logging mechanisms, and elements supporting incident response. This approach enabled consistent configuration management throughout the architecture and facilitated modifications during the project implementation [**?**]. To ensure environmental consistency and version control, explicit declarations of the required Terraform version and the cloud provider version were applied.

The environment layer serves as the entry point for infrastructure deployment and is responsible for integrating individual modules into a cohesive architecture. Within the `envs/dev` directory, the provider configuration, environment variables, and module invocations are defined that implement specific components of the research environment. The use of a separate environment layer enables straightforward adaptation of the infrastructure to different run-time contexts by modifying variable values, without the need to alter module definitions.

This approach also improves the readability of the configuration and simplifies the management of dependencies between resources. The configuration of the cloud provider and the required tool versions was defined in dedicated configuration files within the environment layer. Environment-specific parameters, such as the deployment region and network settings, were declared as input variables, allowing them to be easily modified without affecting the overall structure of the project.

4.1. Implementation of the Network Layer

The network module is responsible for provisioning the foundational network layer of the experimental environment. As part of the implementation, a virtual private cloud (VPC) was defined together with two subnets: a public subnet and a private subnet. This separation enables explicit control over network traffic direction and improves resource isolation, which is essential both from a security perspective and for conducting controlled network traffic experiments.

The network layer was designed to provide Internet access for resources deployed in the public subnet while preserving the isolation of the private subnet, which has no direct connection with the Internet. This design reflects a typical architecture commonly observed in production cloud environments, where publicly accessible components are separated from internal resources.

The virtual network was created using the `awsvpc` resource with DNS support enabled, allowing proper name resolution and simplifying the instance configuration. The VPC and subnet address spaces were parameterized using variables `vpccidr, publicsubnetcidr and privatesubnetcidr` variables, allowing for easy modification of the network topology without altering the module's internal logic.

To simplify deployment, a single availability zone (AZ) was used, dynamically selected from those available within the target region. Within the module, two subnets with distinct roles were defined. The public subnet was configured to allow instances launched within it to receive public IP addresses, thereby enabling inbound and outbound Internet connectivity. The private subnet serves as an internal segment intended for resources that should not be directly exposed to the public network.

This network segmentation supports the testing of attack scenarios originating from external traffic and the observation of internal traffic flows between resources within the infrastructure. The overall design aligns with common best practices employed in cloud environments such as Amazon Web Services, while remaining sufficiently flexible for experimental and research-oriented use cases.

4.2. Computing Instance Definition and Log Access Controlling

As part of the module, two EC2 instances were provisioned, each serving a distinct functional role. The attacker instance is used to generate network traffic in accordance with predefined test scenarios, while the victim instance represents the target resource against which the effects of network activities are observed. The use of role-based instance tags facilitates automation in subsequent stages of the project, particularly with respect to incident response mechanisms.

A key aspect of the configuration is the assignment of instances to appropriate subnets and security groups. The attacker instance is deployed within a public subnet, ensuring its ability to communicate with external networks. The victim instance is launched according to the module configuration, with its placement in a public or private subnet determined by the adopted architectural assumptions. In this study, the placement is aligned with the architecture diagram of the experimental environment to ensure consistency.

Access to write data to the S3 storage resource is restricted exclusively to the service responsible for delivering network flow logs. To achieve this, a dedicated resource policy was defined, allowing object-write operations only for the appropriate system service and enforcing the use of required access control attributes.

This approach ensures that log data cannot be modified or written by unauthorized entities while simultaneously satisfying the technical requirements of the network traffic logging mechanism.

4.3. Test Scenarios and Network Traffic Generation

The test scenarios were designed to generate characteristic traffic patterns that can be identified using data derived from network flow logs. All scenarios were executed in a controlled experimental environment using the previously deployed computing instances assigned the roles of attacker and victim. Each scenario was conducted in a repeatable manner while maintaining constant environmental parameters, thus allowing for a reliable comparison of the results between individual tests.

To establish a baseline for analysis, network traffic generated during normal instance operation was first recorded. This reference traffic includes standard administrative communication, as well as connections initiated by the operating system. Such traffic is characterized by a low number of connections, a stable data volume, and the absence of abrupt temporal variations. The recorded baseline traffic served as a reference point for subsequent analyses of scenarios that introduce anomalous network behavior.

One of the scenarios analyzed involved port scanning of the target instance. This activity generates a large number of short-lived connections initiated within a narrow time window, often directed to multiple destination ports. In network flow logs, port scanning manifests itself as an increased number of connection attempts, low data transfer volumes, and short session durations. A distinguishing feature of this scenario is the repetition of source IP addresses combined with varying destination ports.

Another test scenario consisted of repeated authentication attempts against a network service running on the target instance.

This type of activity produces a sequence of connections to the same destination port, typically occurring at regular time intervals. In flow log data, such traffic is characterized by a high number of connections targeting a single port, moderate session durations, and recurring communication patterns.

All experimental evaluations were conducted in the controlled AWS cloud environment described in the previous sections. The analysis was based on data collected from AWS VPC Flow Logs, which were aggregated and stored in a centralized data repository. The stored flow records were subsequently queried and analyzed using Amazon Athena.

Each experimental scenario was executed under identical infrastructure conditions to ensure comparability of observed network traffic patterns. Additionally, baseline traffic reflecting the normal operation was captured to serve as a reference point. This baseline enabled the identification of the differences between legitimate network communication and anomalous traffic.

The objective of this study was not to evaluate the detection performance of a specific algorithm. Instead, it focused on identifying network traffic characteristics that may serve as reliable indicators of cyber threats using solely metadata extracted from network flow records.

```
CREATE EXTERNAL TABLE IF NOT EXISTS vpc_flow_logs (
  version INT,
  account_id STRING,
  interface_id STRING,
  srcaddr STRING,
  dstaddr STRING,
  srcport INT,
  dstport INT,
  protocol INT,
  packets BIGINT,
  bytes BIGINT,
  start_time BIGINT,
  end_time BIGINT,
  action STRING,
  log_status STRING
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ' '
LOCATION 's3://network-analysis-dev-flowlogs-260905331134/';
```

Fig. 3. Definition of an external table for network flow logs.

5. Network Traffic Analysis

The network traffic analysis was conducted on data collected in the form of network flow logs. AWS Athena was used as the analytical tool, enabling the execution of SQL-based analytical queries without the need for prior data preprocessing. The objective of the analysis was to identify characteristic network traffic patterns generated by individual test scenarios and to evaluate the feasibility of anomaly detection based solely on traffic metadata.

5.1. Data Pre-processing for Analysis

The first step of the analysis involved defining the data structure that represents the flow logs. For this purpose, an external table was created, pointing to the location of the data stored in the S3 resource (Fig. **??**). The table reflects the core fields of the flow logs, including source and destination IP addresses, ports, data transfer volumes, and timestamps.

The anomaly analysis was performed by examining specific traffic characteristics for each attack scenario using AWS VPC FlowLogs. In the port scanning scenario, the primary indicator was a high number of connection attempts originating from the same source IP address to multiple destination ports. For the repeated authentication scenario, the analysis focused on the frequency and temporal distribution of connections that target the SSH service.

In the potential data exfiltration scenario, the primary metric was the volume of data transferred. Detection was based on identifying unusually large data transfers relative to the baseline network communication observed within the environment. Given the exploratory nature of this study, no detection thresholds were defined. Instead, the evaluation relied on comparing network traffic patterns recorded during normal system operation with those generated by experimental attack scenarios.

5.2. Port Scanning Analysis

To identify port scanning, an aggregation-based analysis was performed, focusing on the number of connections and the number of unique destination ports observed for all source–destination IP address pairs, without prior knowledge of the communication origin or target.

Based on the query results, a list of IP address pairs that exhibit the highest number of unique destination ports was drawn up, serving as a key indicator of potential port scanning activity (Tab. **??**).

The results indicate the presence of several IP address pairs generating traffic with a clearly anomalous pattern. In particular, one pair stands out, exhibiting more than one hundred unique destination ports, which distinctly distinguishes it from typical application traffic, which is usually limited to a small number of ports.

Based on the query results, a list of IP address pairs generating the highest number of connections to the SSH service was compiled (Tab. **??**).

The results indicate a significant variation in the intensity of SSH traffic between individual IP address pairs. The highest number of connections was observed for an internal address pair, with 240 connections recorded to port 22. This volume by far exceeds typical administrative traffic and may indicate repeated authentication attempts occurring within a short time window.

The remaining records include both traffic between internal resources and connections initiated from an external IP address. In these cases, the number of connections is relatively low and can be interpreted as either isolated access attempts or elements of automated scanning of publicly available services.

Traffic on ports 443 and 80 corresponds to standard network communication over HTTPS and HTTP protocols, respectively. Their presence is typical for cloud environments and does not show the regular patterns characteristic of beacon-type communication. Port 8080, despite a relatively low number of connections, was identified as a potential candidate for further analysis due to its atypical usage and previous observations suggesting its involvement in automated communication. Unlike standard system ports, non-standard ports are more frequently used as communication channels in beaconing scenarios.

A high number of repeated connections to a single port within a short period of time is a characteristic indicator of brute-force authentication attempts. The case analyzed demonstrates a clear deviation from standard administrative traffic, which is typically characterized by a limited number of sessions of longer duration. However, in cloud environments, similar traffic patterns may also be generated by automation systems or infrastructure management tools, which further complicates the unambiguous interpretation of the results.

Tab. 1. Results of port scanning analysis.

SRC IP	DST IP	Connections	Dst. Ports
10.0.1.254	10.0.1.180	257	122
185.125.190.58	10.0.1.254	51	51
91.189.91.157	10.0.1.254	45	45
185.125.190.57	10.0.1.254	43	43
185.125.190.56	10.0.1.180	42	42
91.189.91.157	10.0.1.180	32	32
194.58.204.20	10.0.1.180	31	31
172.232.146.46	10.0.1.254	31	31
45.83.221.52	10.0.1.180	30	30
185.125.190.56	10.0.1.180	29	29

Tab. 2. Analysis of SSH connection counts.

SRC IP	DST IP	SSH connections
10.0.1.180	10.0.1.254	240
83.27.3.170	10.0.1.254	12
83.27.3.170	10.0.1.180	11
10.0.1.254	10.0.1.180	7

5.3. Data Exfiltration Analysis

Data exfiltration is characterized by an increased volume of outbound traffic from the analyzed resource, often occurring in a manner that deviates from the typical communication profile. Unlike port scanning or beaconing-type communication, the primary indicator of potential exfiltration is not the number of connections but the total amount of data transferred in a given direction. To identify potential data transfers, an analysis was performed aggregating the total number of bytes transferred and the number of flows for each destination port, without any prior knowledge of the roles of the source and destination in the communication.

In analyzing data exfiltration, the critical factor is the total volume of transmitted data, rather than the number of connections. Elevated outbound traffic can indicate a potential security breach. However, its interpretation requires considering the operational context of the system.

In practice, detecting such events solely from network traffic metadata may yield ambiguous conclusions, particularly in environments with highly variable workload profiles. However, it is important to note that an increased volume of transmitted data does not necessarily indicate a security incident.

Similar traffic patterns may also occur during legitimate operations such as backup processes, system updates, or administrative file transfers. Consequently, analyses based solely on network flow metadata may yield ambiguous results and should be supplemented with context from other data sources to improve detection accuracy.

6. Summary and Future Work

The objective of this study was to design an analytical procedure and to analyze network traffic in an IaaS cloud environment using network flow logs. Within the scope of the work, a research environment was created in the cloud, data collection mechanisms were configured, and selected scenarios generating characteristic network traffic patterns were analyzed.

The experiments conducted in this study were qualitative in nature and focused on identifying network traffic patterns associated with selected categories of security threats. Consequently, classification metrics such as precision, recall, and false positive rate were not evaluated, as these measures are more appropriate for approaches based on formal classification techniques or ML models.

Nevertheless, the findings clearly demonstrate that AWS VPC Flow Logs serve as a valuable source of information for the preliminary detection of anomalous activities in IaaS environments. Furthermore, the results confirm that events identified through flow log analysis should be further investigated using additional telemetry sources to allow accurate incident assessment and validation.

The analysis conducted confirmed that network flow logs enable effective anomaly detection based on traffic metadata, such as the number of connections, the number of unique destination ports, the volume of transferred data, and the number of flows. In particular, patterns characteristic of port scanning, repeated SSH connections, and potential high-volume data transfers were observed. The analysis also revealed that publicly exposed resources are subject to automated scanning by external systems almost immediately after deployment.

This phenomenon underscores the necessity of continuous network traffic monitoring, even in small and relatively simple cloud environments. Furthermore, the study highlighted the limitations of relying solely on network traffic metadata.

Based solely on flow logs, it is not possible to unambiguously distinguish between legitimate operations, such as system updates or administrative file transfers, and potential attempts at data exfiltration. This underlines the need to treat network traffic analysis as an early warning mechanism that requires further correlation with additional information sources.

Overall, the work demonstrated that the proposed approach allows for the effective detection of atypical communication patterns and provides a solid foundation for the further development of security monitoring systems. In the future, the analysis could be extended to include automatic event classification, correlation with system logs, and application of ML techniques to reduce false positives.

The analysis indicates that even under limited data visibility, it is possible to identify significant deviations from normal network behavior. The results highlight the need to adopt a multisource approach, in which the analysis of network flow logs is only one component of a broader security monitoring system.

Future work will extend this research by incorporating a quantitative evaluation of detection performance, including metrics such as detection rate, false positive rate, and detection time. Another promising direction is the comparative evaluation of anomaly detection using AWS VPC Flow Logs against the native security capabilities within the AWS ecosystem.

References

[1] S. Kanthed, "Monitoring of Cloud Computing Environments: Concepts, Solutions, Trends, and Future Directions", *International Journal on Science and Technology*, vol. 15, pp. 1–16, 2024 (https://doi.org/10.71097/IJSAT.v15.i1.2836).

[2] A. Mycek, "Monitoring, Management, and Analysis of Security Aspects of IaaS Environments", *Journal of Telecommunications and Information Technology*, vol. 94, pp. 108–116, 2023 (https://doi.org/10.26636/jtit.2023.4.1419).

[3] S. Shetty, B. Biswal, and H. Maziku, "Auditing and Analysis of Network Traffic in Cloud Environment", *International Journal of Business Process Integration and Management*, vol. 7, pp. 153–165, 2014 (https://doi.org/10.1504/IJBPIM.2014.063519).

[4] B. Thomas *et al.*, "A Study of Cloud-native Intrusion Detection Using VPC Flow Logs and Ensemble Learning", *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 15, pp. 985–993, 2026 (https://doi.org/10.17148/IJARCCE.2026.151131).

[5] H.B. Illa, "AI-Driven Incident Detection Using AWS CloudWatch and VPC Flow Logs", *South Asian Journal of Engineering and Technology*, vol. 14, pp. 70–85, 2024 (https://doi.org/10.26524/sajet.2024.14.32).

[6] M. Collins *et al.*, "Superflows: A New Tool for Forensic Network Flow Analysis", *arXiv*, 2024 (https://doi.org/10.48550/arXiv.2403.01314).

[7] K. Hsieh *et al.*, "NetVigil: Robust and Low-cost Anomaly Detection for East-West Data Center Security", *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*, pp. 1771–1789, 2024.

[8] Z. Wang *et al.*, "Diagnosing Application-network Anomalies for Millions of IPs in Production Clouds", *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, pp. 885–899, 2024.

[9] M.S. Islam *et al.*, "Anomaly Detection in Large-scale Cloud Systems: An Industry Case and Dataset", *2025 IEEE/ACM 47th International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, Ottawa, Canada, 2025 (https://doi.org/10.1109/ICSE-SEIP66354.2025.00039).

[10] J. Kalibjian, "Security Considerations When Performing Telemetry Post-processing and Analysis in Cloud Environments", *International Telemetering Conference Proceedings*, vol. 58, 2023.

[11] W. Dawoud, I. Takouna, and C. Meinel, "Infrastructure as a Service Security: Challenges and Solutions", *2010 The 7th International Conference on Informatics and Systems (INFOS)*, Cairo, Egypt, 2010.

[12] F. Minna *et al.*, "Understanding the Security Implications of Kubernetes Networking", *IEEE Security & Privacy*, vol. 19, pp. 46–56, 2021 (https://doi.org/10.1109/MSEC.2021.3094726).

[13] A. Mycek and M. Łukaczyk, "Security and Hardening of Kubernetes in Public Clouds: A Comparative Study of EKS, AKS, and GKE", *IEEE Access*, vol. 14, pp. 80990–81009, 2026 (https://doi.org/10.1109/ACCESS.2026.3697610).

[14] T.A.K. Manne, "Leveraging AWS Security Hub and GuardDuty for Continuous Threat Intelligence", *The Journal of Scientific and Engineering Research*, vol. 11, pp. 268–275, 2024 (https://doi.org/10.5281/zenodo.17199795).

[15] A. Mycek, D. Grzonka, and J. Tchórzewski, "Agent-based Simulation and Analysis of Infrastructure-as-Code Process to Build and Manage Cloud Environment", *Proc. of the 37th ECMS International Conference on Modelling and Simulation*, pp. 513–520, 2023 (https://doi.org/10.7148/2023-0513).

[16] A. Mycek and M. Łukaczyk, "Security of Containerization Platforms: Threat Modelling, Vulnerability Analysis, and Risk Mitigation", *Proc. of the 38th ECMS International Conference on Modelling and Simulation*, pp. 585–591, 2024 (https://doi.org/10.7148/2024-0585).

[17] M. Łukaczyk and A. Mycek, "Automation of Security Policies in DevSecOps Environments: Implementation of Zero Trust Principles Using Ansible and Terraform in Linux Systems", *Proc. of the 39th ECMS International Conference on Modelling and Simulation*, pp. 241–247, 2025 (https://doi.org/10.7148/2025-0241).

Hubert Wójcik, B.Eng.
Department of Computer Science
E-mail: hubert.wojcik78@student.pk.edu.pl
Cracow University of Technology, Cracow, Poland
https://www.pk.edu.pl

Mirosław Roszkowski, M.Sc.
CUT Doctoral School, Cracow, Poland
https://orcid.org/0009-0008-4464-9430
E-mail: miroslaw.roszkowski@pk.edu.pl
Cracow University of Technology, Cracow, Poland
https://www.pk.edu.pl

Andrzej Mycek, M.Sc.
CUT Doctoral School, Cracow, Poland
https://orcid.org/0000-0002-2720-6071
E-mail: andrzej.mycek@pk.edu.pl
Cracow University of Technology, Cracow, Poland
https://www.pk.edu.pl

Adaptive Entropy Prediction-based Lossless Compression for Efficient Data Transmission in Terrestrial and Underwater IoT Systems

Ankur Sisodia and Swati Vishnoi

Noida Institute of Engineering and Technology, Greater Noida, India

https://doi.org/10.26636/jtit.2026.3.2711

Submitted: 8 Jul. 2026, Accepted: 20 Jul. 2026

Abstract — Demand for quick lossless data compression systems has become more important recently due to the growing deployment of Internet of Things (IoT) devices in terrestrial and underwater environments, under restricted bandwidth, latency and energy consumption conditions. Current dictionary-based and entropy-driven compression methods depend on reactive adaptation and fixed block processing, which makes them less useful in situations where sensing is constantly changing. This paper presents a self-optimizing entropy prediction-assisted lossless compression framework (EP-SLZW), where the compression technique is based on the expected data redundancy and current channel condition. To achieve optimal results, lightweight entropy prediction is employed in combination with adaptive block segmentation and dual dictionary learning between processing and compression. The optimization component of the proposed method enhances the compression strategy by taking care of all important environmental factors to provide the best solutions for radio frequency and acoustic communication channels. The method is evaluated with the help of the MQTT and CoAP protocols through NetSim simulations and hardware experiments. The results show that the proposed approach performs better in comparison to conventional LZW as well as Huffman and run-length encoding techniques, providing better throughput, compression ratio, and less end-to-end delay when dealing with resource-constrained IoT devices.

Keywords — adaptive dictionary, IoT networks, lossless data compression, prediction of entropy

1. Introduction

The Internet of Things (IoT) has become a crucial technology enabling various intelligent systems, including smart city infrastructure, precision agriculture, industrial control, and environmental monitoring. Systems of this type depend on extensive sensor networks which gather information to support automatic management and intelligent decision processes. IoT networks support multiple communication modes, including terrestrial RF systems and underwater acoustic communication. Networks must process sensor data on a continuous basis, facing restrictions regarding their bandwidth, energy usage, and response time. Low channel capacity, high propagation delays, signal attenuation, and rapidly changing ambient noise levels create additional difficulties for underwater networks. Efficient data processing becomes a critical requirement that operators need to take into consideration in order to maintain integrity of their networks [1], [2].

Lempel-Ziv-Welch (LZW), Huffman data coding, and run-length encoding are traditional compression techniques users widely implement due to their straightforward nature and good performance [3]. These methods need fixed block structures, or require real-time data statistics for their adaptive mechanisms, which makes it difficult to handle the rapid changes in data patterns occurring during sensing operations in IoT. Sensor readings show two main patterns, as they experience periods of strong repeated measurements and unexpected new values that result from environmental changes and various other events. The use of static strategies decreases compression efficiency and leads to higher cost of communication.

Researchers have studied adaptive compression systems that adjust their block size and dictionary formats according to the detected entropy and temporary redundancy patterns. Most existing methods continue to use fixed settings because they perform better than established methods, but they still function as reactive systems changing their parameters after major data transformations [4]. Moreover, systems use identical compression methods on all types of communication channels because they are not capable of categorizing the operational characteristics of terrestrial and underwater communication. A system's inability to anticipate its working environment leads to excessive data retransmissions that extend total response times and waste energy resources, particularly in underwater IoT networks, where retransmission expenses are extremely high [5].

This work presents a novel lossless compression method which uses a self-optimizing system with entropy prediction to solve connection problems between different terrestrial and underwater IoT networks. The proposed solution uses lightweight entropy prediction to facilitate predictive adaptation, which enables the compression engine to adapt to changes in the redundancy of the incoming data. The system

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

achieves an equilibrium between three competing demands, i.e. compression efficiency, computational resources, and energy usage, by implementing predictive adaptation together with dynamic block segmentation and dual dictionary learning. It uses an environment-aware optimization strategy to modify its compression operations based on the distinct radio frequency and acoustic channel characteristics.

The proposed method was evaluated through performance testing that relied on NetSim simulations and hardware tests with multiple IoT sensor nodes using the MQTT and CoAP communication protocols. The suggested approach is compared with existing lossless compression techniques using parameters such as compression ratio, throughput, end-to-end delay, and computational complexity. The experimental results show that the proposed EP-SLZW method performs better than the existing approaches while being suitable for resource-constrained IoT devices. The research work presents the following main contributions, which are described below:

1) A self-adaptive design with lightweight entropy prediction is implemented to forecast data redundancy changes during different sensing conditions. This capability is used to adjust the compression method before the data stream experiences significant statistical changes.
2) A predictive block size control method is proposed that regulates data compression by monitoring changes in entropy patterns. This approach is used to make intelligent decisions about block size selection, which leads to better compression results while keeping extra processing costs at a minimum.
3) Local and global dictionaries are implemented to manage memory consumption and speed of convergence while achieving the appropriate compression ratio for unevenly distributed data.
4) The framework uses its control system to switch between different data compression methods, depending on the terrestrial radio frequency and underwater acoustic channels.
5) The proposed framework is evaluated through NetSim simulations and testing on actual multi-sensor IoT hardware which uses both MQTT and CoAP protocols. The system demonstrates improved compression ratios, throughput and end-to-end delay performance compared to traditional LZW and Huffman coding and run-length encoding methods.

2. Related Works

Data compression is a key method to make communication in IoT networks more efficient, especially when bandwidth and energy are limited. The early data processing relied on classical lossless algorithms, including Huffman coding, run-length encoding and Lempel-Ziv-Welch (LZW), as these algorithms offer straightforward implementation and produce consistent decoding results [6]. Such methods are used in low-power sensing platforms, but their effectiveness depends on the data characteristics and the specific configuration of the sensor blocks. Sensors in multiple IoT applications demonstrate statistical patterns that change over time. The typical applications of fixed block sizes and static dictionary structures face reduced effectiveness in this concept [7].

Researchers have proposed several adaptive compression techniques that adjust block size and encoding settings based on actual entropy measurements and symbol distribution patterns. The compression ratio for image and sensor data streams has improved through entropy-based block segmentation, which works best in regions that exhibit repetitive patterning [8]. Compressive sensing was also explored together with block-based adaptive coding methods for transmitting underwater images and enhancing the performance of acoustic sensor networks under restricted bandwidth conditions. Other methods adjust compression parameters when substantial alterations occur in data patterns, leading to enhanced efficiency in data reduction [9]. This leads to temporary performance issues because it takes time to react, which affects applications that depend on instant detection of quick changes in data.

Researchers have shown great interest in finding better ways to improve dictionary-based compression methods. The development of LZW and LZ77 compression algorithms which use dynamic dictionary pruning, context modelling and memory-aware management, helps achieve faster convergence and improved compression [10]. Sliding-window dictionaries are researched as a solution to control memory usage, while other work examined hierarchical or clustered dictionaries for IoT systems that operate in distributed environments. Most dictionary optimization methods still need to use local data, since they cannot handle long-term data patterns that occur between data blocks or forecast upcoming data trends. The system becomes less adaptable, because it cannot handle multiple data types that exist together in different environments [11].

An underwater IoT (U-IoT) network has emerged recently as a research domain due to the fact that these networks face particular communication challenges, including extended propagation delays, high signal loss, and restricted transmission capacity. Underwater environments rely on compression methods that prioritize energy efficiency and latency reduction over maximum compression ratio [12]. In the literature, three different approaches are presented, including acoustic channel-aware encoding schemes, predictive data aggregation, and hybrid compression-routing methods to decrease retransmissions and packet loss. Most of these methods depend on specific physical layer assumptions and routing protocols that restrict their application to mixed terrestrial and underwater operations [13].

Current research investigates data compression methods that employ basic forecasting models and neural predictors to determine future data behavior [14]. Most existing predictive systems currently work to estimate traffic patterns or develop optimal routing solutions that do not integrate with the data compression process. This research fails to investigate dictionary adaptation techniques and compression policies tailored to specific environments, creating opportunities for better optimization across all system layers and different operational environments [15], [16].

Tab. 1. Comparison of existing IoT compression approaches.

Method	Adaptive block size	Entropy prediction	Dictionary learning	Environment aware (terrestrial, underwater)	Energy and delay optimization	Key limitations
Run-length encoding (RLE)	No	No	No	No	No	Poor performance for non-repetitive data
Huffman coding	No	No	No	No	Partial	Fixed coding tree, sensitive to data distribution
Classical LZW	No	No	Static	No	Partial	The dictionary grows blindly, no anticipation
Entropy-based adaptive compression	Yes	No	Limited	No	Partial	Reactive adaptation only
Compressive sensing (IoUT)	Yes	No	No	Partial	Yes	High reconstruction complexity
Predictive data aggregation	No	Yes	No	Partial	Yes	Not integrated with compression stage
Dynamic dictionary LZ variants	Partial	No	Yes	No	Partial	No environment awareness
Adaptive LZW	Partial	No	Adaptive	No	Partial	Limited adaptation to communication environments
Trie-based shared dictionary compression	No	No	Shared trie dictionary	No	Partial	Higher memory requirement and synchronization overhead
Proposed	Yes	Yes	Dual (local+global)	Yes	Yes	—

While recent studies propose enhanced compression techniques such as dynamic dictionary learning, dictionary compression, predictive aggregation of data, and advanced lightweight entropy-based encoding developed specifically for the edge and IoT architectures, there is still an inherent necessity for additional memory or computing power, or for pre-learning which is problematic in the case when the microcontroller capabilities are limited. Table 1 shows a comparison of existing IoT compression approaches.

3. System Model and Problem Formulation

The study investigates IoT networks containing various types of sensor nodes that operate in both terrestrial and water environments. The terrestrial segment uses radio frequency (RF) communication, while the underwater segment uses an acoustic or short-range optical link for sending data. Sensor nodes which form group $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ operate continuously to monitor environmental conditions by measuring temperature, humidity, pressure, light intensity, and distance. The nodes transmit their detected data to a sink or gateway node by using multi-hop or single-hop communication through MQTT and CoAP.

The network function during time intervals $t = 1, 2, \ldots$, which each node uses to generate its data stream is:

$$D_i(t) = \{x_1, x_2, \ldots, x_{L_t}\} , \tag{1}$$

where L_t is the number of samples taken at time t.

3.1. Data and Channel Model

The data stream being detected operates like a separate system that depends on memory functions, while its statistical characteristics develop over time due to the fact that environmental factors and event-based behavioral patterns evolve. The instantaneous entropy of a data block B_t at time t is defined as:

$$H(B_t) = -\sum_{k=1}^{K} p_k \log_2 p_k , \tag{2}$$

where p_k represents the probability of symbol k appearing in B_t and K defines the total number of symbols in the alphabet.

The communication channel exhibits major variations which depend on different environmental conditions. The terrestrial IoT system operates with a data transmission capacity as follows:

$$R_t = B_t \log_2(1 + \mathrm{SNR}_T) , \tag{3}$$

where B_T is the channel bandwidth and SNR_T is the signal-to-noise ratio. Underwater data transmission capabilities operating at lower speeds are defined as:

$$R_U = B_U \log_2(1 + \mathrm{SNR}_U) , \tag{4}$$

with conditions $B_U \ll B_T$ and $\mathrm{SNR}_U < \mathrm{SNR}_T$.

3.2. Compression Model

Every node applies a lossless compression function $\mathcal{C}(\cdot)$ before sending data. The original data size is represented by S_o bits while the compressed data size is represented by S_c. The compression ratio is calculated as:

$$C_R = \frac{S_o}{S_c} .$$

The duration of compression processing is represented by T_c and the transmission time is calculated as:

$$T_{tx} = \frac{S_c}{R} .$$

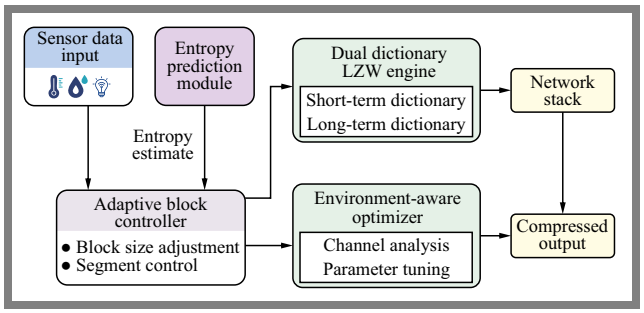


Fig. 1. EP-SLZW framework architecture.

The setting of R determines which value from $R \in \{R_T, R_U\}$ to use. The complete end-to-end delay calculation establishes $T_{e2e} = T_c + T_{tx} + T_{prop}$, where T_{prop} represents the duration required for data to traverse through intermediate nodes.

3.3. Energy Consumption Model

The total energy consumption for each data block is as follows:

$$E_{total} = E_{comp} + E_{tx} \,, \tag{5}$$

where compression energy is $E_{comp} = P_{cpu} \cdot T_c$, and transmission energy is $E_{tx} = P_{tx} \cdot T_{tx}$.

The system uses P_{cpu} as the CPU power consumption and P_{tx} as the transmission power consumption. Underwater networks use higher P_{tx} to operate due to specific environment conditions.

3.4. Problem Formulation

The compression framework manages three components which determine its performance. The optimization problem is stated as max C_r min T_{e2e}, E_{total} and

$$\mathcal{D}\big(\mathcal{C}(D_i(t))\big) = D_i(t), \;\; \forall i, t \,, \tag{6}$$

where:

- $B_{min} \leqslant B_t \leqslant B_{max}$,
- $T_c \leqslant T_{max}$,
- $\mathcal{D}(\cdot)$ is the function that decompresses,
- B_t is the adaptive block size,
- B_{min} and B_{max} are the limits on the size of blocks that can be used,
- T_{max} is the longest time that a real-time application can wait for compression.

4. Proposed Framework

The proposed entropy prediction-assisted self-optimizing LZW (EP-SLZW) framework is a data processing layer for sequential block raw sensor streams. It combines entropy prediction, adaptive block segmentation, dual-dictionary compression, and environment-aware optimization into a single system, as shown in Fig. 1.

Initially, sensor readings are buffered and put into preliminary data blocks. The system determines present entropy H_t value for every block at time t to predict future data redundancy changes. EP-SLZW uses more than just reactive entropy assessments, as it implements a basic forecasting system relying on an exponential weighted moving average (EWMA) model to predict subsequent block entropy in the following manner:

$$\hat{H}_{t+1} = \alpha H_t + (1 - \alpha)\, \hat{H}_t \,, \tag{7}$$

where H_t is determined entropy and $\hat{H}_t$ is previous predicted entropy.

The range of $0 < \alpha < 1$ determines the degree to which the predictor reacts to new information. The predicted entropy helps the framework prepare for changes in data randomness by varying the compression settings to keep things running smoothly until performance drops excessively. The block segmentation strategy uses the predicted entropy value $\hat{H}_{t+1}$, to determine the size of the upcoming data block. The current block size is represented by B_t. The update rule is as follows:

$$B_{t+1} = \begin{cases} \min(B_t + \Delta, B_{max}), & \text{if } \hat{H}_{t+1} < \tau \\ \min(B_t - \Delta, B_{min}), & \text{if } \hat{H}_{t+1} \geqslant \tau \end{cases} \,, \tag{8}$$

The operational constraints of the proposed framework are determined by the relationship between entropy threshold τ and block adjustment factor Δ, which define the range of the block size bounded by B_{min} and B_{max}. The data stream becomes extremely redundant when entropy values drop below the threshold, which leads to selecting larger block sizes that maximize dictionary reuse. The framework selects smaller block sizes, because rising predicted entropy values require dictionary size control, which accelerates compression to prevent performance drops during times of high data randomness.

The proposed framework achieves its primary extension through the implementation of dual-dictionary LZW compression, which provides exceptional flexibility to handle diverse data patterns. Two dictionaries are used – as a local dictionary and a global dictionary. The local dictionary reset function at block start enables quick adaptation to unexpected changes that occur in sensor data patterns. On the other hand, the global dictionary is retained over several blocks, which enables the detection of slowly varying or periodic sensor data patterns over long time periods. The dictionary selection process uses a normalized entropy factor that defines the process as:

$$\eta_t = \frac{H_t}{H_{max}} \,, \tag{9}$$

where H_{max} denotes the maximum possible entropy of the symbol alphabet.

Based on this metric, the dictionary selection rule is given by:

$$\text{Dictionary} = \begin{cases} \text{global}, & \eta_t < \gamma \\ \text{local}, & \eta_t \geqslant \gamma \end{cases} \,,$$

The two operational modes for EP-SLZW are activated through γ which acts as the boundary between these two modes. The algorithm uses two types of redundancy patterns to store data. The dictionary size increases only during periods in which system activity becomes less predictable,

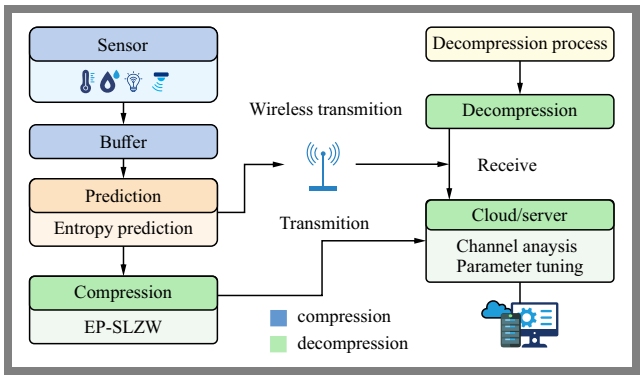


Fig. 2. Data flow of the proposed compression-decompression process.

because this method enables to store both temporary and permanent data. The method uses an environment indicator that helps it understand the different IoT communication environments. The numerical values of all hyperparameters (α, τ, Δ, and γ) and the methodology used for their selection are presented in Sec. 6.

Terrestrial communication systems use RF signals which assign an indicator value of $\lambda = 0$. Underwater communication systems which use acoustic or optical links assign an indicator value of $\lambda = 1$. The approach achieves its target goal through high processing throughput combined with moderate compression latency. Aggressive data compression techniques are used for underwater communication to save energy and bandwidth. The system uses environmental conditions to dynamically adjust its values τ, Δ, and γ.

The receiver uses standard LZW decompression to extract data from compressed blocks, which includes synchronous updates of their dictionaries. The proposed compression framework maintains lossless data recovery, which ensures accurate retrieval of original data. The compressed metadata inform the decoder which dictionaries to use, which lets it do the same thing with two dictionaries. Block decompression enables the restoration of original data streams $D' = \mathcal{D}(\mathcal{C}(D)) = D$.

The data processing scheme of the proposed compression-decompression process is presented in Fig. 2.

Algorithm 1 Proposed compression framework

Input: Sensor data stream D, initial block size B_0, block size bounds B_{min}, B_{max}, entropy threshold τ, dictionary switching threshold γ, prediction factor α, environment indicator $\lambda \in \{0, 1\}$
Output: Compressed data stream C
Initialize:

Set $t \leftarrow 0$
Set $B_t \leftarrow B_0$
Global dictionary D_g
Set predicted entropy $\hat{H}_0 \leftarrow H_0$
Compressed stream $C \leftarrow \emptyset$
while $D \neq \emptyset$ **do**
    Extract block B_t from D of size B_t
    Compute the block entropy as
    $H(t) = -\sum_{k=1}^{K} p_k \log_2 p_k$
    Predict the next entropy using EWMA by Eq. (7)
    Compute the normalized entropy by Eq. (10)
Dictionary selection
    if $\eta_t < \gamma$ **then**
        select global dictionary D_g
    else
        initialize local dictionary D_l
    end if
    Block size adaptation by Eq. (8)
Tunning of environment-aware parameter
    if $\gamma = 1$ (underwater) **then**
        increase τ, decrease Δ
    else
        maintain default values
    end if
    Compress block B_t using selected dictionary with LZW procedure to obtain C_t
    Append C_t to C
    Update predicted entropy $\hat{H}_t \leftarrow \hat{H}_{t+1}$
    Increment $t \leftarrow t + 1$
end while
Return compressed stream C

5. Computational Complexity and Energy Analysis

This section evaluates the proposed EP-SLZW framework by assessing its memory requirements, processing demands and energy consumption in order to determine its suitability for IoT devices with restricted resources operating in both terrestrial and underwater environments. The size of the data block is defined by its symbol count n and its dictionary size M. The first block needs to be scanned to determine the symbol frequency because this information is necessary to calculate the entropy for each block. The process requires $O(n)$ time to complete. The prediction of entropy is based on the exponential weighted moving average model.

The dual-dictionary LZW compression process is the most important part of the framework. The dictionary operations use hash-based indexing enable dictionaries to process every symbol in their contents during dictionary lookup and insertion functions at a fixed time interval. The compression process requires $O(n)$ time, as it handles all blocks of data. The LZW decoding process requires standard decompression to complete in $O(n)$ time, which operates at linear speed based on the block size. The framework processes each block with a total computational load of $O(n)$ which matches the standard LZW and Huffman methods, because they both show the same asymptotic performance. The additional prediction and control systems add minimal workload to the system while maintaining its ability to scale efficiently.

The memory needed for EP-SLZW includes space for the input block, entropy statistics, and two dictionaries. Let M_g and M_l be the sizes of the global and local dictionaries, respectively. The total amount of used memory is $O(n + M_g + M_l)$. The local dictionary is cleared after each block operation.

Algorithm 2 Proposed decompression framework

Input: Compressed data stream C, initial block size B_0, block size bounds B_{min}, B_{max}, dictionary switching threshold γ, entropy prediction factor α, environment indicator $\lambda \in \{0, 1\}$
Output: Sensor data stream D' is reconstructed
Initialize:

Set $t \leftarrow 0$
Set $B_t \leftarrow B_0$
Initialize global dictionary D_g
Set predicted entropy $\hat{H}_0 \leftarrow H_0$
Initialize reconstructed stream $D' \leftarrow \emptyset$
while compressed stream C is not empty **do**
    Extract compressed block C_t from C
    Retrieve the dictionary selection flag from metadata of C_t
  Dictionary initialization
    if flag indicates global dictionary **then**
        use D_g
    else
        initialize local dictionary D_l
    end if
    Apply the standard LZW decoding on C_t using selected dictionary to obtain block B_t
    Append B_t to reconstructed stream D'
    Compute the block entropy using:
    $H(t) = -\sum_{k=1}^{K} p_k \log_2 p_k$
    Predict the next entropy using EWMA by Eq. (7)
    Block size adaptation by Eq. (8)
    Update predicted entropy $\hat{H}_t \leftarrow \hat{H}_{t+1}$
    Increment $t \leftarrow t + 1$
end while
Return compressed stream D'

The system maintains memory balance by periodically removing elements from the global dictionary. The memory consumption remains controlled since the system limits its usage to microcontroller platforms which operate with minimal power in common IoT devices. The total energy used per block is modeled in the following way:

$$E_{total} = E_{comp} + E_{tx} \ , \tag{10}$$

where E_{comp} represents the energy required for compression work and E_{tx} denotes the transmission energy used for wireless data transmission.

The energy required for compression work is:

$$E_{comp} = P_{cpu} \cdot T_c \ . \tag{11}$$

The energy used for transmission is determined using P_{cpu} being the average amount of power that a processor uses and T_c – the time it takes to compress each block. The energy used for transmission is derived as follows:

$$E_{tx} = P_{tx} \cdot \frac{S_c}{R} \ . \tag{12}$$

The formula describes transmission power P_{tx} as the product of S_c (representing compressed data size) and R (denoting the maximum achievable channel data rate).

The processing power requirements of EP-SLZW increase, because the scheme needs additional resources for entropy prediction and dictionary management, yet its superior compression capabilities result in S_c being reduced. The underwater IoT environments demonstrate this effect with increased strength

$$P_{tx}^{(U)} \gg P_{tx}^{(T)} \ , \ \ R_U \ll R_T \ .$$

The system uses U and T as superscripts to identify underwater and terrestrial environments. Higher energy savings through are achieved data volume reduction, as EP-SLZW transmits less data than it needs to send at E_{tx} while E_{comp} experiences only a minimal rise:

$$E_{total}^{EP-SLZW} < E_{total}^{baseline} \ .$$

The adaptive block size enables precise management of dictionary size changes that occur during times of high data stream entropy, since it prevents dictionary expansion during random sensing intervals. The system maintains minimal processor usage, while it solves energy stability problems which emerge when data patterns occur abruptly.

6. Experimental Setup

To evaluate the performance of the EP-SLZW framework, we used two testing methods, i.e. simulation testing and hardware testing, across both terrestrial and underwater IoT environments. The NetSim simulator executes network simulations to create large-scale IoT network models that assess system performance under various operational conditions. Simulation tests use a 1000 × 1000 m sensing area while testing multiple sensor node configurations (ranging from 50 to 500) to determine how network density affects its performance.

Sensor nodes are deployed randomly inside the sensing area and send data using a fixed rate of five packets (carrying 1024 bytes) per second (Figs. 3, 4). The study evaluates how application layer communication protocols impact compression performance through the use of MQTT and CoAP protocols. The terrestrial network maintains a constant wireless bandwidth of 2 Mbps throughout the entire simulation period which extends from 250 s to 750 s, as shown in Tab. 2. Underwater simulations test different physical layer settings to create an acoustic channel simulating reduced bandwidth, extended propagation delay, and increased packet loss rates. Identical routing and medium access conditions throughout their experiments are maintained to enable unbiased assessment of compression algorithm performance.

The proposed framework functions at the sensor node layer, and its effectiveness gets tested through a comparison with standard lossless compression techniques, i.e. LZW and Huffman coding and run-length encoding (RLE). The NetSim protocol stack includes all algorithms that operate with identical configuration settings for packet creation, network routing and data storage.

An additional evaluation was performed using a physical IoT testbed comprising a low-power microcontroller platform, an Arduino Uno board with multiple sensors (including

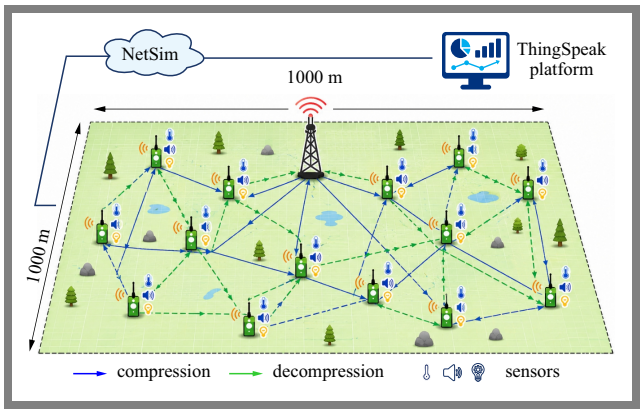


Fig. 3. Topology of terrestrial IoT simulation (NetSim).

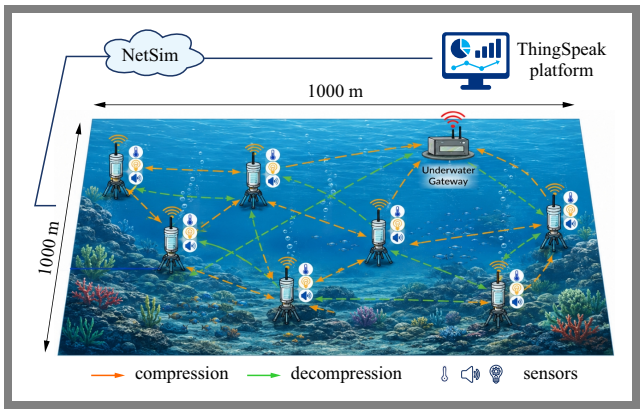


Fig. 4. Underwater IoT simulation topology (NetSim).

a temperature-humidity, an ultrasonic distance, and a light intensity sensor) (Tab. 3). The framework operates through periodic sensor reading that is processed at the current location before forwarding the data. The research simulates restricted communication environments through an infrared transceiver module which enables data to be transferred wirelessly at short distances, since RF signals face challenges in underwater transmission. The receiver system transmits uncompressed data to an ESP8266 Wi-Fi module that forwards the information to a cloud server, where it may be accessed by the ThingSpeak platform. Such an approach enables real-time data logging and visualization.

The same hardware setup is used for terrestrial and underwater tests. Underwater conditions sensor modules and IR transceivers are positioned inside waterproof cases that partially submerge to simulate underwater noise and attenuation effects.

Although the infrared (IR) transceiver employed in the hardware prototype in the underwater communication experiment can be attributed to the use of the channel model, it is not aimed at simulating all the properties of the underwater acoustic channel. Instead, it is supposed to serve as an illustration of the application of the compression framework in the underwater communication problem with limited transmission distance, attenuation, and bandwidth. Therefore, features such as delay, attenuation, and multipath propagation are modeled separately in NetSim. Consequently, the outcomes of the experiment can be interpreted as the demonstration of the application of the compression framework, and not the complete underwater acoustic communication system.

Tab. 2. Simulation parameters for terrestrial and underwater IoT networks.

Parameter	Value
Simulation platform	NetSim
Simulation area	1000 × 1000
Number of sensor nodes	50 – 500
Node deployment	Random
Application protocols	MQTT, CoAP
Packet size	1024 bytes
Packet generation rate	5 packets/s
Simulation time	250 – 750 s
Terrestrial bandwidth	2 Mbps
Underwater bandwidth	Low (acoustic channel model)
Routing configuration	Fixed for all experiments
Compression algorithms	EP-SLZW, LZW, Huffman, RLE
Performance metrics	Compression ratio, throughput, end-to-end delay, compression time
Channel type	RF (terrestrial), acoustic (simulation), IR-based proof-of-concept (hardware)

Tab. 3. Hardware testbed configuration.

Component	Model / Specification	Purpose
Microcontroller	Arduino Uno (ATmega328P)	Sensor interfacing and compression processing
Temperature–humidity sensor	DHt11	Environmental data acquisition
Ultrasonic sensor	HC-SR04	Distance measurement
Light sensor	LDR module	Light intensity detection
IR transmitter	940 nm IR LED module	Short-range wireless data transmission
IR receiver	TSOP38238 / equivalent	Short-range wireless data reception
Display module	LCD 16 × 2	Local monitoring of sensor readings
Wi-Fi module	ESP8266	Cloud connectivity (ThingSpeak)
Power supply	5 V regulated + battery backup	Stable system operation
Cloud platform	ThingSpeak	Data storage and visualization
Communication protocols	MQTT, CoAP	Application-layer data transmission

Four metrics are used to evaluate performance: compression ratio, throughput, end-to-end delay, and compression time. The experiments were carried out 30 times, in an independent manner, with each iteration relying on different seeds. Nevertheless, the same network configurations and traffic conditions were applied. The data shows the average standard deviation (SD) values of the results of 30 independent experiments. A confidence interval (CI) of 95% was also estimated for all metrics. Finally, the paired Student t-test was performed on the results of repeated experiments to assess the significance of the results obtained with respect to the baseline methods.

Tab. 4. Hyperparameter configuration of EP-SLZW.

Parameter	Value	Description	Selection method
α	0.35	EWMA smoothing coefficient	Grid search on validation dataset
τ	0.55	Entropy threshold for adaptive block sizing	Validation experiments
Δ	64 bytes	Block size adjustment step	Sensitivity analysis
γ	0.60	Threshold for local/ global dictionary switching	Empirical tuning
B_{min}	256 bytes	Minimum allowable block size	Memory constraint of sensor nodes
B_{max}	4096 bytes	Maximum allowable block size	Compression efficiency and latency trade-off

6.1. Hyperparameter Configuration

To ensure repeatability of the results obtained during the simulations and to achieve an optimal balance between compression ratio, processing time, and delivery delay, the parameter tuning has been performed. Several parameter values have been tested on the validation set, and the final combination has been chosen according to the maximum compression rate along with minimum processing time and delay. Table 4 shows the hyperparameter values utilized during all simulations and hardware tests.

7. Results and Discussion

The proposed system achieves higher compression ratios and better throughput efficiency while reducing end-to-end latency compared to all baseline algorithms tested. Both terrestrial RF networks and underwater communication systems with restricted bandwidth prove effective with the developed improvements which work in simulation environments and physical testing facilities. The EP-SLZW system achieved an average compression ratio of 7.68 during terrestrial simulation experiments, as shown in Tab. 5. The average compression ratios for Huffman coding, LZW and RLE equaled 4.87, 3.92, and 2.41, respectively. Throughput exceeds standard LZW performance by 93% and RLE performance by nearly 200%. The end-to-end delay time is 21% shorter when using Huffman coding and 26% shorter when using LZW, in comparison to other methods. The compression process requires additional time because of the need to execute operations that predict entropy while handling dictionary management tasks.

The underwater simulation results, shown in Tab. 6, illustrate how performance differences between systems become more evident. EP-SLZW achieves a compression ratio of 9.41, which exceeds (two-fold) the compression ratio of Huffman coding, yet reaches almost three times the compression ratio of RLE. The system provides throughput performance improved by more than 115%, exceeding the throughput of LZW, and it delivers end-to-end delay which exceeds that of RLE and LZW by 55 and 44%, respectively. The results demonstrate that underwater environments experiencing low channel capacity and high retransmission costs benefit the most from aggressive data size reductions.

Tab. 5. Performance comparison in terrestrial simulation (average values).

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.41	0.98	1820	940
Huffman	4.87	1.52	1340	1710
LZW	3.92	1.36	1495	1540
EP-SLZW	7.68 ±0.12	2.94 ±0.08	1185 ±28	1825 ±42

Tab. 6. Performance comparison in underwater simulation (average values).

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.05	0.62	8120	960
Huffman	4.12	1.05	5280	1830
LZW	3.46	0.89	6125	1615
EP-SLZW	9.41 ±0.14	1.91 ±0.07	2715 ±25	3085 ±45

Tab. 7. Performance comparison using a terrestrial hardware testbed.

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.36	0.83	2150	1015
Huffman	4.54	1.27	1610	1935
LZW	3.71	1.12	1785	1680
EP-SLZW	6.92 ±0.11	1.88 ±0.09	1325 ±24	2150 ±41

Tab. 8. Performance comparison using an underwater hardware testbed.

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	1.98	0.59	4980	1120
Huffman	3.87	0.94	3510	2040
LZW	3.25	0.86	3820	1815
EP-SLZW	6.47 ±0.16	1.54 ±0.06	2630 ±23	2890 ±43

The proposed framework shows increased stability under actual conditions, and hardware tests demonstrate its performance during sensor noise processing limits and unstable wireless links. The method maintains a compression ratio of 6.92 in land tests (Tab. 7), while achieving a latency that is 26% lower than that of LZW. 79% higher throughput is achieved than in the case of LZW in underwater hardware tests (Tab. 8), with delay reduced by 31%.

7.1. Interpretation of performance gains

The designed system improves performance, as it enables better handling of smaller data payloads. Smaller data packets use less bandwidth, which means that there are fewer network collisions and a lower chance of having to resend data. The experimental results demonstrate that underwater IoT networks maintain consistent performance across different packet sizes, which shows their ability to handle various traffic patterns. The experimental results demonstrate that the

Tab. 9. Ablation study results.

Configuration	Compression ratio	Throughput [kbps]	End-to-end delay [ms]
Fixed block + single dictionary	3.85	1.34	1510
Adaptive block only	5.21	1.87	1375
Adaptive block + dual dictionary	6.43	2.42	1260
Without entropy prediction	6.71	2.63	1235
Without environment awareness	7.12	2.72	1274
EP-SLZW	7.68	2.94	1185

proposed framework achieves energy savings that exceed the energy expenses that EP-SLZW introduces. Underwater IoT networks benefit from this because wireless data transmission requires more energy than local computation.

7.2. Ablation Study

To assess the impact of each of the key components in the proposed framework, an ablation study was conducted using the terrestrial simulation setup (Tab. 9).

The system demonstrates decreased processing efficiency along with increased execution latency when the compression algorithm lacks entropy prediction, which shows that dynamic data environments need predictive adaptation. To continue with the analysis of the environment-adaptive optimization system, the environment-adaptive setting was turned off while the remaining elements, such as entropy value estimation, the blocks of adaptive splitting, and dual dictionary compression, remain working. In this case, it should be noted that the compression parameters of both terrestrial and underwater communication channels remained the same, regardless of their specifications. One may notice from Tab. 9 that the algorithm with the disabled environment-adaptive module managed to compress the data achieving a compression ratio of 7.68, instead of 7.12, which also allowed achieving the throughput of 2.94 kbps compared with 2.72 kbps when the environment-adaptive module was in use. The average end-to-end delay has increased from 1185 ms in the case of the disabled environment-adaptive module to 1274 ms when the algorithm with enabled environment-adaptive optimization technique was used. Therefore, using the adaptive optimization system improved overall system performance. The contribution of the adaptive optimization system to the increase in overall throughput equaled 8%, while the change of the average end-to-end delay amounted to 7%. The proactive adaptation methods of the EP-SLZW framework combat the performance decline that typically occurs during high-rate data transmissions.

To assess if the improvements in performance obtained did not occur due to the randomness of the experiments, all simulations and experiments were carried out 30 times under the same conditions, but with different random seeds. The mean standard deviation, and 95% confidence interval for all the metrics were computed. The paired t-test was then carried out for the proposed EP-SLZW method and the old LZW algorithm, which served as a benchmark. The p-values for all metrics remained below the level of 0.05, indicating that the improvement obtained with EP-SLZW is statistically significant, as seen in Tab. 10.

Tab. 10. Statistical significance analysis.

Performance metric	Mean improvement	95% CI	p-value
Compression ratio	95.9%	93.8 – 98.0	<0.001
Throughput	116.2%	112.4 – 119.7	<0.001
End-to-end delay	20.7%	18.9 – 22.5	<0.001
Compression time	+18.5%	16.8 – 20.1	0.003

The low values of standard deviation and the narrow confidence intervals prove that the proposed method is functionally stable. Furthermore, the results of the statistical tests indicate that the advantages in the compression rates, speed of operation, and end-to-end lag are statistically significant at a 95% level of confidence.

7.3. Comparison with Recent Adaptive Compression Methods

Besides the traditional lossless encoding methodologies that have been investigated through experimental examination, new compression approaches with adaptive dictionaries have emerged lately. These include adaptive LZW encoding methods and common dictionaries with tries for edge computing. However, while these methods utilize adaptive memory allocation or common dictionaries, they do not adopt a predictive technique for evaluating entropy and communication environment awareness. Additionally, their high memory requirements and elevated synchronization overhead render them unsuitable for use in IoT sensor nodes due to resource constraints. The proposed method employs optimization methods for entropy, adaptive segmentation, and environmental awareness while ensuring computational complexity of $O(n)$.

8. Conclusion and Future Work

The proposed method uses multiple techniques that include entropy prediction and adaptive block segmentation together with dual-dictionary learning and environment-aware optimization to create a system that automatically adapts to different data and communication scenarios. The system operates by relying on mechanisms differing from those of standard reactive compression approaches. Three essential components are established, including a system model and an algorithmic design, together with a method for measuring analytical complexity and energy consumption, to construct the foundational framework.

The EP-SLZW method demonstrates better results than traditional LZW and Huffman coding and run-length encoding, as it achieves better compression ratios, higher throughput and shorter end-to-end delays according to multiple simulation studies and actual hardware tests. The results demonstrated that the framework performs best in underwater IoT environ-

ments, since these situations require urgent data transmission to manage their limited bandwidth and expensive energy consumption. Ablation analysis demonstrated that all architectural components contribute to a significant performance improvement of the entire system. The dynamic sensing environment requires active dictionary management, for it directly impacts the performance of the system.

The algorithm requires additional computational resources, due to the need to perform entropy estimation tasks and implement predictive control functions. The processing demands of this system remain acceptable for most IoT microcontroller platforms. However, upcoming research will focus on achieving greater cost reductions through the implementation of lightweight machine learning predictors and hardware-assisted dictionary management systems.

The current work suffered from specific constraints, as it has used the infrared communication module to conduct the hardware testing of the system in underwater conditions. Although this approach validates the applicability of this compression method, the nature of communication in the underwater environment cannot be fully simulated. Future considerations will focus on testing the performance of the EP-SLZW method, i.e. on conducting the tests under real underwater circumstances, using acoustic modems. Furthermore, the comparison made in the current project was limited to traditional lightweight lossless compression algorithms, because the aim was for the approach to be adopted in low-power microcontroller systems in the first place.

References

[1] K. Sharma and S.K. Shivandu, "Integrating Artificial Intelligence and Internet of Things (IoT) for Enhanced Crop Monitoring and Management in Precision Agriculture", *Sensors International*, vol. 5, art. no. 100292, 2024 (https://doi.org/10.1016/j.sintl.2024.100292).

[2] Z. Li *et al.*, "Recent Progress on Underwater Wireless Communication Methods and Applications", *Journal of Marine Science and Engineering*, vol. 13, art. no. 1505, 2025 (https://doi.org/10.3390/jmse13081505).

[3] A. Hassan *et al.*, "Arithmetic N-gram: an Efficient Data Compression Technique", *Discover Computing*, vol. 27, art. no. 1, 2024 (https://doi.org/10.1007/s10791-024-09431-y).

[4] B.A. Lungisani, A.M. Zungeru, C. Lebekwe, and A. Yahya, "Autoencoder-based Image Compression for Wireless Sensor Networks", *Scientific African*, vol. 24, art. no. e02159, 2024 (https://doi.org/10.1016/j.sciaf.2024.e02159).

[5] A. Pal *et al.*, "Communication for Underwater Sensor Networks: A Comprehensive Summary", *ACM Transactions on Sensor Networks*, vol. 19, pp. 1–44, 2022 (https://doi.org/10.1145/3546827).

[6] V. Alevizos *et al.*, "A Logarithmic Compression Method for Magnitude-rich Data: The LPPIE Approach", *Technologies*, vol. 13, art.no. 278, 2025 (https://doi.org/10.3390/technologies13070278).

[7] J.C. Gambiroža,T. Mastelić, I.K. Nižetić, and M. Čagalj, "Lost in Data: Recognizing Type of Time Series Sensor Data Using Signal Pattern Classification", *International Journal of Data Science and Analytics*, vol. 20, pp. 397–408, 2025 (https://doi.org/10.1007/s41060-023-00413-9).

[8] V. Kovtun, "Adaptive Information-constrained Mapping for Feature Compression in Edge AI and Federated Systems", *Scientific Reports*, vol. 15, art. no. 30915, 2025 (https://doi.org/10.1038/s41598-025-16604-2).

[9] H. Hu *et al.*, "Full-process Adaptive Encoding and Decoding Framework for Remote Sensing Images Based on Compression Sensing", *Remote Sensing*, vol. 16, art. no. 1529, 2024 (https://doi.org/10.3390/rs16091529).

[10] S.A. Alwahab, "New Technology to Improve an Image Compression Using the LZW Algorithm", *Al-Iraqia Journal for Scientific Engineering Research*,vol. 2, pp. 41–46, 2023 (https://doi.org/10.58564/IJSER.2.3.2023.86).

[11] R. Adam *et al.*, "Lossless Compression with Trie-based Shared Dictionary for Omics Data in Edge-cloud Frameworks", *Journal of Sensor and Actuator Networks*, vol. 14, art. no. 41, 2025 (https://doi.org/10.3390/jsan14020041).

[12] S. Chen *et al.*, "Application of Optical Communication Technology for UAV Swarm", *Electronics*, vol. 14, art. no. 994, 2025 (https://doi.org/10.3390/electronics14050994).

[13] W. Fang *et al.*, "Energy Efficient Unified Computing Framework for Smart Grids with AI-driven Communication, Supercomputing, and Energy Perception Orchestration", *Sustainable Computing: Informatics and Systems*, vol. 49, art. no. 101289, 2026 (https://doi.org/10.1016/j.suscom.2025.101289).

[14] A. Fathalla, K. Li, A. Salah, and M.F. Mohamed, "An LSTM-based Distributed Scheme for Data Transmission Reduction of IoT Systems", *Neurocomputing*, vol. 485, pp. 166–180, 2022 (https://doi.org/10.1016/j.neucom.2021.02.105).

[15] M. Fira, H.N. Costin,and L. Goras, "A Study on Dictionary Selection in Compressive Sensing for ECG Signals Compression and Classification", *Biosensors*, vol. 12, art. no. 146, 2022 (https://doi.org/10.3390/bios12030146).

[16] R. Bender-Salazar, "Design Thinking as an Effective Method for Problem-setting and Needfinding for Entrepreneurial Teams Addressing Wicked Problems", *Journal of Innovation and Entrepreneurship*, vol. 12, art. no. 24, 2023 (https://doi.org/10.1186/s13731-023-00291-2).

Ankur Sisodia, Ph.D., Assistant Professor
Department of CSE (IOT)
https://orcid.org/0000-0002-2253-0382
E-mail: ankur22887@gmail.com
Noida Institute of Engineering and Technology, Greater Noida, India
https://www.niet.co.in

Swati Vishnoi, Assistant Professor
Department of CSE
https://orcid.org/0000-0001-6846-108X
E-mail: swativishnoi1@gmail.com
Noida Institute of Engineering and Technology, Greater Noida, India
https://www.niet.co.in

A Spatio-temporal Residual Attention Network with SNR-adaptive Weighting for MIMO-OFDM Channel Estimation

Ousama El-azizi[1], Rachid Fateh[1,2], and Safi Said[1]

[1] *Sultan Moulay Slimane University, Beni Mellal, Morocco,*
[2] *Ibn Zohr University, Agadir, Morocco*

https://doi.org/10.26636/jtit.2026.3.2688

Submitted: 24 Jun. 2026, Accepted: 10 Aug. 2026

Abstract — This paper addresses channel estimation in MIMO-OFDM systems under time-varying Rician fading and spatially correlated propagation conditions, where reliable coherent detection requires accurate channel state information. Although least-squares (LS) estimation is simple and computationally efficient, it remains highly sensitive to noise. In addition, conventional deep learning-based estimators may suffer from tracking bias when temporal channel variations are not properly handled. To overcome these limitations, this paper proposes SMART-CE, a spatio-temporal residual attention network with SNR-adaptive weighting. The proposed approach refines a sequence of LS estimates by extracting spatial-frequency features, exploiting temporal channel correlation, and adaptively balancing historical and current channel information according to the operating SNR. The final channel estimate is obtained through a residual correction of the current LS estimate. Simulation results on an 8 × 8 MIMO-OFDM system show that SMART-CE generally achieves lower NMSE and BER than CNN, residual CNN, residual-attention CNN, recursive least-squares (RLS), and normalized least-mean-square (NLMS) baselines, with the most pronounced gains observed in the low-to-medium SNR regime.

Keywords — channel estimation, deep learning, MIMO-OFDM, NLMS, NMSE, residual attention, Rician fading, RLS, SNR-adaptive weighting

1. Introduction

Multiple-input multiple-output orthogonal frequency-division multiplexing (MIMO-OFDM) is a key technology for modern broadband wireless communication systems. MIMO improves spectral efficiency and link reliability by exploiting the spatial domain, whereas OFDM mitigates frequency-selective fading by dividing a wideband channel into multiple narrowband subcarriers. Owing to these advantages, MIMO-OFDM has become a suitable transmission scheme for high-data-rate and reliable wireless networks [1], [2].

The performance of MIMO-OFDM systems strongly depends on the availability of accurate channel state information (CSI) at the receiver, which is required for coherent detection, equalization, beamforming, and resource allocation. In real systems, CSI is commonly acquired through pilot-aided estimation. The least-squares (LS) estimator is widely used because of its simplicity and low computational cost.

However, LS is highly sensitive to noise and does not exploit channel statistics. More advanced estimators, such as minimum mean square error (MMSE), can improve estimation accuracy, but they require prior knowledge of the channel covariance matrix and noise variance, which may be unavailable or difficult to estimate in practical time-varying environments [3], [4].

In mobile scenarios, the channel varies over time due to Doppler effects, while successive channel realizations remain temporally correlated. This correlation can be exploited to enhance estimation accuracy and mitigate channel aging effects [5], [6]. Adaptive filtering techniques, such as normalized least-mean-square (NLMS) and recursive least-squares (RLS) [7]–[9], have been proposed to track time-varying channels by recursively updating the channel estimate [10].

Although these methods can follow slow channel variations, their performance is sensitive to the step size, pilot structure, and noise level. Moreover, excessive temporal smoothing may introduce tracking bias at high signal-to-noise ratio (SNR), where the current LS estimate is already relatively accurate. Consequently, a naive use of adaptive estimates may degrade rather than improve the final estimation performance.

Recent advances in deep learning (DL) have motivated a new research direction for wireless physical-layer design. DL-based channel estimation methods have shown promising performance by learning nonlinear mappings between noisy observations and the corresponding channel responses. In particular, convolutional neural networks (CNNs) are well suited for OFDM-based systems because the channel response across antennas and subcarriers can be represented as a structured two-dimensional feature map [1], [3], [11].

Residual learning further simplifies the estimation task by enabling the network to learn a correction over a conventional estimator instead of reconstructing the entire channel directly [4], [12]. In addition, attention mechanisms improve feature representation by emphasizing the most informative spatial-frequency components [12], [14], [15]. To capture temporal dynamics, recurrent architectures and temporal learning mod-

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

ules have also been combined with CNN-based estimators to model inter-slot channel correlation in time-varying scenarios [16]–[18].

Despite these advances, two important limitations remain. First, many DL-based estimators rely on a single-snapshot LS estimate and therefore do not fully exploit the temporal correlation available across successive slots. Second, although hybrid model-driven and data-driven schemes have been investigated [2], [13], [19], a fixed or additive fusion strategy may be suboptimal when the reliability of temporal information changes with the operating SNR and channel dynamics.

In particular, an adaptive estimate that is useful at low SNR may introduce motion-induced bias at high SNR. This observation motivates the need for an SNR-aware temporal aggregation mechanism that can adaptively balance noise reduction and bias avoidance [20].

In this paper, we propose SMART-CE, a spatio-temporal residual attention network with SNR-adaptive temporal fusion for MIMO-OFDM channel estimation. Unlike single-snapshot estimators, SMART-CE processes a temporal sequence of LS estimates using a shared residual attention encoder to exploit inter-slot channel correlation. A lightweight SNR-aware hyper-network dynamically modulates the temporal attention weights, allowing the network to average past slots at low SNR for noise reduction while emphasizing the current slot at high SNR to mitigate motion-induced bias.

A convolutional block attention module (CBAM) is then applied to the fused spatio-temporal features to enhance informative spatial-frequency components. In parallel, an exponential moving average (EMA)-based auxiliary branch provides causally smoothed contextual information through feature-level fusion. Finally, the channel estimate is obtained as a residual correction over the current-slot LS estimate, which stabilizes learning and preserves a direct connection to the conventional baseline.

The SNR-conditioned temporal weighting mechanism is the main methodological contribution of SMART-CE. It adjusts the use of historical observations according to the operating SNR, reducing noise at low SNR while limiting tracking bias as the current observation becomes more reliable. The residual encoder, CBAM refinement, EMA context branch, and residual output connection are established components used to implement and support this temporal fusion. They are not presented as separate methodological contributions.

The main contributions of this paper are summarized as follows:

- we formulate the tradeoff between temporal noise reduction and motion-induced tracking bias, explaining why a fixed temporal averaging rule may not be suitable across different SNR regimes,
- an SNR-conditioned temporal weighting mechanism is introduced, that controls the contributions of the current and previous LS estimates according to the operating SNR,
- the SNR-adaptive temporal fusion with a shared residual encoder is integrated, CBAM-based feature refinement, and an EMA-derived auxiliary context branch,
- we compare SMART-CE with the classical NLMS and RLS adaptive estimators and with CNN, Res-CNN, and RA-CNN learning-based baselines. All methods are evaluated using the same channel realizations, pilot observations, noise conditions, and current-slot channel targets.

The remainder of this paper is organized as follows. Section 2 presents the MIMO-OFDM system model and formulates the channel estimation problem. Section 3 details the proposed SMART-CE framework, including the shared temporal residual attention encoder, the SNR-aware hyper-network, the CBAM module, the EMA auxiliary branch, and the residual output stage. Section 4 reports and discusses the numerical simulation results. Finally, Section 5 concludes the paper.

2. System Model and Problem Formulation

2.1. MIMO-OFDM System Model

Consider a MIMO-OFDM system in which a transmitter equipped with N_t antennas communicates with a receiver equipped with N_r antennas over a time-varying wireless channel. The number of active subcarriers is denoted by N_{sc}. The channel is assumed to remain approximately constant within one OFDM symbol but may vary from one time slot to another due to mobility and Doppler effects. Perfect time and frequency synchronization is assumed.

Let $\mathbf{s}_a^{\kappa} \in \mathbb{C}^{N_{\mathrm{sc}}\times 1}$ denote the vector of frequency-domain symbols transmitted from the a-th transmit antenna during the κ-th OFDM symbol, where $a = 1, \ldots, N_t$. The corresponding time-domain signal before cyclic prefix insertion is obtained as:

$$\mathbf{x}_a^{\kappa} = \mathbf{F}_{N_{\mathrm{sc}}}^{H}\, \mathbf{s}_a^{\kappa}\,, \tag{1}$$

where $\mathbf{F}_{N_{\mathrm{sc}}}$ is the normalized N_{sc}-point DFT matrix and $(\cdot)^H$ denotes the Hermitian transpose.

A cyclic prefix longer than the maximum delay spread is appended to preserve the circular convolution property and avoid inter-symbol interference.

After cyclic prefix removal and DFT operation at the receiver, the received signal at the n-th subcarrier and the κ-th time slot is written as:

$$\mathbf{y}^{\kappa}[n] = \mathbf{H}^{\kappa}[n]\mathbf{s}^{\kappa}[n] + \mathbf{w}^{\kappa}[n]\,, \tag{2}$$

where $\mathbf{y}^{\kappa}[n] \in \mathbb{C}^{N_r\times 1}$ is the received signal vector, $\mathbf{s}^{\kappa}[n] \in \mathbb{C}^{N_t\times 1}$ is the transmitted signal vector, and $\mathbf{H}^{\kappa}[n] \in \mathbb{C}^{N_r\times N_t}$ is the MIMO channel frequency response at subcarrier n. The vector $\mathbf{w}^{\kappa}[n] \in \mathbb{C}^{N_r\times 1}$ denotes additive complex Gaussian noise with independent entries following $\mathcal{CN}(0, \sigma_w^2)$. The channel frequency response matrix is expressed as:

$$\mathbf{H}^{\kappa}[n] = \begin{bmatrix} H_{1,1}^{\kappa}[n] & \cdots & H_{1,N_t}^{\kappa}[n] \\ \vdots & \ddots & \vdots \\ H_{N_r,1}^{\kappa}[n] & \cdots & H_{N_r,N_t}^{\kappa}[n] \end{bmatrix}, \tag{3}$$

where $H_{r,a}^{\kappa}[n]$ denotes the channel coefficient between the a-th transmit antenna and the r-th receive antenna.

For a frequency-selective channel with L taps, this coefficient can be written as:

$$H_{r,a}^{\kappa}[n] = \sum_{\ell=0}^{L-1} h_{r,a}^{\kappa}[\ell]\, e^{-j\frac{2\pi n \ell}{N_{\mathrm{sc}}}}, \tag{4}$$

where $h_{r,a}^{\kappa}[\ell]$ is the ℓ-th channel impulse response tap.

2.2. Time-varying Rician Channel Model

To capture mobility-induced temporal variations, one of the considered propagation scenarios is modeled as a time-varying Rician channel [21], [22]. The channel matrix at slot κ is decomposed into a line-of-sight (LoS) component and a non-line-of-sight (NLoS) component as:

$$\mathbf{H}^{\kappa}[n] = \sqrt{\frac{K}{K+1}}\,\mathbf{H}_{\mathrm{LoS}}[n] + \sqrt{\frac{1}{K+1}}\,\mathbf{H}_{\mathrm{NLoS}}^{\kappa}[n], \tag{5}$$

where K denotes the Rician factor.

The NLoS component evolves according to a first-order autoregressive process inspired by the Jakes Doppler model [21], [23]:

$$\mathbf{H}_{\mathrm{NLoS}}^{\kappa}[n] = \alpha\,\mathbf{H}_{\mathrm{NLoS}}^{\kappa-1}[n] + \sqrt{1-\alpha^2}\,\mathbf{U}^{\kappa}[n], \tag{6}$$

where $\mathbf{U}^{\kappa}[n]$ is a complex Gaussian innovation matrix with i.i.d. $\mathcal{CN}(0,1)$ entries.

The temporal correlation coefficient is given by:

$$\alpha = J_0(2\pi f_d T_s), \tag{7}$$

where $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind, f_d is the maximum Doppler frequency, and T_s is the slot duration.

The Doppler frequency is related to the mobile velocity v and the carrier frequency f_c by $f_d = v f_c / c$, where c is the speed of light. A larger α corresponds to stronger temporal correlation between consecutive slots, which can be exploited to improve channel estimation.

2.3. Spatially Correlated MIMO Channel Model

The second considered propagation scenario accounts for spatial correlation between antenna elements. Following the Kronecker correlation model, the channel matrix is expressed as:

$$\mathbf{H}^{\kappa}[n] = \mathbf{R}_r^{1/2}\,\mathbf{H}_{\mathrm{iid}}^{\kappa}[n]\,\mathbf{R}_t^{1/2}, \tag{8}$$

where $\mathbf{H}_{\mathrm{iid}}^{\kappa}[n]$ is an independent complex Gaussian channel matrix, while $\mathbf{R}_t$ and $\mathbf{R}_r$ denote the transmit and receive spatial correlation matrices, respectively.

In the exponential correlation model, their entries are given by:

$$[\mathbf{R}_t]_{i,j} = \rho_t^{|i-j|}, \qquad [\mathbf{R}_r]_{i,j} = \rho_r^{|i-j|}, \tag{9}$$

where ρ_t and ρ_r control the correlation levels at the transmitter and receiver sides.

This model is useful for evaluating the ability of the estimator to exploit structured spatial dependencies across antenna arrays.

2.4. Pilot-aided LS Estimation

During pilot transmission, known pilot symbols are transmitted to estimate the channel response. Let T_p denote the number of pilot symbols. The received pilot matrix at subcarrier n and slot κ is written as:

$$\mathbf{Y}_p^{\kappa}[n] = \mathbf{H}^{\kappa}[n]\,\mathbf{S}_p[n] + \mathbf{W}_p^{\kappa}[n], \tag{10}$$

where $\mathbf{Y}_p^{\kappa}[n] \in \mathbb{C}^{N_r \times T_p}$ is the received pilot matrix, $\mathbf{S}_p[n] \in \mathbb{C}^{N_t \times T_p}$ is the known pilot matrix, and $\mathbf{W}_p^{\kappa}[n]$ is the corresponding noise matrix.

The pilot matrix is constructed from orthogonal DFT sequences satisfying:

$$\mathbf{S}_p[n]\,\mathbf{S}_p^H[n] = \mathbf{I}_{N_t}, \qquad T_p \geqslant N_t, \tag{11}$$

which limits noise enhancement in the LS estimate.

The LS estimate of the channel at slot κ and subcarrier n is obtained as:

$$\hat{\mathbf{H}}_{\mathrm{LS}}^{\kappa}[n] = \mathbf{Y}_p^{\kappa}[n]\,\mathbf{S}_p^H[n]\left(\mathbf{S}_p[n]\,\mathbf{S}_p^H[n]\right)^{-1}. \tag{12}$$

Although LS has low computational complexity and does not require channel statistics, its accuracy is strongly affected by noise, especially at low SNR.

2.5. Temporal Input Sequence

Unlike single-snapshot estimators, the proposed framework exploits a temporal observation window of T consecutive slots. The sequence of LS estimates is collected as:

$$\mathcal{S}_{\mathrm{LS}} = \left\{\hat{\mathbf{H}}_{\mathrm{LS}}^{(t)}[n]\right\}_{t=1}^{T}, \tag{13}$$

where t denotes the relative slot index within the observation window and $\hat{\mathbf{H}}_{\mathrm{LS}}^{(T)}[n]$ corresponds to the current slot to be estimated.

This sequence provides the spatio-temporal input used by SMART-CE to exploit both the current pilot observation and the correlation across previous slots.

For the auxiliary temporal branch, a causal EMA of the LS sequence is computed as:

$$\hat{\mathbf{H}}_{\mathrm{EMA}}^{(t)}[n] = \beta\,\hat{\mathbf{H}}_{\mathrm{EMA}}^{(t-1)}[n] + (1-\beta)\,\hat{\mathbf{H}}_{\mathrm{LS}}^{(t)}[n],\ t = 2,\ldots,T, \tag{14}$$

with initialization $\hat{\mathbf{H}}_{\mathrm{EMA}}^{(1)}[n] = \hat{\mathbf{H}}_{\mathrm{LS}}^{(1)}[n]$, where $\beta \in (0,1)$ is the smoothing factor.

The final smoothed estimate is denoted by:

$$\hat{\mathbf{H}}_{\mathrm{adapt}}[n] = \hat{\mathbf{H}}_{\mathrm{EMA}}^{(T)}[n]. \tag{15}$$

The EMA branch provides a causal temporal context that can reduce noise when successive channels are strongly correlated. However, because the channel may evolve across the window, excessive smoothing can introduce tracking bias. This tradeoff motivates the SNR-adaptive temporal fusion mechanism introduced in the next section.

2.6. Real-valued Tensor Representation

For the learning-based estimator, the complex-valued channel matrices are converted into real-valued tensors by stacking

their real and imaginary parts. For each time slot, the LS estimate is represented as:

$$\hat{\mathcal{H}}_{\mathrm{LS}}^{(t)} \in \mathbb{R}^{2\times N_{\mathrm{sc}}\times N_r N_t}, \tag{16}$$

where the first dimension contains the real and imaginary components, the second dimension corresponds to the OFDM subcarriers, and the last dimension contains the vectorized MIMO channel coefficients.

The temporal sequence of LS estimates used by SMART-CE is then written as:

$$\mathcal{H}_{\mathrm{seq}} = \left\{\hat{\mathcal{H}}_{\mathrm{LS}}^{(t)}\right\}_{t=1}^{T} \in \mathbb{R}^{T\times 2\times N_{\mathrm{sc}}\times N_r N_t}. \tag{17}$$

The current-slot LS estimate corresponds to $\hat{\mathcal{H}}_{\mathrm{LS}}^{(T)}$. Similarly, the EMA auxiliary estimate is represented as:

$$\hat{\mathcal{H}}_{\mathrm{adapt}} \in \mathbb{R}^{2\times N_{\mathrm{sc}}\times N_r N_t}. \tag{18}$$

The true channel tensor used as the training target is denoted by $\mathcal{H} \in \mathbb{R}^{2\times N_{\mathrm{sc}}\times N_r N_t}$ and corresponds to the channel at the current slot.

2.7. Problem Formulation

The objective of channel estimation is to recover the current channel response from noisy pilot-based observations. In the proposed framework, the final estimate is formed as a residual correction over the current-slot LS estimate:

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_{\mathrm{LS}}^{(T)} + f_\theta\left(\mathcal{H}_{\mathrm{seq}}, \hat{\mathcal{H}}_{\mathrm{adapt}}, \gamma\right), \tag{19}$$

where $f_\theta(\cdot)$ denotes the SMART-CE estimator with trainable parameters θ and γ is the SNR value provided to the SNR-aware module.

The residual formulation allows the network to learn the estimation error of the current LS input rather than reconstructing the full channel from scratch. This design provides a direct connection to the conventional LS baseline, stabilizes training, and helps preserve the fine structure of the current channel observation. The detailed architecture of $f_\theta(\cdot)$ is presented in the next section.

Notation: Matrices, vectors, and scalars are denoted by boldface uppercase, boldface lowercase, and normal letters, respectively. Superscripts $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^{-1}$ denote transpose, Hermitian transpose, and matrix inverse, respectively. The operator $\mathbb{E}\{\cdot\}$ denotes expectation, $J_0(\cdot)$ is the zeroth-order Bessel function of the first kind, and $\|\cdot\|_F$ represents the Frobenius norm.

3. Proposed Channel Estimation Technique

This section presents SMART-CE, the proposed spatio-temporal residual attention network with SNR-adaptive temporal fusion. Unlike single-snapshot estimators that rely only on the current LS estimate, SMART-CE exploits three complementary sources of information: temporal correlation from a sequence of LS estimates, spatial-frequency structure through residual attention and CBAM, and instantaneous SNR information through a lightweight hyper-network that adaptively controls temporal aggregation. The overall architecture is shown in Fig. 1.

The proposed estimator takes three inputs: the temporal LS sequence $\mathcal{H}_{\mathrm{seq}} \in \mathbb{R}^{T\times 2\times N_{\mathrm{sc}}\times N_r N_t}$, the EMA-based auxiliary estimate $\hat{\mathcal{H}}_{\mathrm{adapt}} \in \mathbb{R}^{2\times N_{\mathrm{sc}}\times N_r N_t}$, and the instantaneous SNR value γ. It outputs a residual correction $\Delta\mathcal{H}$, which is added to the current-slot LS estimate $\hat{\mathcal{H}}_{\mathrm{LS}}^{(T)}$ to obtain the final channel estimate.

3.1. Shared Residual Attention Encoder

The first component of SMART-CE is a residual attention encoder that is applied independently to each slot of the LS sequence using shared weights. Weight sharing reduces the number of trainable parameters and enforces a consistent feature extraction process across time [16].

For each slot $t = 1, \ldots, T$, the corresponding LS tensor $\hat{\mathcal{H}}_{\mathrm{LS}}^{(t)} \in \mathbb{R}^{2\times N_{\mathrm{sc}}\times N_r N_t}$ is first projected into a C-dimensional feature space as:

$$\mathbf{Z}_0^{(t)} = \phi\left(\mathcal{B}\left(\mathbf{W}_0 * \hat{\mathcal{H}}_{\mathrm{LS}}^{(t)}\right)\right), \tag{20}$$

where $*$ denotes convolution, $\mathbf{W}_0$ is the input convolution kernel, $\mathcal{B}(\cdot)$ denotes batch normalization, and $\phi(\cdot)$ is the ReLU activation function.

The input projection uses a 3×3 convolution from 2 to 64 channels, followed by batch normalization and ReLU. The shared encoder contains six residual attention blocks. Each block uses two 3×3 convolutions with 64 input and output channels, batch normalization, ReLU after the first convolution, and a squeeze-and-excitation channel gate with a reduction ratio of 8.

The feature map is then refined by N_{RB} residual attention blocks. For the i-th block, the transformation can be written as:

$$\tilde{\mathbf{Z}}_i^{(t)} = \mathcal{B}\left(\mathbf{W}_{i,2} * \phi\left(\mathcal{B}\left(\mathbf{W}_{i,1} * \mathbf{Z}_{i-1}^{(t)}\right)\right)\right), \tag{21}$$

followed by channel-attention recalibration and residual addition:

$$\mathbf{Z}_i^{(t)} = \phi\left(\mathbf{Z}_{i-1}^{(t)} + \mathcal{A}_{\mathrm{ch}}\left(\tilde{\mathbf{Z}}_i^{(t)}\right)\right), \tag{22}$$

where $\mathcal{A}_{\mathrm{ch}}(\cdot)$ denotes the channel-attention operation.

For an input feature map $\mathbf{X} \in \mathbb{R}^{C\times N_{\mathrm{sc}}\times N_r N_t}$, it is given by:

$$\mathcal{A}_{\mathrm{ch}}(\mathbf{X}) = \mathbf{X} \odot \sigma\left(\mathbf{W}_{e2}\phi\left(\mathbf{W}_{e1}\mathrm{GAP}(\mathbf{X})\right)\right), \tag{23}$$

where $\mathrm{GAP}(\cdot)$ denotes global average pooling, $\mathbf{W}_{e1}$ and $\mathbf{W}_{e2}$ are learnable excitation weights, $\sigma(\cdot)$ is the sigmoid function, and $\odot$ denotes element-wise multiplication.

The residual connection improves gradient flow and enables the encoder to learn informative corrections over the input features [4], [12].

The per-slot feature map produced by the shared encoder is denoted by:

$$\mathbf{F}^{(t)} = \mathbf{Z}_{N_{\mathrm{RB}}}^{(t)} \in \mathbb{R}^{C\times N_{\mathrm{sc}}\times N_r N_t}. \tag{24}$$

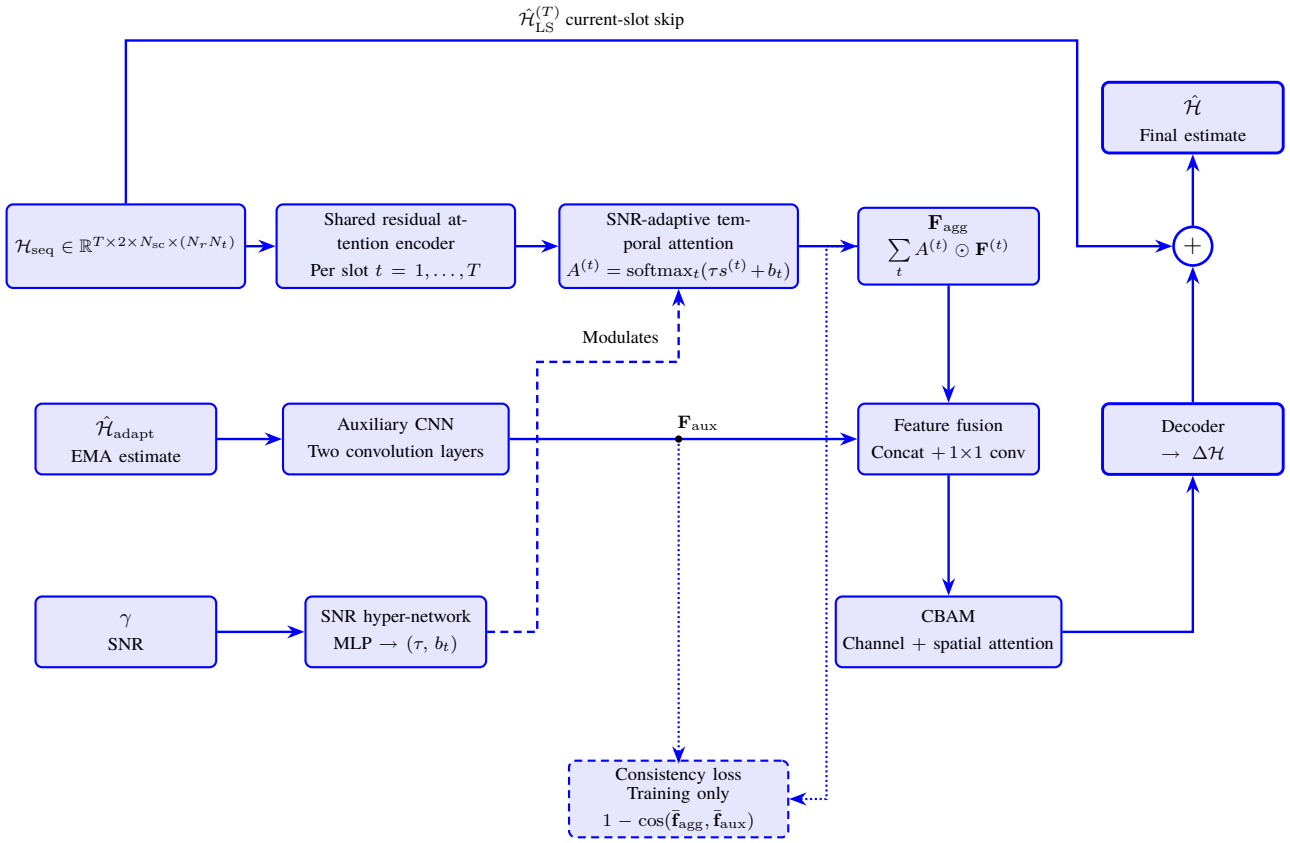


Fig. 1. Overall architecture of the proposed SMART-CE framework.

3.2. SNR-aware Temporal Attention

A key component of SMART-CE is the SNR-aware aggregation of the per-slot features. The reliability of past channel observations depends on both the noise level and the channel dynamics. At low SNR, exploiting several temporally correlated slots can reduce the impact of noise. At high SNR, however, the current LS estimate becomes more reliable and excessive reliance on past slots may introduce tracking bias due to residual channel variations.

To account for this tradeoff, a lightweight hyper-network $f_{\mathrm{hyp}}(\cdot)$ maps the normalized SNR to a temporal temperature τ and a set of per-slot bias terms $\{b_t\}_{t=1}^{T}$ [20]:

$$[\tau,\, b_1, \ldots, b_T] = f_{\mathrm{hyp}}\left(\gamma/\gamma_0\right), \quad \tau = \mathrm{softplus}(\cdot) + \tau_0\,, \tag{25}$$

where γ_0 is a normalization constant and $\tau_0 > 0$ ensures a strictly positive temperature.

For the five-slot input window, the hyper-network is an MLP with layer dimensions $1 \rightarrow 32 \rightarrow 32 \rightarrow 6$ and ReLU after the first two linear layers. Its six outputs are one temperature parameter and five slot-dependent bias terms. The SNR input is expressed in dB and normalized by $\gamma_0 = 20$ dB, and the positive-temperature offset is $\tau_0 = 0.5$.

For each time slot, an element-wise temporal score is computed from the corresponding feature map using a 1×1 convolution:

$$\mathbf{s}^{(t)} = \mathbf{W}_s * \mathbf{F}^{(t)} \in \mathbb{R}^{1\times N_{\mathrm{sc}}\times N_r N_t}. \tag{26}$$

The temporal attention weights are then obtained by an SNR-modulated softmax over the temporal dimension:

$$\mathbf{A}^{(t)} = \frac{\exp\left(\tau\, \mathbf{s}^{(t)} + b_t\right)}{\sum_{t'=1}^{T} \exp\left(\tau\, \mathbf{s}^{(t')} + b_{t'}\right)}\,. \tag{27}$$

The aggregated spatio-temporal feature is given by:

$$\mathbf{F}_{\mathrm{agg}} = \sum_{t=1}^{T} \mathbf{A}^{(t)} \odot \mathbf{F}^{(t)}. \tag{28}$$

Through this mechanism, the network can learn to assign smoother temporal weights when noise dominates and to emphasize the current slot when the instantaneous LS observation becomes more reliable. This SNR-aware fusion allows SMART-CE to balance noise reduction and tracking-bias mitigation without using a fixed temporal averaging rule.

The SNR value provided to the hyper-network is assumed to be available from standard receiver-side noise variance estimation in pilot-aided systems.

3.3. EMA Auxiliary Branch and CBAM-based Feature Refinement

In parallel with the main temporal branch, the EMA-based auxiliary estimate $\hat{\mathcal{H}}_{\mathrm{adapt}}$ is processed by a lightweight convolutional branch $\psi(\cdot)$ to produce a complementary contextual

feature:

$$\mathbf{F}_{\text{aux}} = \psi\left(\hat{\mathcal{H}}_{\text{adapt}}\right) \in \mathbb{R}^{C \times N_{\text{sc}} \times N_r N_t} . \tag{29}$$

The aggregated temporal feature and the auxiliary feature are concatenated along the channel dimension and fused by a 1×1 convolution:

$$\mathbf{F}_{\text{fus}} = \phi\big(\mathcal{B}\left(\mathbf{W}_f * [\mathbf{F}_{\text{agg}},\, \mathbf{F}_{\text{aux}}]\right)\big) . \tag{30}$$

The auxiliary branch uses $\text{Conv}_{3\times3}(2, 64)$–BN–ReLU–$\text{Conv}_{3\times3}(64, 64)$–BN–ReLU. After concatenation, a 1×1 convolution followed by batch normalization and ReLU projects the resulting 128-channel tensor to 64 channels.

The auxiliary estimate is therefore injected at the feature level rather than being directly added to the channel estimate. This design allows the network to exploit smoothed temporal context while limiting the risk of propagating a biased auxiliary estimate to the output.

All convolutional layers use unit stride. Padding is one for the 3×3 convolutions, three for the 7×7 CBAM spatial-attention convolution, and zero for the 1×1 projection and output convolutions. These settings preserve the spatial-frequency dimensions through feature extraction, fusion, attention, and decoding.

The fused feature is further refined using CBAM [15], which applies channel-wise and spatial attention sequentially. The channel-attention stage recalibrates the fused feature as:

$$\mathbf{F}_c = \mathcal{A}_{\text{ch}}\left(\mathbf{F}_{\text{fus}}\right) , \tag{31}$$

where $\mathcal{A}_{\text{ch}}(\cdot)$ denotes the channel-attention operation defined in the previous subsection.

The spatial-attention stage then emphasizes the most informative subcarriers and spatial-channel components:

$$\mathbf{F}_{\text{cbam}} = \mathbf{F}_c \odot \sigma\big(\mathbf{W}_{sp} * [\text{AvgPool}_c(\mathbf{F}_c),\, \text{MaxPool}_c(\mathbf{F}_c)]\big) , \tag{32}$$

where $\text{AvgPool}_c(\cdot)$ and $\text{MaxPool}_c(\cdot)$ denote average and maximum pooling along the channel dimension, respectively, and $\mathbf{W}_{sp}$ is the spatial-attention convolution kernel.

The channel-attention stage uses a reduction ratio of 8. The spatial-attention stage applies a 7×7 convolution to the concatenated channel-wise average and maximum maps.

3.4. Residual Output Stage

The refined feature is decoded into a residual correction by two convolutional layers. The decoder uses a 3×3 convolution from 64 to 32 channels, followed by batch normalization, ReLU, and a 1×1 convolution from 32 to 2 output channels:

$$\Delta\mathcal{H} = \mathbf{W}_{d2} * \phi\big(\mathcal{B}\left(\mathbf{W}_{d1} * \mathbf{F}_{\text{cbam}}\right)\big) . \tag{33}$$

The final channel estimate is then obtained by adding this correction to the current-slot LS estimate:

$$\hat{\mathcal{H}} = \hat{\mathcal{H}}_{\text{LS}}^{(T)} + \Delta\mathcal{H} . \tag{34}$$

The last convolutional layer is initialized with zero weights and zero bias, so that $\Delta\mathcal{H} = \mathbf{0}$ at the beginning of training and the initial output coincides with the current LS estimate. This initialization provides a stable starting point for optimization and encourages the network to learn only the residual estimation error rather than reconstructing the full channel response from scratch. Moreover, the residual connection preserves a direct path from the current pilot-based observation to the output, which helps retain fine channel details, especially when the LS estimate becomes more reliable at high SNR.

The temporal branch, the auxiliary EMA branch, and the residual output stage therefore play complementary roles. The temporal branch exploits inter-slot correlation, the auxiliary branch provides smoothed contextual information, and the residual path anchors the final estimate to the current-slot LS observation. This design improves robustness without directly adding a potentially biased smoothed estimate to the output.

3.5. Training Objective

The network is trained in a supervised manner by minimizing a composite loss function. The primary term is the normalized mean square error between the estimated channel and the true current-slot channel:

$$\mathcal{L}_{\text{NMSE}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \frac{\left\|\mathcal{H}_i - \hat{\mathcal{H}}_i\right\|_F^2}{\|\mathcal{H}_i\|_F^2 + \varepsilon} , \tag{35}$$

where N_b is the batch size and ε is a small positive constant introduced for numerical stability.

To encourage consistency between the aggregated temporal representation and the auxiliary EMA-based representation, an additional feature-consistency term is used during training:

$$\mathcal{L}_{\text{cons}} = \frac{1}{N_b} \sum_{i=1}^{N_b} \left(1 - \frac{\left\langle \bar{\mathbf{f}}_{\text{agg},i},\, \bar{\mathbf{f}}_{\text{aux},i} \right\rangle}{\left\|\bar{\mathbf{f}}_{\text{agg},i}\right\| \left\|\bar{\mathbf{f}}_{\text{aux},i}\right\| + \varepsilon}\right) , \tag{36}$$

where $\bar{\mathbf{f}}_{\text{agg},i}$ and $\bar{\mathbf{f}}_{\text{aux},i}$ denote the globally averaged temporal and auxiliary feature vectors of the i-th sample, respectively, and $\langle \cdot, \cdot \rangle$ denotes the inner product.

The Adam optimizer [24] is used for training, together with a plateau-based learning-rate scheduler and validation-based checkpoint selection. The total training loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{NMSE}} + \lambda\, \mathcal{L}_{\text{cons}} , \tag{37}$$

where λ controls the strength of the consistency regularization. In the implemented configuration, this term is used only as an auxiliary training regularizer. It is not required during inference and does not introduce additional computational cost at test time.

The consistency term encourages alignment between the learned spatio-temporal representation and the EMA-derived smoothed context, while the NMSE term directly optimizes channel reconstruction accuracy. Its empirical contribution is scenario dependent, as investigated in Subsection 4.5.

The complete procedure is summarized in Algorithm 1.

During inference, SMART-CE requires only a single forward pass and does not involve any iterative optimization. Given the LS sequence, the EMA estimate, and the SNR value, the trained network directly produces the refined channel

Algorithm 1 SMART-CE channel estimation

Input: LS sequence $\mathcal{H}_{\text{seq}} = \{\hat{\mathcal{H}}_{\text{LS}}^{(t)}\}_{t=1}^{T}$, EMA estimate $\hat{\mathcal{H}}_{\text{adapt}}$, SNR value γ, and trained parameters θ
Output: Refined channel estimate $\hat{\mathcal{H}}$
Start

1: **for** $t = 1$ to T **do**
2: Feature extraction: $\mathbf{F}^{(t)} = f_\theta\left(\hat{\mathcal{H}}_{\text{LS}}^{(t)}\right)$
3: Compute temporal score: $\mathbf{s}^{(t)} = \mathbf{W}_s * \mathbf{F}^{(t)}$
4: **end for**
5: Compute SNR-dependent temporal parameters:

$$[\tau, b_1, \ldots, b_T] = f_{\text{hyp}}(\gamma/\gamma_0)$$

6: Compute temporal attention weights:

$$A^{(t)} = \text{softmax}_t\big(\tau s^{(t)} + b_t\big)$$

7: Aggregate temporal features:

$$F_{\text{agg}} = \sum_{t=1}^{T} A^{(t)} \odot F^{(t)}$$

8: Extract auxiliary EMA-based feature:

$$F_{\text{aux}} = \psi(\hat{H}_{\text{adapt}})$$

9: Fuse temporal and auxiliary features:

$$F_{\text{fus}} = \phi\big(B(W_f * [F_{\text{agg}}, F_{\text{aux}}])\big)$$

10: Refine by CBAM: $F_{\text{cbam}} = \text{CBAM}(F_{\text{fus}})$
11: Decode residual correction:

$$\Delta H = W_{d2} * \phi\big(B(W_{d1} * F_{\text{cbam}})\big)$$

12: Produce final estimate:

$$\hat{H} = \hat{H}_{\text{LS}}^{(T)} + \Delta H$$

13: **return** $\hat{H}$
End

estimate. This non-iterative inference procedure makes the proposed framework suitable for practical MIMO-OFDM channel estimation scenarios where improved estimation accuracy must be achieved under constrained processing latency.

4. Simulation Results

In this section, we evaluate the channel estimation performance of the proposed SMART-CE framework through numerical simulations. We first describe the simulation setup, the dataset generation procedure, the compared estimation schemes, and the adopted performance metrics. Then, the results are discussed by comparing SMART-CE with LS, adaptive filters, and learning-based baselines under two representative MIMO-OFDM propagation scenarios: time-varying Rician fading and spatially correlated MIMO channels.

4.1. Training and Testing of SMART-CE

The datasets are generated according to the MIMO-OFDM system model described in Sec. 2. For each channel realization, a temporal trajectory of T consecutive slots is generated in order to provide both the current pilot-based observation and the previous temporally correlated observations. The mobility level is varied by drawing the user velocity uniformly within the predefined range, which allows the dataset to cover low- and moderate-mobility conditions. For each slot, the received pilot observations are generated under a randomly selected SNR value and the corresponding LS estimate is computed using orthogonal DFT pilots.

The resulting temporal sequence of LS estimates $\mathcal{H}_{\text{seq}}$ and the EMA-based auxiliary estimate $\hat{\mathcal{H}}_{\text{adapt}}$ are used as inputs to SMART-CE, while the true channel of the current slot $\mathcal{H}$ is used as the supervised learning target. The reported evaluation focuses on two representative MIMO-OFDM propagation scenarios: time-varying Rician fading and spatially correlated MIMO channels.

The training configuration uses an initial learning rate of 10^{-3}, weight decay 10^{-4}, a ReduceLROnPlateau scheduler with reduction factor 0.5, gradient clipping at unit norm, and early stopping with patience 12.

Training is limited to 50 epochs with batch size 128. For each training trajectory, one SNR value is drawn independently from the continuous uniform distribution $\mathcal{U}[-10, 20]$ dB and retained across its five slots. At test time, the SNR is fixed at the value associated with each evaluated point. All learning-based models are trained under the same data generation settings and then evaluated over the considered SNR range on the retained time-varying Rician and spatially correlated MIMO scenarios.

The proposed SMART-CE model is compared with LS, complex-valued NLMS and RLS adaptive estimators, as well as CNN, Res-CNN, and RA-CNN learning-based baselines. NLMS and RLS are initialized using the first-slot LS estimate and process the pilot observations from slots $t = 2, \ldots, T$. The NLMS and RLS implementations, including their parameter configurations, follow the standard complex-valued adaptive-filtering update rules and settings reported in [25]. All estimators are evaluated using the same channel trajectories, pilot observations, noise realizations, SNR values, and current-slot channel targets. CNN, Res-CNN, and RA-CNN use only the current-slot LS estimate, whereas NLMS, RLS, and SMART-CE exploit observations from the five-slot temporal window.

The main simulation parameters are summarized in Tab. 1.

4.2. Evaluation Metrics

The performance is assessed using two complementary metrics. The first metric is NMSE, which directly quantifies the channel estimation accuracy and is defined as:

$$\text{NMSE} = \mathbb{E}\left[\frac{\left\|\mathcal{H} - \hat{\mathcal{H}}\right\|_F^2}{\|\mathcal{H}\|_F^2}\right], \tag{38}$$

where $\mathcal{H}$ and $\hat{\mathcal{H}}$ denote the true and estimated channel tensors, respectively. The tables and robustness figures report NMSE in dB as $10\log_{10}(\text{NMSE})$.

Tab. 1. Simulation parameters.

Parameter	Value
Transmit antennas	$N_t = 8$
Receive antennas	$N_r = 8$
OFDM subcarriers	$N_{\mathrm{sc}} = 64$
Pilot type	Orthogonal DFT pilots
Channel models	Time-varying Rician, spatially correlated MIMO
Rician K-factor	5 dB
Mobile velocity	0 – 60 km/h
Carrier frequency	$f_c = 3.5$ GHz
Inter-slot time interval	$T_s = 1$ ms
Spatial correlation	$\rho_t = \rho_r = 0.7$
Temporal window length	$T = 5$ slots
Training samples	40 000
Validation samples	4000
Testing samples per SNR	3000
Evaluation seeds	$\{11, 22, 33, 44, 55\}$
SNR-robustness seeds	$\{11, 22, 33\}$
SNR-input error range	$[-3, 3]$ dB
Training epochs	50
Batch size	128
Optimizer	Adam
Learning rate	10^{-3}
Training SNR	$\gamma_{\mathrm{train}} \sim \mathcal{U}[-10, 20]$ dB
SNR normalization	$\gamma_0 = 20$ dB
Temperature offset	$\tau_0 = 0.5$
Convolution stride	1
Convolution padding	1 for 3×3, 3 for 7×7, and 0 for 1×1
Base channels	$C = 64$
Residual attention blocks	$N_{\mathrm{RB}} = 6$
EMA smoothing factor	$\beta = 0.6$
Consistency weight	$\lambda = 0.1$
Loss function	NMSE + consistency loss
Reported metrics	NMSE, BER
Hardware accelerator	NVIDIA T4 GPU

The second metric is BER, which evaluates the end-to-end impact of channel estimation errors on data detection. QPSK symbols are transmitted through the true channel, while the receiver performs detection using an MMSE equalizer constructed from the estimated channel. For each subcarrier, the equalized symbol vector is obtained as

$$\hat{\mathbf{x}} = \left(\hat{\mathbf{H}}^H \hat{\mathbf{H}} + \frac{1}{\rho} \mathbf{I}_{N_t} \right)^{-1} \hat{\mathbf{H}}^H \mathbf{y} \,, \tag{39}$$

where $\hat{\mathbf{H}}$ is the estimated channel matrix, $\mathbf{y}$ is the received signal vector generated through the true channel, ρ is the linear SNR, and $\mathbf{I}_{N_t}$ is the identity matrix of size N_t.

The BER is then computed by comparing the detected QPSK bits with the transmitted bits and averaging the error rate over all test realizations at each SNR value.

4.3. Multi-seed Protocol

To quantify variability across test realizations without changing the trained estimators, the selected checkpoint of each learning-based method is evaluated on five independently generated held-out test sets with seeds $\{11, 22, 33, 44, 55\}$. For each seed, all estimators receive the same channel trajectories, pilot observations, and noise realizations. The resulting variability therefore reflects changes in the test data rather than independent network retraining.

For a metric value z_m obtained with evaluation seed m, the reported mean and two-sided 95% Student-t confidence interval are computed as

$$\bar{z} = \frac{1}{M} \sum_{m=1}^{M} z_m \,, \qquad \bar{z} \pm t_{0.975,M-1} \frac{s_z}{\sqrt{M}} \,, \tag{40}$$

where $M = 5$, s_z is the sample standard deviation, and $t_{0.975,4} = 2.776$.

The original NMSE and BER curves are retained, while the five-seed results are reported in tables as the mean and confidence interval.

4.4. Performance Evaluation of the Proposed Model

The performance of SMART-CE is evaluated in terms of NMSE and BER over the considered SNR range. The reported results focus on two representative MIMO-OFDM propagation scenarios: time-varying Rician fading and spatially correlated MIMO channels. The proposed method is compared with LS, RLS, NLMS, CNN, ResCNN, and RA-CNN baselines.

Figures 2 and 3 report the point-estimate NMSE and BER curves, respectively, for the time-varying Rician channel over the complete SNR range. SMART-CE achieves the lowest NMSE and BER values across the evaluated SNR range, with the most pronounced gains over the learning-based baselines in the low-to-medium SNR regime. These curves represent the reference point-estimate evaluation and are considered separately from the five-seed statistics reported below.

Figures 4 and 5 compare the point-estimate NMSE and BER results for the spatially correlated MIMO channel. Each channel realization remains fixed over the five-slot observation window and is regenerated independently for every test sample, allowing NLMS and RLS to combine repeated observations without motion-induced tracking bias. SMART-CE generally attains lower NMSE than the CNN-based estimators and adaptive filters at low and medium SNR. As the SNR increases, the NMSE curves become closer, and the adaptive filters achieve slightly lower values at the highest evaluated SNR.

The BER results follow a different trend: SMART-CE maintains the lowest BER across the complete SNR range, with a wider separation at medium and high SNR. Under the evaluated spatially correlated conditions, NLMS and RLS are there-

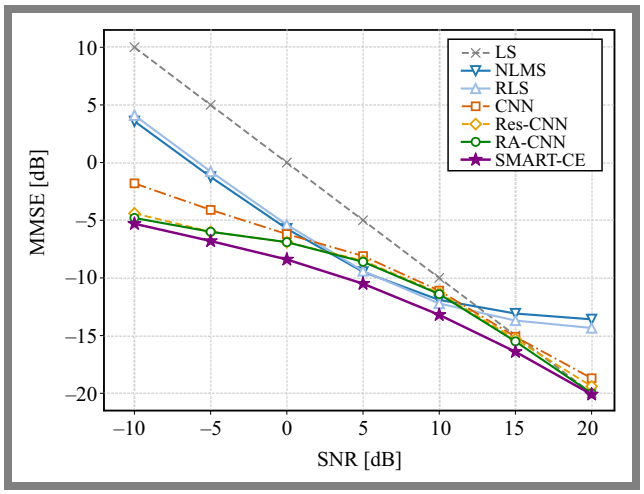


Fig. 2. NMSE performance over the time-varying Rician channel.

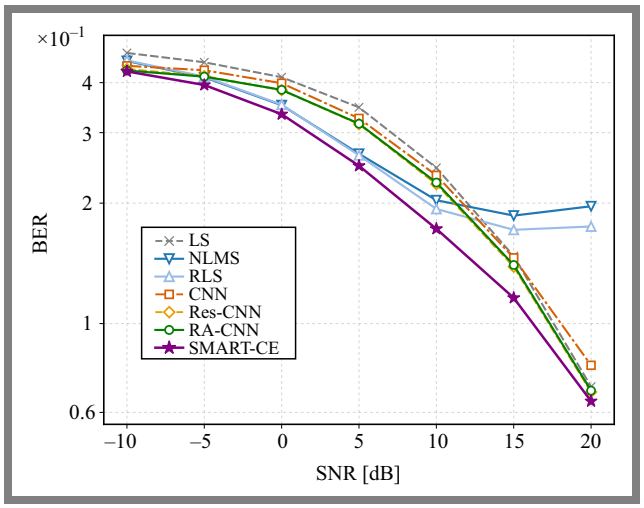


Fig. 3. BER performance over the time-varying Rician channel.

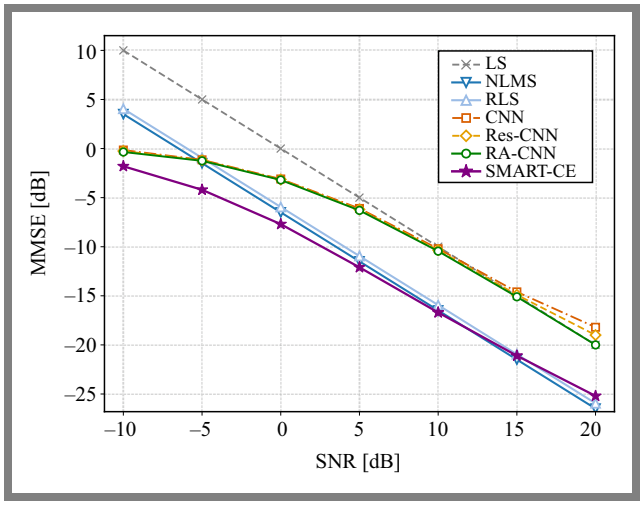


Fig. 4. NMSE performance over the spatially correlated MIMO channel.

fore effective for high-SNR channel reconstruction, whereas the SMART-CE estimates remain more favorable for MMSE-based data detection.

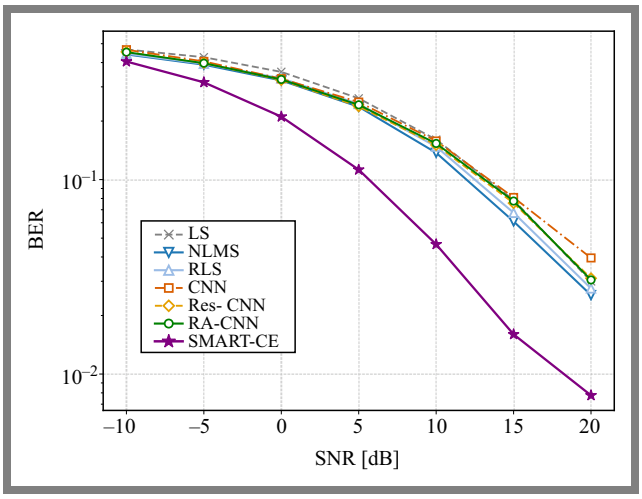


Fig. 5. BER performance over the spatially correlated MIMO channel.

Table 2 reports the mean NMSE and the corresponding 95% Student-t confidence interval for five independently generated held-out test sets over $\mathrm{SNR} \in \{-10, -5, 0, 5, 10\}$ dB. For the time-varying Rician channel, SMART-CE gives the lowest mean NMSE at all five SNR values. At 0 dB, its mean NMSE is approximately 1.40 dB below that of RA-CNN, the closest learning-based baseline at this operating point. For the spatially correlated channel, SMART-CE gives the lowest mean NMSE at four of the five SNR values and remains close to the leading adaptive estimator at 10 dB.

Table 3 reports the corresponding mean BER and 95% Student-t confidence interval. SMART-CE gives the lowest mean BER at all five SNR values in the time-varying Rician channel. In the spatially correlated channel, it gives the lowest mean BER at four of the five SNR values and remains close to the leading adaptive estimator at 10 dB. The narrow intervals indicate limited variability across the five test sets; they do not imply uniform superiority under all propagation conditions.

4.5. Component Ablation Analysis

To isolate the contribution of each component, four ablated variants were retrained using the same data-generation procedure, optimization settings, and validation-based checkpoint selection as the complete SMART-CE model. The variants remove the SNR-conditioned hyper-network, CBAM refinement, EMA auxiliary branch, and feature-consistency loss, respectively. All variants use the same test seed (42), 3000 samples per SNR, and the SNR set $\{-10, -5, 0, 5, 10\}$ dB in both propagation scenarios.

Figure 6 reports the resulting NMSE and BER curves. As shown in Fig. 6a and Fig. 6c, removing the SNR hyper-network substantially increases both NMSE and BER over the time-varying Rician channel. The same trend is observed for the spatially correlated MIMO channel in Fig. 6b and Fig. 6d. This comparison is particularly important because the SNR-conditioned temporal weighting mechanism constitutes the main methodological contribution of SMART-CE.

Tab. 2. Mean NMSE and 95% Student-t confidence-interval half-width over five fixed-checkpoint evaluation seeds.

Time-varying Rician					
Method	**–10**	**–5**	**0**	**5**	**10**
LS	10.00 ±0.002	5.00 ±0.003	0.00 ±0.002	–5.00 ±0.002	–10.00 ±0.001
NLMS	3.59 ±0.002	–1.24 ±0.004	–5.74 ±0.001	–9.42 ±0.038	–11.84 ±0.045
RLS	4.10 ±0.003	–0.77 ±0.004	–5.39 ±0.001	–9.35 ±0.032	–12.15 ±0.041
CNN	–1.81 ±0.004	–4.05 ±0.006	–6.20 ±0.005	–8.11 ±0.002	–11.06 ±0.001
Res CNN	–4.38 ±0.009	–5.96 ±0.003	–6.91 ±0.001	–8.52 ±0.001	–11.39 ±0.001
RA-CNN	–4.84 ±0.005	–5.98 ±0.002	–6.93 ±0.002	–8.55 ±0.001	–11.44 ±0.001
SMART-CE	**–5.36 ±0.002**	**–6.79 ±0.002**	**–8.33 ±0.001**	**–10.51 ±0.034**	**–13.23 ±0.023**
Spatially correlated MIMO					
Method	**–10**	**–5**	**0**	**5**	**10**
LS	10.01 ±0.006	5.01 ±0.005	0.01 ±0.003	–4.99 ±0.002	–9.99 ±0.004
NLMS	3.51 ±0.006	–1.49 ±0.004	–6.49 ±0.003	–11.49 ±0.003	–16.49 ±0.004
RLS	4.05 ±0.005	–0.95 ±0.004	–5.95 ±0.003	–10.95 ±0.003	–15.95 ±0.004
CNN	–0.18 ±0.002	–1.44 ±0.003	–3.25 ±0.002	–5.72 ±0.003	–10.16 ±0.005
Res CNN	–0.80 ±0.002	–2.07 ±0.002	–4.06 ±0.002	–6.13 ±0.005	–10.23 ±0.005
RA-CNN	–0.63 ±0.001	–1.67 ±0.002	–3.58 ±0.002	–6.39 ±0.004	–10.14 ±0.006
SMART-CE	**–2.25 ±0.003**	**–4.59 ±0.004**	**–7.92 ±0.004**	**–12.12 ±0.003**	**–16.34 ±0.011**

Tab. 3. Mean BER and 95% Student-t confidence-interval half-width over five fixed-checkpoint evaluation seeds.

Time-varying Rician					
Method	**–10**	**–5**	**0**	**5**	**10**
LS	0.4745 ±0.0004	0.4499 ±0.0020	0.4099 ±0.0011	0.3441 ±0.0021	0.2519 ±0.0029
NLMS	0.4514 ±0.0011	0.4128 ±0.0016	0.3473 ±0.0021	0.2629 ±0.0040	0.2067 ±0.0051
RLS	0.4528 ±0.0004	0.4146 ±0.0015	0.3501 ±0.0024	0.2624 ±0.0027	0.1982 ±0.0047
CNN	0.4409 ±0.0005	0.4288 ±0.0021	0.3947 ±0.0019	0.3230 ±0.0020	0.2403 ±0.0031
Res CNN	0.4317 ±0.0006	0.4157 ±0.0006	0.3802 ±0.0017	0.3123 ±0.0022	0.2293 ±0.0026
RA-CNN	0.4282 ±0.0006	0.4138 ±0.0010	0.3795 ±0.0018	0.3130 ±0.0024	0.2313 ±0.0029
SMART-CE	**0.4264 ±0.0007**	**0.3938 ±0.0014**	**0.3303 ±0.0023**	**0.2455 ±0.0032**	**0.1757 ±0.0027**
Spatially correlated MIMO					
Method	**–10**	**–5**	**0**	**5**	**10**
LS	0.4729 ±0.0026	0.4446 ±0.0016	0.4018 ±0.0016	0.3478 ±0.0018	0.2750 ±0.0036
NLMS	0.4415 ±0.0008	0.3903 ±0.0015	0.3205 ±0.0021	0.2318 ±0.0017	0.1382 ±0.0018
RLS	0.4444 ±0.0007	0.3954 ±0.0013	0.3280 ±0.0016	0.2412 ±0.0006	0.1472 ±0.0021
CNN	0.4658 ±0.0022	0.4206 ±0.0016	0.3849 ±0.0018	0.3436 ±0.0020	0.2741 ±0.0039
Res CNN	0.4507 ±0.0034	0.4134 ±0.0016	0.3781 ±0.0018	0.3400 ±0.0016	0.2729 ±0.0032
RA-CNN	0.4527 ±0.0028	0.4152 ±0.0015	0.3808 ±0.0018	0.3378 ±0.0015	0.2737 ±0.0036
SMART-CE	**0.4207 ±0.0011**	**0.3720 ±0.0017**	**0.3090 ±0.0020**	**0.2249 ±0.0020**	**0.1400 ±0.0022**

For the time-varying Rician channel, the NMSE gap between the complete model and the variant without the SNR hyper-network decreases from 8.238 dB at –10 dB to 1.547 dB at 10 dB, with an average degradation of 4.170 dB across the five evaluated SNR values. For the spatially correlated channel, the NMSE gap ranges from 2.163 to 5.626 dB, with an average degradation of 3.372 dB. The corresponding average BER increases are 0.05359 and 0.04675, respectively.

The larger degradation at low SNR is consistent with the intended role of the hyper-network: historical observations are most useful when the current LS estimate is noise limited, whereas their influence should be reduced as the current observation becomes more reliable. These results support the proposed SNR-conditioned balance between temporal noise reduction and motion-induced tracking bias. As further illustrated in Fig. 6a–d, the EMA branch also contributes

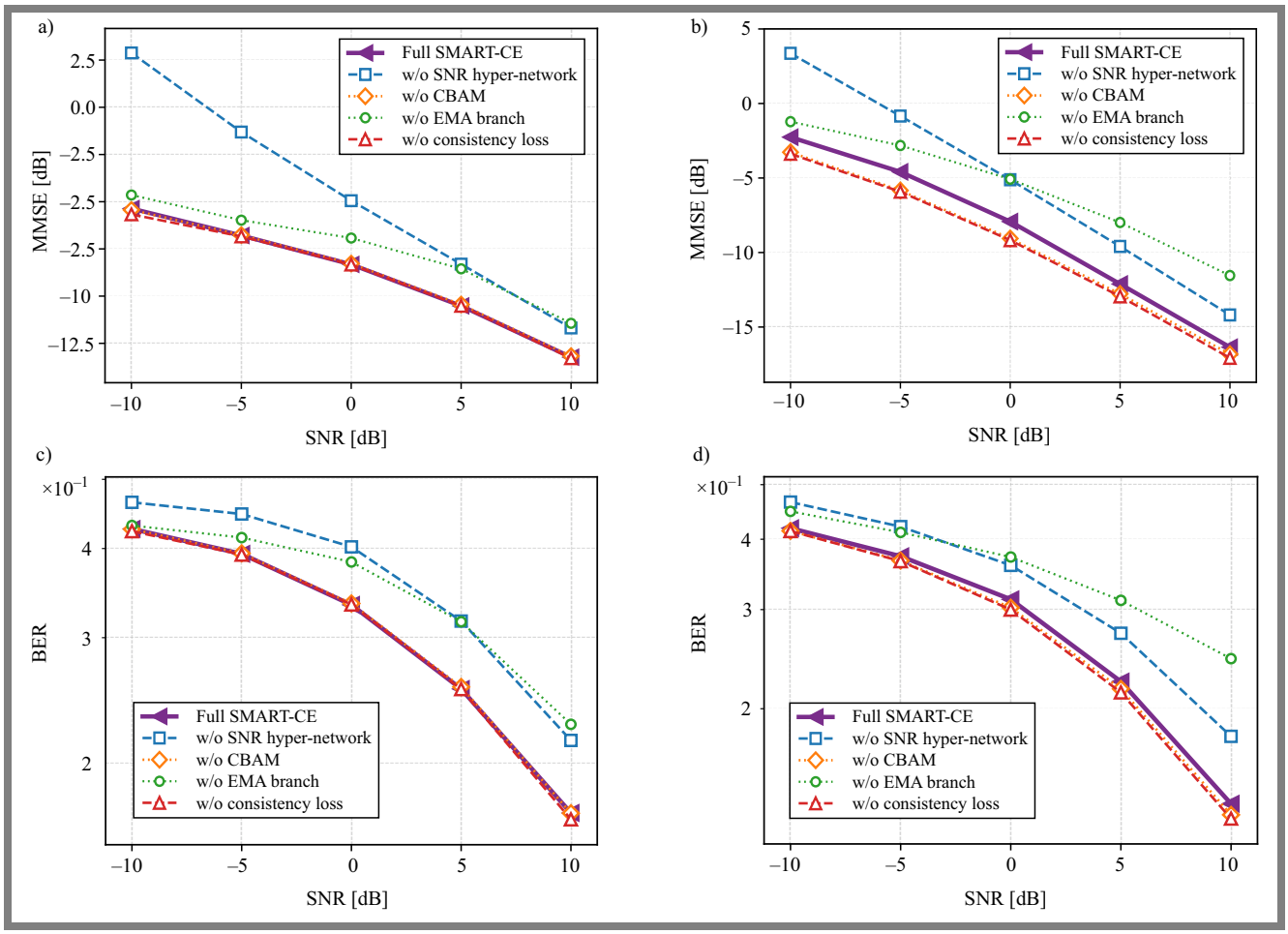


Fig. 6. Component-wise ablation of SMART-CE over the representative SNR range.

consistently, since its removal increases NMSE and BER at all five SNR values in both propagation scenarios. The corresponding average NMSE penalties are 1.336 dB for the Rician channel and 2.918 dB for the spatially correlated channel.

By contrast, removing CBAM or the consistency loss produces only small changes under Rician fading and lower errors under some spatially correlated conditions.

Therefore, the observed performance gain cannot be attributed equally to all architectural components. It is primarily associated with the SNR-conditioned temporal weighting mechanism, while the EMA branch provides complementary temporal context. CBAM and consistency regularization remain supporting components whose effects depend on the evaluated propagation scenario.

4.6. Robustness to Imperfect SNR Information

In practical receivers, the SNR supplied to the hyper-network may differ from the true operating SNR. This mismatch is modeled as $\hat{\gamma} = \gamma + \Delta\gamma$, where γ denotes the true operating SNR and $\Delta\gamma$ is the receiver-side SNR-estimation error. Robustness is evaluated for $\Delta\gamma \in \{-3, -2, -1, 0, 1, 2, 3\}$ dB at true SNR values $\{0, 5, 10, 15\}$ dB.

The trained SMART-CE checkpoint is kept fixed, and each condition is evaluated on three independently generated test sets using seeds $\{11, 22, 33\}$ and 3000 samples per seed. For each channel model, true SNR, and test seed, the same test data are used for all seven perturbations. Therefore, only the SNR value supplied to the hyper-network is modified, allowing the effect of the SNR-input error to be isolated.

Mean NMSE and two-sided 95% Student-t confidence intervals are computed using Eq. (40), with $M = 3$ and $t_{0.975,2} = 4.303$. These intervals quantify variability across independently generated test sets for a fixed SMART-CE checkpoint rather than variability across independent training runs.

Figures 7 and 8 show the NMSE sensitivity of SMART-CE to SNR-estimation errors under time-varying Rician fading and spatially correlated MIMO channels, respectively. The reference point $\Delta\gamma = 0$ corresponds to perfect SNR information. Across the evaluated true-SNR values, the maximum increase in mean NMSE relative to the perfect-SNR case remains below 0.05 dB. These results indicate that SMART-CE is robust to moderate SNR-input errors within the tested range.

The validation loss convergence curves of the learning-based estimators are shown in Fig. 9. From the first epoch, SMART-CE achieves a markedly lower validation loss, around 0.178,

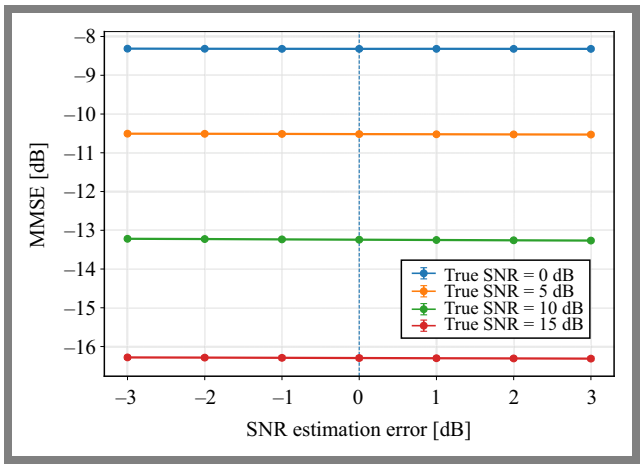


Fig. 7. SMART-CE robustness to SNR-estimation errors under time-varying Rician fading.

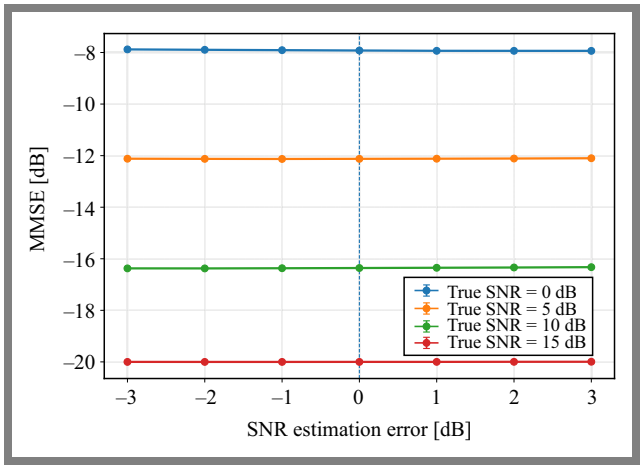


Fig. 8. SMART-CE robustness to SNR-estimation errors under spatially correlated MIMO channels.

compared with CNN, Res-CNN, and RA-CNN, which start at approximately 0.298, 0.240, and 0.260, respectively. This early advantage highlights the effectiveness of the residual initialization strategy, which guides the network output toward the current LS estimate from the initial training stage.

Although all models reach a stable plateau before epoch 35, SMART-CE attains the lowest final validation loss, approximately 0.110. This corresponds to a reduction of about 40% compared with RA-CNN, around 0.183, and 53% compared with CNN, around 0.235. It is also worth noting that Res-CNN and RA-CNN converge to almost the same final loss, indicating that spatial attention alone, in the absence of temporal information, provides only a marginal improvement over residual learning. Overall, these results demonstrate that the combined use of temporal correlation, SNR-adaptive fusion, and CBAM-based feature refinement enables SMART-CE to achieve faster, more stable, and more accurate convergence.

Table 4 compares the parameter count, forward-pass computational complexity, and batch-one inference latency of the learning-based estimators. SMART-CE has the highest computational cost among the evaluated models because it processes a sequence of five LS estimates and incorporates residual-attention blocks, SNR-adaptive temporal fusion,

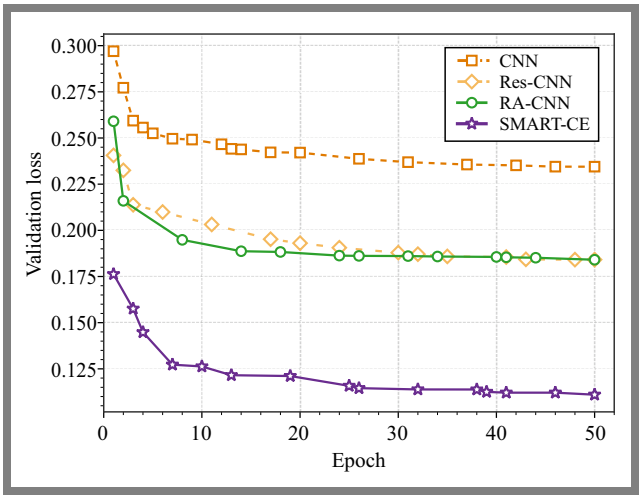


Fig. 9. Validation loss convergence of the learning-based estimators.

Tab. 4. Computational complexity and batch-one inference latency of the learning-based channel estimators on an NVIDIA T4 GPU.

Model	Parameters	GFLOPs	Runtime [ms/sample]
CNN	296 962	2.426	1.077 ±0.049
Res-CNN	463 746	3.785	2.455 ±0.619
RA-CNN	469 890	3.785	3.550 ±0.504
SMART-CE	518 987	18.700	16.130 ±4.097

CBAM refinement, and an auxiliary EMA branch. The inference runtime is reported as the mean and standard deviation over 100 batch-one forward passes performed after 30 warm-up iterations on an NVIDIA T4 GPU. FLOPs account for convolutional and linear-layer operations, with multiplications and additions counted separately, whereas element-wise operations and softmax computations are excluded.

5. Conclusions

This paper proposed SMART-CE, a spatio-temporal residual attention framework for MIMO-OFDM channel estimation. The proposed network combines a shared residual attention encoder, an SNR-adaptive temporal fusion mechanism, CBAM-based feature refinement, and an EMA-based auxiliary branch to exploit both temporal correlation and spatial-frequency channel structure.

The final estimate is obtained through a residual correction of the current LS estimate, which stabilizes learning and preserves a direct connection to the pilot-based observation. The ablation results identify the SNR-conditioned temporal weighting mechanism as the main source of the consistent performance gain, with the EMA branch providing complementary temporal context.

SMART-CE achieved lower NMSE and BER than the evaluated baselines over most of the tested conditions. Although SMART-CE requires a higher computational cost than simpler CNN-based baselines, it performs channel estimation through a single forward pass without iterative optimization.

References

[1] I. Almeida, J. Guerreiro, and R. Dinis, "On Deep Learning Hybrid Architectures for MIMO-OFDM Channel Estimation", *Electronics*, vol. 14, art. no. 4692, 2025 (https://doi.org/10.3390/electronics14234692).

[2] Q. Wu *et al.*, "A Convolutional-transformer Residual Network for Channel Estimation in Intelligent Reflective Surface Aided MIMO Systems", *Sensors*, vol. 25, art. no. 5959, 2025 (https://doi.org/10.3390/s25195959).

[3] B. Zheng, C. You, W. Mei, and R. Zhang, "A Survey on Channel Estimation and Practical Passive Beamforming Design for Intelligent Reflecting Surface Aided Wireless Communications", *IEEE Communications Surveys & Tutorials*, vol. 24, pp. 1035–1071, 2022 (https://doi.org/10.1109/COMST.2022.3155305).

[4] A. Melgar *et al.*, "Deep Neural Network: An Alternative to Traditional Channel Estimators in Massive MIMO Systems", *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, pp. 657–671, 2022 (https://doi.org/10.1109/TCCN.2022.3164888).

[5] H. Hirose, S. Yang, T. Ohtsuki, and M. Bouazizi, "Deep Learning-based Channel Estimation to Mitigate Channel Aging in Massive MIMO with Pilot Contamination", *IEEE Access*, vol. 13, pp. 1834–1845, 2025 (https://doi.org/10.1109/ACCESS.2024.3518763).

[6] J. Yuan, H.Q. Ngo, and M. Matthaiou, "Machine Learning-based Channel Prediction in Massive MIMO with Channel Aging", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 2960–2974, 2020 (https://doi.org/10.1109/TWC.2020.2969627).

[7] R. Fateh, A. Darif, and S. Safi, "Performance Evaluation of MC-CDMA Systems with Single User Detection Technique Using Kernel and Linear Adaptive Method", *Journal of Telecommunications and Information Technology*, vol. 86, pp. 1–11, 2021 (https://doi.org/10.26636/jtit.2021.151621).

[8] R. Fateh *et al.* "Machine Learning Based System Identification with Binary Output Data Using Kernel Methods", *Journal of Telecommunications and Information Technology*, vol. 95, pp. 17–25, 2024 (https://doi.org/10.26636/jtit.2024.1.1430).

[9] R. Fateh *et al.*, "A Novel Kernel Algorithm for Finite Impulse Response Channel Identification", *Journal of Telecommunications and Information Technology*, vol. 92, pp. 84–93, 2023 (https://doi.org/10.26636/jtit.2023.169823).

[10] W. Shi *et al.*, "Adaptive Channel Estimation Based on Multidirectional Structure in Delay-Doppler Domain for Underwater Acoustic OTFS System", *Remote Sensing*, vol. 16, art. no. 3157, 2024 (https://doi.org/10.3390/rs16173157).

[11] B. Marinberg, A. Cohen, E. Ben-Dror, and H.H. Permuter, "A Study on MIMO Channel Estimation by 2D and 3D Convolutional Neural Networks", *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, New Delhi, India, 2020 (https://doi.org/10.1109/ANTS50601.2020.9342797).

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016 (https://doi.org/10.1109/CVPR.2016.90).

[13] X. Ma, Z. Gao, F. Gao, and M. Di Renzo, "Model-driven Deep Learning Based Channel Estimation and Feedback for Millimeter-wave Massive Hybrid MIMO Systems", *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 2388–2406, 2021 (https://doi.org/10.1109/JSAC.2021.3087269).

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation Networks", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, USA, 2018 (https://doi.org/10.1109/CVPR.2018.00745).

[15] S. Woo, J. Park, J.-Y. Lee, and I.S. Kweon, "CBAM: Convolutional Block Attention Module", *Proc. of European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018 (https://doi.org/10.1007/978-3-030-01234-2_1).

[16] X. Tian and Q. Zheng, "A Massive MIMO Channel Estimation Method Based on Hybrid Deep Learning Model with Regularization Techniques", *International Journal of Intelligent Systems*, vol. 2025, art. no. 2597866, 2025 (https://doi.org/10.1155/int/2597866).

[17] C. Eom and C. Lee, "Hybrid Neural Network-based Fading Channel Prediction for Link Adaptation", *IEEE Access*, vol. 9, pp. 117257–117266, 2021 (https://doi.org/10.1109/ACCESS.2021.3106739).

[18] S. Moon, H. Kim, and I. Hwang, "Deep Learning-based Channel Estimation and Tracking for Millimeter-wave Vehicular Communications", *Journal of Communications and Networks*, vol. 22, pp. 177–184, 2020 (https://doi.org/10.1109/JCN.2020.000012).

[19] T. Yassine and L. Le Magoarou, "mpNet: Variable Depth Unfolded Neural Network for Massive MIMO Channel Estimation", *IEEE Transactions on Wireless Communications*, vol. 21, pp. 5703–5714, 2022 (https://doi.org/10.1109/TWC.2022.3142737).

[20] R. Li, J. Sun, J. Xue, and C. Masouros, "Scenario-aware Learning Approaches to Adaptive Channel Estimation", *IEEE Transactions on Communications*, vol. 72, pp. 874–889, 2024 (https://doi.org/10.1109/TCOMM.2023.3330878).

[21] W.C. Jakes, *Microwave Mobile Communications*, New York, USA: Wiley, 656 p., 1974 (ISBN: 9780780310698).

[22] C.B. Peel and A.L. Swindlehurst, "Effective SNR for Space-time Modulation Over a Time-varying Rician Channel", *IEEE Transactions on Communications*, vol. 52, pp. 17–23, 2004 (https://doi.org/10.1109/TCOMM.2003.822148).

[23] R.H. Clarke, "A Statistical Theory of Mobile-radio Reception", *The Bell System Technical Journal*, vol. 47, pp. 957–1000, 1968 (https://doi.org/10.1002/j.1538-7305.1968.tb00069.x).

[24] D.P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization", *ArXiv*, 2015 (https://doi.org/10.48550/arXiv.1412.6980).

[25] R. Fateh, A. Darif, and S. Safi, "Kernel and Linear Adaptive Methods for the BRAN Channels Identification", *Advanced Intelligent Systems for Sustainable Development (AI2SD'2020)*, vol. 1418, 2022 (https://doi.org/10.1007/978-3-030-90639-9_47).

Ousama El-azizi, Ph.D.
LIMATI Laboratory
https://orcid.org/0009-0006-4299-0083
E-mail: elazizi.oussama@usms.ac.ma
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

Rachid Fateh, Professor
LIMATI Laboratory, IMIS Research Laboratory
https://orcid.org/0000-0002-0574-2105
E-mail: fateh.smi@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma
Ibn Zohr University, Agadir, Morocco
https://www.uiz.ac.ma

Safi Said, Professor
LIMATI Laboratory
https://orcid.org/0000-0003-3390-9037
E-mail: safi.said@gmail.com
Sultan Moulay Slimane University, Beni Mellal, Morocco
https://www.usms.ac.ma

FEC Hybrid Method Based on Reed–Solomon Codes for Transmitting Video Signal Over Limited Bandwidth and High Loss Rate Channels

Leonid V. Kaliuzhnyi, Viktor B. Rusalovskiy, Artem D. Ishchenko, and Vasyl M. Didenko

State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine

https://doi.org/10.26636/jtit.2026.3.2637

Submitted: 28 May 2026, Accepted: 24 Jul. 2026

Abstract — The paper considers the adaptive forward error correction (FEC) technique for real-time video transmission over a feedback-free channel with limited bandwidth and non-stationary interference. A hybrid Reed–Solomon coding method is proposed in GF(2^8), combining differentiated redundancy across specific frame types (I, P, and B), with adaptive code parameter control driven by physical layer channel state estimation. The channel is modeled by a three-state Markov process called Gilbert–Elliott–Jamming (GEJ), in which an explicit jamming state is added to the classical Gilbert–Elliott model. A closed-form expression for the block decoding failure probability under the erasure model is derived, and applicability bounds for representative RS configurations are determined. A redundancy adaptation function with a guaranteed upper bound on overhead is introduced together with a control algorithm. Experiments conducted under four scenarios (stable Wi-Fi, mobile 4G, tactical channel and active jamming) show that the mission success rate increases from 60% for fixed RS and 50% for SRT to 78% under sustained jamming, while the time to first frame decreases from 600 – 1200 ms to 200 ms. The method is implementable on the ESP32-S3 and ARM Cortex-M4 platforms with an encoding throughput of 4 to 8 Mbps.

Keywords — adaptive forward error correction, jamming-resistant video, Reed–Solomon codes, video streaming

1. Introduction

Video transmissions account for over 80% of global Internet traffic and continue to grow faster than the underlying infrastructure is capable of absorbing them [1], [2]. This growth is driven not only by entertainment streaming but also, increasingly, by interactive and mission-critical services such as remote surgery and telemedicine, disaster response and emergency communications, remote education, industrial teleoperation, as well as video links of unmanned aerial and robotic platforms. In such cases, an interruption of the picture is not merely a comfort issue, but a safety- and outcome-critical failure. Within this broader class, the most demanding applications are those in which the decoded picture is part of a control loop. These include the following: military and dual use unmanned aerial vehicles, tactical radio links, telemedicine, and critical infrastructure monitoring.

The efficiency of modern video delivery is based on predictive compression. Contemporary codecs such as H.264/AVC [3], H.265/HEVC [4] and the royalty-free AV1 solution [5] organize the stream into groups of pictures (GOPs) built from three frame types: intracoded I-frames, which are self-contained, as well as intercoded P- and B-frames, which are predicted from their neighbors. This creates a strong asymmetry in importance: the loss of an I frame invalidates the related entire GOP, whereas the loss of a single B-frame degrades only the frame itself. In this paper, a protection scheme that ignores this asymmetry and either wastes redundancy on the least important frames or leaves the most important ones exposed is developed.

In this paper, the impact of residual losses is ultimately judged in terms of perceived quality. Full-reference metrics such as peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [6] quantify the fidelity against a reference, while the limitations of purely error-based measures are well documented [7]. Perceptual metrics, such as video multimethod assessment fusion (VMAF) [8], correlate more closely with subjective scores and have become a de-facto standard for streaming evaluation. These metrics provide an objective benchmark based on which the resilience of a transmission scheme is assessed.

Furthermore, combat experience gathered in 2022 – 2025 has shown that civilian real-time protocols, such as RTP, RTMP, HLS, and analog FPV channels operating in the 2.4 and 5.8 GHz bands, are not usable in zones of active electronic warfare (EW), because wideband jamming destroys both the video downlink and the UAV control uplink [9], [10]. Retransmission-based protocols, i.e., TCP and RTP/RTCP with ARQ, introduce end-to-end latency of 100 – 200 ms, which is unacceptable for real-time control and the loss of key frames (I frames) interferes with decoder synchronization.

Forward error correction (FEC) offers a good alternative. However, a fixed FEC scheme is not well matched to a non-

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

stationary channel: at low loss rates it wastes bandwidth, while at high loss rates it does not deliver sufficient resilience.

From this dependency, we formulate the research problem as follows. The purpose of the research is to design an adaptive FEC method, based on Reed–Solomon codes, that guarantees, in the feedback-free transmission mode, a specific target resilience level ($P_{fail} \leqslant \varepsilon$) while minimizing total redundancy ($O_{total} \rightarrow \min$) on a channel with non-stationary characteristics.

Several aspects of the problem still lack a solution in the literature. First, the classical Gilbert–Elliott model captures impulsive interference, but not long, periodic blackouts that characterize active jamming, and the analytical methodology for feedback-free channels with explicit jamming states has not been fully developed. Second, most published AFEC algorithms do not provide an analytical upper bound on total redundancy, which complicates deployment in bitrate-constrained channels. Third, the adaptive schemes usually apply a uniform FEC rate to all packets, ignoring the differing importance of I, P and B frames within the group-of-pictures (GOP) structure. Fourth, in one-way scenarios, for example, a UAV transmitting to a ground station with a jammed uplink, the algorithm has to rely only on a priori channel data and physical layer measurements.

The object of this research is the process of transmitting real-time video signals over a channel with limited bandwidth and strong interference, operating in feedback-free one-way mode. The purpose of the work is to develop an adaptive hybrid Reed–Solomon coding method that ensures $P_{fail} \leqslant 10^{-3}$ at packet loss rates of up to 20%, with a guaranteed upper bound on total redundancy equaling $O_{total} \leqslant 30\%$ and an end-to-end latency below 150 ms, with the solution remaining implementable on embedded microcontroller platforms of the ESP32-S3 and ARM Cortex-M4 class.

The research tasks set in support of this purpose are as follows:

- to derive a mathematical channel model that accounts for non-stationary jamming states,
- to derive a closed-form expression for the block decoding failure probability under the erasure model,
- to formalize the redundancy adaptation function and the differentiated frame-type FEC control algorithm,
- to compare the RS code against alternative FEC schemes,
- to implement and verify the method on an experimental software testbed.

2. Related Works

The theory of forward error correction for erasure channels is mature. Reed–Solomon codes [11], introduced as polynomial codes over finite fields, are maximum distance separable (MDS) and, therefore, optimal in their trade-off between redundancy and correction ability. Canonical treatments of their algebraic decoding are provided in [12] and [13]. For multimedia transport specifically, the low delay error protection design space – from block codes to rateless and streaming constructions – is surveyed in [14].

These foundations fix the achievable protection for a given overhead, but say nothing about how the overhead should be chosen when the channel is non-stationary, which is the main concern of the literature focusing on adaptive solutions.

As far as adaptive, channel state-driven FEC is concerned, the most recent work tunes the coding parameters to the estimated channel state. The ABRF scheme of [15] performs joint bitrate-FEC control and reports a 68 – 95% reduction in VMAF degradation compared to earlier adaptive FEC schemes. In [16], DeepRS, an LSTM network, predicts packet loss patterns and adjusts the RS parameters. Reinforcement learning controllers such as RL-AFEC [17] and R-FEC [18] select FEC parameters online from interaction with the network.

These methods report substantial quality gains, but they share two properties that matter here: they apply a single, uniform code rate to all packets of a frame, ignoring the GOP importance asymmetry, and they either rely on an accurate online loss estimate or on a learned predictor, without providing an analytical guarantee on the resulting redundancy.

In streaming, cross-layer, and neural approaches a parallel line of work targets low latency through streaming code – such as the Tambur method provided in [19] for videoconferencing – and the network adaptive error control presented in [20] and through cross-layer coordination between the application and transport layers, as in VOXEL [21]. A more radical direction replaces the codec itself with a neural solution.

The GRACE algorithm introduced in [22] reports a large reduction in unrecoverable frames and a gain in the mean opinion score at the cost of a learned encoder-decoder pair. These approaches improve resilience but move away from the lightweight, analytically tractable operation required on resource-constrained hardware.

On the channel-modeling side, the Gilbert–Elliott two-state model remains the standard abstraction of a bursty erasure channel, because it represents a single “bad” regime. It underestimates the long periodic blackouts produced by active jamming.

To the best of our knowledge, no existing methods address the combination of the three following elements jointly: an explicit jamming state inside the channel model, a closed-form analytical bound on FEC overhead, and an explicit use of GOP frame type differentiation in the adaptive control.

3. Proposed Method

3.1. Mathematical Channel Model

We modeled the channel as a three-state Markov GEJ (Gilbert–Elliott–Jamming) process with states $\{G, B, J\}$ and a transition probability matrix. In state G (good), the packet-loss probability is $p_G \ll 0.01$, in state B (bad) it lies in $p_B \in [0.05, 0.15]$ (impulsive interference), and in state J (jamming) it is $p_J \approx 1.0$ (a near-complete blackout).

Tab. 1. GEJ model parameters for typical operating scenarios.

Scenario	π_G	π_B	π_J	p_G	p_B	p_J	L_B (avg)	L_J (avg)
S1: Wi-Fi 5 GHz	0.97	0.03	0.00	0.001	0.05	—	50 pkts	—
S2: 4G/LTE	0.85	0.13	0.02	0.002	0.08	0.95	200 pkts	1000 pkts
S3: Tactical	0.60	0.30	0.10	0.005	0.15	0.99	500 pkts	5000 pkts
S4: Active EW	0.40	0.30	0.30	0.005	0.15	0.99	500 pkts	5000 pkts

The transition matrix is parametrized by P_{GB}, P_{BG}, P_{GJ}, P_{JG}, P_{BJ} and P_{JB}, forming an irreducible ergodic chain in most realistic regimes. The problem of sweep jamming, which breaks stationarity, is treated separately.

The stationary distribution $\pi = (\pi_G, \pi_B, \pi_J)$ is the left principal eigenvector of the transition matrix, normalized by $\pi_G + \pi_B + \pi_J = 1$. The typical scenario parameters [10] are collected in Tab. 1. The steady-state packet loss probability is:

$$p_{\text{avg}} = \pi_G \cdot p_G + \pi_B \cdot p_B + \pi_J \cdot p_J \ . \tag{1}$$

The burst length distribution within each state follows an exponential law:

$$P(L > l) = e^{\frac{-l}{L_{avg}}} \ .$$

The model was calibrated by Baum–Welch estimation on two real, publicly available packet loss traces that bracket the regimes of interest: a benign link without jamming (the `CRAWDAD DUE` packet delivery dataset [10], overall loss ≈ 33.6%) and a link recorded under a periodic RF jammer (the `CRAWDAD vanetjamming2014` dataset [23], overall loss ≈ 77.9%). Under the Kolmogorov–Smirnov criterion, the three-state GEJ and the two-state Gilbert–Elliott models fit the benign trace comparably, achieving a KS error of 0.02 vs. 0.03 – without jamming the third state is largely redundant. Under active periodic jamming, the explicit jamming state becomes decisive (KS error 0.25 vs. 0.88), because a single bad state cannot simultaneously reproduce the short impulsive bursts and the long deliberate blackout.

3.2. Block Decoding Failure Probability

Consider a systematic Reed-Solomon code RS(n, k) over the finite Galois field GF(2^8), $n = 255$. Its error-correcting capability is:

$$t = \left\lfloor \frac{(n-k)}{2} \right\rfloor . \tag{2}$$

As a maximum distance separable (MDS) code, RS attains the Singleton bound $d_{\min} = n-k+1$ and corrects up to t symbol errors with unknown positions or up to $2\,t$ erasures with known positions. In UDP transmission with packet sequence numbers, lost packets always have known positions, so the erasure regime applies, doubling the protection capability.

The probability of failing to decode a block of length n under independent Bernoulli losses with rate p is:

$$P_{\text{fail}}(p, n, t) = \sum_{i=2t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i} \ . \tag{3}$$

Tab. 2. Block decoding failure probability for representative RS configurations (erasure model).

p [%]	Cyclic $g(x)=x^{16}+x^{12}+x^3+1$	RS(255, 231, t=12)	RS(255, 191, t=32)
1	$9.2 \cdot 10^{-1}$	$3.0 \cdot 10^{-17}$	$6.4 \cdot 10^{-70}$
5	≈ 1	$1.1 \cdot 10^{-3}$	$7.8 \cdot 10^{-28}$
10	≈ 1	$5.7 \cdot 10^{-1}$	$1.2 \cdot 10^{-12}$
20	≈ 1	≈ 1	$1.9 \cdot 10^{-2}$
30	≈ 1	≈ 1	$9.5 \cdot 10^{-1}$

Numerical values of Eq. (3) for representative RS configurations are provided in Tab. 2. From the table, we can read off applicability bounds: RS(255, 231, 12) keeps $P_{\text{fail}} \leqslant 10^{-3}$ as long as $p \leqslant 5\%$, and RS(255, 191, 32) does so up to $p \leqslant 10\%$. For p greater than 20%, the code alone is no longer enough and one has to add interleaving or a frequency hop.

For burst losses of length L distributed exponentially, the block failure probability is approximated by:

$$P_{\text{loss}}^{\text{burst}} = e^{-\frac{2t}{L_{\text{avg}}}} \ . \tag{4}$$

An interleaving depth D distributes a burst of length L across D blocks, reducing the effective burst length by a factor of D:

$$P_{\text{loss}}^{\text{int}} = e^{-\frac{2tD}{L_{\text{avg}}}} \ . \tag{5}$$

Beyond the block-failure probability, the effective rate R_{eff} captures both the nominal code rate k/n and the full-block loss probability:

$$R_{\text{eff}}(p, n, t) = \frac{k}{n} \cdot \left(1 - P_{\text{fail}}(p, n, t)\right) \ . \tag{6}$$

The decoding complexity of Berlekamp–Massey + Forney over GF(2^8) is as follows:

$$C_{\text{dec}}(n, t) = O(n \cdot t + t^2) \ . \tag{7}$$

For RS(255, 223, 16) this amounts to roughly 12 000 elementary GF operations per block, which translates to 12 µs on an ARM Cortex-M4 clocked at 168 MHz or approximately 1 µs on an ESP32-S3 that features hardware GF acceleration. The effective channel throughput under a given code is:

$$C = \frac{k}{n} \cdot (1-p) \cdot B \ , \tag{8}$$

where B is the physical radio-link capacity [bit/s].

3.3. Adaptive Redundancy Function and Differentiated Protection

The proposed adaptation function combines packet loss probability and the average burst length into a single scalar coeffi-

cient such as:

$$\kappa(p, L) = \min\left\{1 + 2p + \frac{L}{200},\ 3\right\}. \tag{9}$$

The cap $\kappa_{\max} = 3$ prevents unbounded overhead growth in catastrophically degraded channels. The redundancy for frame type $F \in \{I, P, B\}$ is as follows:

$$r_F = \min\left\{w_F \cdot r_{\text{base}} \cdot \kappa(p, L),\ 0.6\right\}, \tag{10}$$

where $w_I = 1.0$, $w_P = 0.3$, $w_B = 0.1$ are weights and $r_{\text{base}} \in [0.1, 0.4]$ constitutes the baseline rate driven by target VMAF.

The total redundancy for a GOP of period G with size ratio $\rho = S_I/S_P$ and $\rho \in [4, 10]$ is:

$$O_{\text{total}} = \frac{\rho \cdot r_I + (G-1) \cdot r_P}{\rho + (G-1)}. \tag{11}$$

For a typical H.264 stream with GOP $= 30$ and $\rho = 6$, Eq. (11) gives $O_{\text{total}} = 23.4\%$, against 40% for uniform protection at the same I-frame rate. This is a relative reduction of 41.5%.

3.4. RS Code Selection

Table 3 summarizes the comparative properties of the FEC schemes against key criteria. We selected the RS code as the baseline FEC algorithm, because it satisfies three requirements at once. It is MDS, i.e. it offers the optimal correction capability for a given redundancy, it tolerates burst losses up to $2\,t$ erasures, and its decoding cost remains moderate – about 12 000 GF operations per block at $t = 16$, which is well within the throughput capacity of an ESP32-S3 at 4 Mbps encoding.

The alternatives we considered fail to satisfy at least one of those criteria: RaptorQ does not provide a strict overhead bound, LDPC needs a more powerful platform than a Cortex-M4, and plain XOR-FEC offers very limited correction capability.

3.5. Parameter Sweep

To identify the best RS configuration for a given channel state, we ran a sweep over the (t, p) parametric plane. Five configurations with $n = 255$ and $k \in \{239, 223, 207, 191, 175\}$ so that $t \in \{8, 16, 24, 32, 40\}$ were tested at $p \in \{1, 5, 10, 15, 20, 25\}\%$. The test vector was the publicly available `mobile_cif.y4m` sequence (Xiph media, CIF 352×288), and the full RS encoding and decoding were performed on GF(2^8) using `reedsolo 1.7`.

This amounts to $5 \times 6 \times 30 = 900$ independent encode–decode runs with distinct seeds of random loss, requiring in total 65 min on an Intel CPU. At each of the 30 grid points, we computed P_{fail} from Eq. (3) and averaged the measured VMAF-proxy over 30 frames. UDP-baseline on the same 30 frames: $\text{VMAF}_{\text{udp}}(p) = \{71.8; 47.7; 37.9; 33.1; 28.3; 23.4\}$ for $p = \{1; 5; 10; 15; 20; 25\}$ [%]. The results are reported in the Tab. 4.

Each configuration has a clear applicability range, as one may notice in Tab. 4 and Fig. 1. The configuration $t = 8$ keeps VMAF $\geqslant 95$ up to $p = 5\%$ but then collapses at $p \geqslant 10\%$,

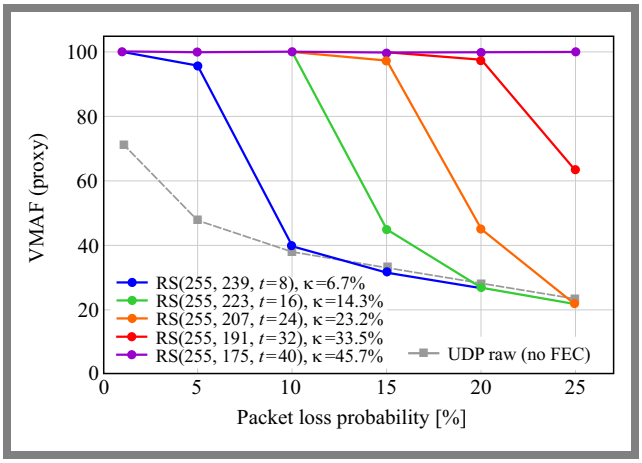


Fig. 1. Frame recovery quality for the family of RS configurations ($n = 255$, k varies; `mobile_cif` test vector).

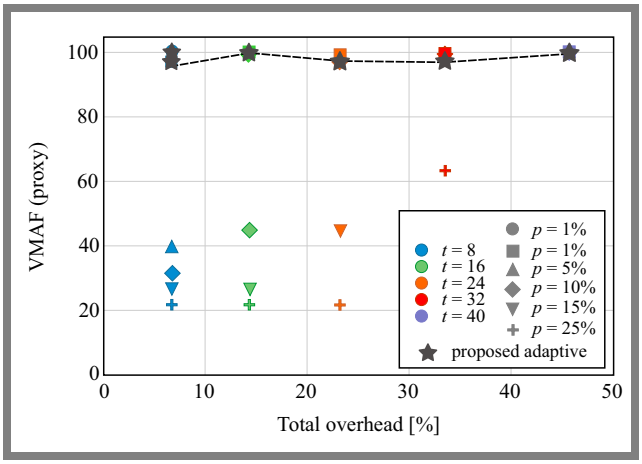


Fig. 2. Pareto chart overhead vs. quality for RS configurations.

$t = 16$ holds up to $p = 10\%$, $t = 24$ up to 15%, $t = 32$ up to 20%, and $t = 40$ covers the entire range up to 25%, at the price of 45.7% overhead. Therefore, a fixed configuration sized for the worst case ($t = 40$) wastes $6.8\times$ the necessary bandwidth in a stable channel at $p = 1\%$ (45.7% vs. 6.7%).

This is the main argument in favor of adaptive switching. The Pareto frontier in the overhead – VMAF plane (Fig. 2) gives the decision rule $t^*(p) = \arg\min\{\text{overhead} \mid \text{VMAF} \geqslant 95\}$: RS(255, 239, 8) at $p \in \{1, 5\}\%$; RS(255, 223, 16) at $p = 10\%$; RS(255, 207, 24) at $p = 15\%$; RS(255, 191, 32) at $p = 20\%$; RS(255, 175, 40) at $p = 25\%$.

3.6. Adaptive FEC Control Algorithm

Algorithm 1 estimates the channel state from physical layer metrics and selects RS code parameters before encoding each frame.

The complexity is $O(w + \log n)$, where w is the observation window size. For a typical $w = 30$, this is 30 comparison operations per call, which is negligible in addition to the cost of RS encoding itself.

Tab. 3. Comparison of FEC schemes for real-time video transmission over channels with non-stationary interference.

Scheme	Correction capability	Decoding complexity	Burst resilience	Overhead at $t = 16$	MCU suitability
RS(255, 223)	16 (MDS)	$O(n\,t + t^2)$	High	14.3%	Yes
BCH(255, 215)	5	$O(n\,t)$	Medium	15.7%	Yes
Hamming(15, 11)	1	$O(n)$	Low	26.7%	Yes
RaptorQ	No hard bound	$O(n)$	High	Variable	Partial
LDPC	Near-Shannon	$O(n \log n)$	High	Variable	No
Plain XOR-FEC	1 per pair	$O(n)$	Low	100% (1:1)	Yes

Tab. 4. Sweep over RS(255, k, t) $\times$ p configurations.

t	k	Overhead [%]	VMAF at $p = 1\%$	VMAF at $p = 5\%$	VMAF at $p = 10\%$	VMAF at $p = 15\%$	VMAF at $p = 20\%$	VMAF at $p = 25\%$
8	239	6.7	100.0	95.9	39.7	31.5	26.7	21.7
16	223	14.3	100.0	100.0	100.0	44.9	26.7	21.7
24	207	23.2	100.0	100.0	100.0	97.4	45.1	21.7
32	191	33.5	100.0	100.0	100.0	100.0	97.4	63.3
40	175	45.7	100.0	100.0	100.0	100.0	100.0	100.0

Algorithm 1 Adaptive FEC controller

Input: packet history H (RSSI, SNR, AckRate over window W) of size w, current frame type $F \in \{I, P, B\}$, jamming-detector thresholds $\theta_{RSSI}, \theta_{SNR}$.
Output: RS code parameters (n, k, t) for encoding frame F
Start

$emp_p \leftarrow \big(1 - \mathrm{mean}(AckRate, H)\big)$ ▷ empirical loss prob.
$med_{RSSI} \leftarrow \mathrm{median}(RSSI, H)$
$var_{RSSI} \leftarrow \mathrm{variance}(RSSI, H)$
$p_{hat} \leftarrow \alpha \cdot emp_p + (1 - \alpha) \cdot p_{prev}$ ▷ EWMA filter ($\alpha = 0.3$)
$L_{hat} \leftarrow \mathrm{autocorr_burst_length}(H)$
$j_{flag} \leftarrow (med_{RSSI} < \theta_{RSSI}) \cup (var_{RSSI} > \theta_{SNR})$ ▷ jamming detector
$w_F \leftarrow \mathrm{match}\ F : I \to 1.0\,; P \to 0.3\,; B \to 0.1$
$r_{base} \leftarrow \mathrm{match}\ F : I \to 0.4\,; P \to 0.2\,; B \to 0.1$
$\kappa \leftarrow \min(1 + 2 \cdot p_{hat} + L_{hat}/200, 3)$ ▷ adaptation function
$r \leftarrow \mathrm{clamp}(w_F \cdot r_{base} \cdot \kappa, 0.05, 0.6)$
if $j_{flag} = 1$ **then**
    $r \leftarrow \max(r, 0.5)$ ▷ jamming regime
end if
$n \leftarrow 255$ ▷ fixed for GF(256)
$t \leftarrow \lfloor r \cdot n/2 \rfloor$
$k \leftarrow n - 2 \cdot t$
return (n, k, t)

End

3.7. Experimental Verification and Results

The method was verified on a software testbed written in Python 3.11, using the following libraries: `pyfinite` $\geqslant 1.9$ for GF arithmetic and `reedsolo` $\geqslant 1.7$ for systematic RS encoding. The testbed parameters are summarized in Tab. 5.

Tab. 5. Software experimental testbed parameters.

Parameter	Value
Platform	Intel i7-12700H, 32 GB RAM, Linux Ubuntu 22.04
Language	Python 3.11
RS libraries	`pyfinite` $\geqslant 1.9$ (GF(256)), `reedsolo` $\geqslant 1.7$
Channel emulator	Custom GEJ implementation
Transport	UDP loopback with controlled latency / drop-rate
Test stream	Synthetic H.264-like, GOP = 30, $\rho = 6$, bitrate 0.5 – 10 Mbps
Quality metrics	VMAF (`libvmaf 2.3.1`), PSNR, SSIM
Integral metrics	MSR, TTFF, EFR
Runs per scenario	$K = 30$ with distinct seeds
Total sample size	4 scenarios $\times$ 5 solutions $\times$ 30 seeds = 600 runs

Each of the four scenarios S1 – S4 listed in Tab. 1 was run 30 times with different seeds from the random generator.

The testbed compares five solutions:

- UDP without FEC,
- Fixed RS(255, 223, 16),
- SRT with default settings,
- WFB-NG-emulation with RS 12/8,
- The proposed method using Algorithm 1.

Aggregate results are reported in the Tab. 6.

Three observations follow from Tab. 6. In the stable channel S1, all five solutions deliver comparable quality (VMAF of

Tab. 6. Aggregate experimental results (mean ± std over $K = 30$ channel emulator runs).

Metric	Scenario	UDP raw	Fixed RS	SRT	WFB-NG	Proposed
VMAF avg	S1	90 ±1	91±1	91 ±1	90 ±1	91 ±1
VMAF avg	S2	65 ±6	84±3	88 ±2	82 ±4	89 ±2
VMAF avg	S3	28 ±9	75±5	78 ±4	62 ±7	81 ±3
MSR [%]	S4	<10	40	50	60	78
Latency p95 [ms]	S2	60	80	350	90	95
Overhead [%]	S3	0	25	≈5 (ARQ)	33	30
TTFF [ms]	S4	—	800	1200	600	200

approx. 91), but the proposed method reaches this with an overhead of 8 - 12%, while fixed RS uses 22 – 25%, a 2- to 3-fold saving in bandwidth. In S2 and S3, where the loss rate is 5 – 20%, the proposed algorithm is consistently 5 – 19 VMAF points above fixed RS and WFB-NG, and matches SRT in terms of quality, while running with 3.5 times lower latency (95 vs. 350 ms).

In S4, where the channel is under sustained EW, MSR increases from 60% with WFB-NG to 78% with the proposed method, and TTFF – after the channel exits a jamming state – drops from 600 – 1200 ms to 200 ms, due to the directed I-frame protection and fast resync mechanism.

A two-tailed Student's t-test at significance level of $\alpha = 0.05$ confirms a statistically significant separation between the proposed approach and each of the baselines in scenarios S2 – S4. The proposed method vs. fixed RS and the proposed approach vs. WFB-NG generate p-values below 0.01, and the proposed vs. SRT in S3 results in $p = 0.03$.

To illustrate the recovery effect, Fig. 3 shows the result of transmitting a test CIF frame from `mobile_cif.y4m` (352 × 288, fragmented into 149 UDP packets of 1024 bytes each) at three loss rates: p = 5%, 15% and 25%.

At p = 5%, UDP raw degrades to PSNR = 15.6 dB and SSIM = 0.85, leaving the frame visually unintelligible. The adaptive scheme switches to RS(255, 223, 16) at an overhead of 14.3% and recovers the frame in full (PSNR = 99 dB, SSIM = 1.00). At p = 15%, the raw UDP drops to PSNR 11.9 dB, SSIM 0.76 and the adaptive scheme now selects RS(255, 191, 32) at an overhead of 33.5% and again returns PSNR 99 dB. At p = 25%, the raw UDP collapses to PSNR 8.2 dB, SSIM 0.44, the adaptive scheme moves to RS(255, 175, 40) at an overhead of 45.7% and recovers the frame.

3.8. Comparison with Recent Adaptive-FEC Methods

Table 7 illustrates the proposed method against representative published adaptive FEC systems along the dimensions that matter for a resource-constrained one-way link.

The features of the present method are the explicit GOP frame type differentiation and, above all, the closed-form analytical bound on total redundancy (Sec. 3.3), which none of the learning-based systems provides and which is what makes the scheme deployable within the fixed compute and energy budget of a microcontroller class node.

The price of this simplicity is the controller's reactivity: instead of predicting variations, it relies on recent physical-layer measurements to estimate the channel. At the onset of a new interference regime, this estimate lags, causing frames transmitted during the transition to be protected according to the previous, milder state.

This is the principal limitation of the approach and the reason why a predictive channel state estimator in the light of DeepRS [16] is identified as the first line of further work (Sec. 4). We stress that the comparison provided in Tab. 7 is qualitative: the quantitative gains reported for [15], [16], and [22] were obtained under different channels, codecs, quality metrics, and are therefore not directly commensurable with the results reported here.

4. Conclusions

The work reported in this paper develops and verifies a hybrid forward error correction method based on Reed–Solomon codes with adaptive control of the coding parameters for a one-way channel with non-stationary characteristics. The main findings can be summarized as follows.

On a benign real link, the three-state GEJ and the classical two-state Gilbert–Elliott model fit the loss-burst distribution comparably (Kolmogorov–Smirnov error 0.02 vs. 0.03), whereas on a real trace recorded under active periodic jamming the explicit jamming state is decisive (0.25 vs. 0.88).

This supports including an explicit jamming state precisely when the channel is subject to deliberate interference and shows that the third state is not required under benign conditions.

The closed-form Eq. (3) for the block decoding failure probability under the erasure model gives applicability bounds for representative RS configurations: RS(255, 231, 12) keeps $P_{\text{fail}} \leqslant 10^{-3}$ as long as $p \leqslant 5\%$, and RS(255, 191, 32) does so up to $p \leqslant 10\%$. For p above 20%, interleaving or frequency hopping is needed in addition to the code itself.

The adaptation function $\kappa(p, L) = \min\{1 + 2p + L/200, 3\}$ packs the packet loss probability and the average burst length into a single scalar coefficient. Due to the cap $\kappa_{\max} = 3$,

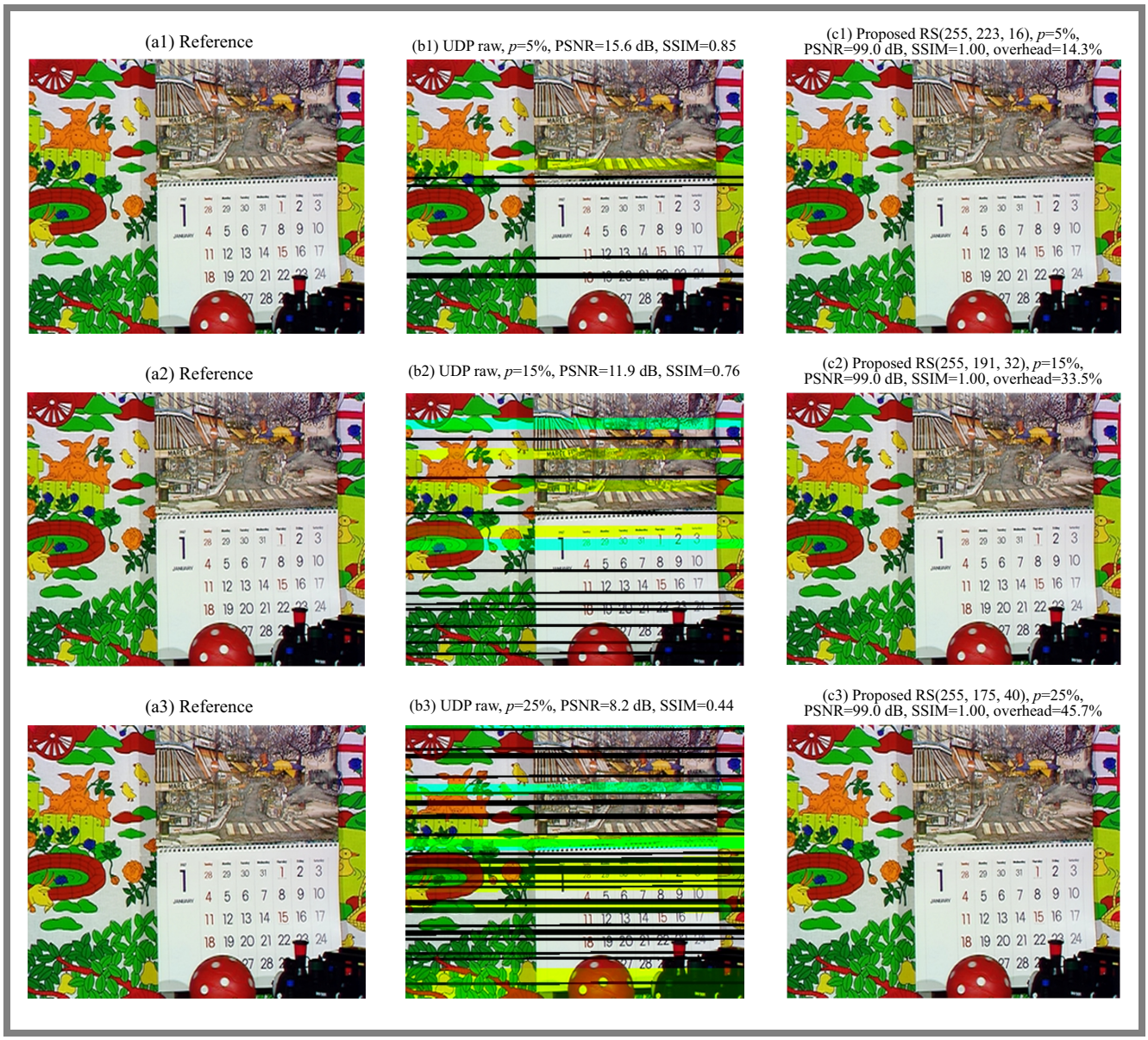


Fig. 3. Visual comparison of mobile CIF frame recovery under different loss rates: a) reference, b) raw UDP – horizontal stripes correspond to lost UDP packets, and c) proposed method with adaptive RS.

Tab. 7. Positioning of the proposed method relative to recent adaptive FEC systems.

Method	Adaptation driver	Rate granularity	Analytical overhead bound	Target platform	Reported gain
ABRF [15]	Joint bitrate FEC optimization	Uniform per frame	No	Server / GPU	Up to 68 – 95% less VMAF degradation
DeepRS [16]	LSTM loss predictor	Uniform per frame	No	Server / GPU	Proactive loss prediction
RL-AFEC [17]	Online reinforcement learning	Uniform per frame	No	Server	RL-tuned FEC in real time
R-FEC [18]	Online reinforcement learning	Uniform per frame	No	WebRTC endpoint	Improved QoE in WebRTC
GRACE [22]	Neural codec (loss-aware)	Codec level	No	GPU	≈95% fewer unrecoverable frames, +38% MOS
Proposed	PHY-layer estimate + GEJ state	Frame type differentiated (I/P/B)	Yes ($\leqslant 30\%$)	ESP32-S3 / ARM Cortex-M4	See Sec. 3.5 – 3.7

the overhead cannot grow without bound, no matter how degraded the channel becomes.

Frame-type-differentiated protection through Eq. (11) gives a total overhead $O_{\text{total}} = 23.4\%$, versus 40% for uniform

protection at the same I-frame rate. This is a relative reduction of 41.5% for a typical H.264 stream with GOP = 30 and $\rho = 6$.

Experiments conducted under the four scenarios (mobile 4G, stable Wi-Fi, tactical channel, active EW), with $K = 30$ runs each, show a statistically significant advantage of the proposed method. The MSR climbs from 60% with fixed RS and 50% with SRT to 78% with sustained jamming, TTFF falls from 600 – 1200 ms to 200 ms, and in stable channels the overhead is $2-3\times$ lower than for fixed RS.

The sweep over five configurations $t \in \{8, 16, 24, 32, 40\}$ and six loss rates $p \in \{1, 5, 10, 15, 20, 25\}$ [%] on the `mobile_cif.y4m` test vector pins down the applicability range of every configuration. It also shows that the adaptive scheme reaches VMAF $\geqslant 97$ at the minimum overhead needed in each case (between 6.7% and 45.7% depending on p), and avoids the $6.8\times$ bandwidth waste incurred by a fixed worst-case configuration.

In conclusion, the novel results of this work may be summarized as follows: (i) a closed-form expression for the block decoding failure probability under the UDP erasure model that links RS code parameters directly to channel characteristics, without requiring simulation, (ii) an adaptation function $\kappa(p, L)$ with a guaranteed upper bound, (iii) a frame-type-differentiated FEC control algorithm with a guaranteed 30% overhead bound that reduces total redundancy by 41.5% while keeping the I-frame protection at the same level, and (iv) an experimentally constructed Pareto chart of RS(255, k, t) configurations in the overhead – VMAF plane, from which five operating modes for the adaptive controller are read off.

The proposed solution applies to video surveillance of critical infrastructure, telemedicine, communication protocols for FPV UAVs and autonomous robotic systems, as well as videoconferencing services over unstable networks. The method runs on the ESP32-S3 (4 Mbps encoding) and ARM Cortex-M4 (1 – 3 Mbps) platforms, with a footprint of at most 1 KB of static data and 10 KB of dynamic memory.

Several lines of further work remain open. First, an LSTM channel-state predictor with a 5-step horizon, in the spirit of DeepRS [16], could be added on top of the present controller for proactive parameter adaptation. Second, the testbed can be extended to real H.265/AV1 streams with on-bench VMAF, PSNR and SSIM evaluation. Third, a cross-layer optimization following VOXEL [21], in which the encoder bitrate and the FEC parameters are jointly controlled, seems to be a natural next step for the future.

References

[1] Cisco, "Cisco Annual Internet Report (2018–2023)", White Paper, 2020 (https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html).

[2] AppLogic Networks (Sandvine), "Global Internet Phenomena Report 2024", 2024 (https://www.sandvine.com/global-internet-phenomena-report-2024).

[3] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, 2003 (https://doi.org/10.1109/TCSVT.2003.815165).

[4] G.J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, pp. 1649–1668, 2012 (https://doi.org/10.1109/TCSVT.2012.2221191).

[5] J. Han *et al.*, "A Technical Overview of AV1", *Proceedings of the IEEE*, vol. 109, pp. 1435–1462, 2021 (https://doi.org/10.1109/JPROC.2021.3058584).

[6] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, "Image Quality Assessment: from Error Visibility to Structural Similarity", *IEEE Transactions on Image Processing*, vol. 13, pp. 600–612, 2004 (https://doi.org/10.1109/TIP.2003.819861).

[7] Z. Wang and A.C. Bovik, "Mean Squared Error: Love It or Leave It? A New Look at Signal Fidelity Measures", *IEEE Signal Processing Magazine*, vol. 26, pp. 98–117, 2009 (https://doi.org/10.1109/MSP.2008.930649).

[8] R. Rassool, "VMAF Reproducibility: Validating a Perceptual Practical Video Quality Metric", *Proc. of IEEE Int. Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, Cagliari, Italy, 2017 (https://doi.org/10.1109/BMSB.2017.7986143).

[9] T. Stockhammer, M.M. Hannuksela, and T. Wiegand, "H.264/AVC in Wireless Environments", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 657–673, 2003 (https://doi.org/10.1109/TCSVT.2003.815167).

[10] S. Fu and Y. Zhang, "CRAWDAD Due/packet-delivery Dataset: Packet-delivery and Delay Measurements of a Low-power Wireless Link", *IEEE DataPort*, 2015 (https://ieee-dataport.org/collections/crawdad).

[11] I.S. Reed and G. Solomon, "Polynomial Codes Over Certain Finite Fields", *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, pp. 300–304, 1960 (https://doi.org/10.1137/0108018).

[12] S.B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs, NJ: Prentice Hall, 512 p., 1994 (ISBN 9780132008099).

[13] J.G. Proakis and M. Salehi, *Digital Communications*, 5th ed., New York: McGraw-Hill, 1150 p., 2007 (ISBN: 9780072957167).

[14] A. Badr, A. Khisti, W.-T. Tan, and J. Apostolopoulos, "Perfecting Protection for Interactive Multimedia: A Survey of Forward Error Correction for Low-delay Interactive Applications", *IEEE Signal Processing Magazine*, vol. 34, pp. 95–113, 2017 (https://doi.org/10.1109/MSP.2016.2639062).

[15] S. Cheng, H. Hu, and X. Zhang, "ABRF: Adaptive BitRate–FEC Joint Control for Real-time Video Streaming", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, pp. 5212–5226, 2023 (https://doi.org/10.1109/TCSVT.2023.3248164).

[16] S. Cheng, H. Hu, X. Zhang, and Z. Guo, "DeepRS: Deep-learning Based Network-adaptive FEC for Real-time Video Communications", *2020 IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Seville, Spain, 2020 (https://doi.org/10.1109/ISCAS45731.2020.9180974).

[17] K. Chen *et al.*, "RL-AFEC: Adaptive Forward Error Correction for Real-time Video Communication based on Reinforcement Learning", *Proc. of the 13th ACM Multimedia Systems Conference*, pp. 96–108, 2022 (https://doi.org/10.1145/3524273.3528184).

[18] I. Lee *et al.*, "R-FEC: RL-based FEC Adjustment for Better QoE in WebRTC", *Proc. of the 30th ACM International Conference on Multimedia*, pp. 2948–2956, 2022 (https://doi.org/10.1145/3503161.3548370).

[19] M. Rudow *et al.*, "Tambur: Efficient Loss Recovery for Videoconferencing via Streaming Codes", *Proc. of 20th USENIX Symposium on Networked Systems Design and Implementation*, pp. 953–971, 2023 (https://www.usenix.org/conference/nsdi23/presentation/rudow).

[20] S.L. Fong *et al.*, "Low-latency Network-adaptive Error Control for Interactive Streaming", *Proc. of the 27th ACM International Conference on Multimedia*, pp. 438–446, 2019 (https://doi.org/10.1145/3343031.3350942).

[21] M. Palmer *et al.*, "VOXEL: Cross-layer Optimization for Video Streaming with Imperfect Transmission", *Proc. of the 17th Inter-*

national Conference on emerging Networking EXperiments and Technologies, pp. 359–374, 2021 (https://doi.org/10.1145/3485983.3494864).

[22] Y. Cheng *et al.*, "GRACE: Loss-resilient Real-time Video through Neural Codecs", *ArXiv*, 2024 (https://doi.org/10.48550/arXiv.2305.12333).

[23] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "CRAWDAD Dataset Uportorwthaachen/vanetjamming2014: IEEE 802.11p Packet Traces Under Constant, Periodic and Reactive RF Jamming", *IEEE DataPort*, 2014 (https://doi.org/10.15783/C7Q306).

Leonid V. Kaliuzhnyi, Ph.D. Candidate
Department of Intelligent
Technologies and Telecommunications
https://orcid.org/0009-0009-8445-1143
E-mail: leonid.kalyuzhnyy@gmail.com
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
https://suitt.edu.ua

Viktor B. Rusalovskiy
Department of Intelligent
Technologies and Telecommunications
https://orcid.org/0009-0003-6134-531X
E-mail: it.ddo.od@gmail.com
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
https://suitt.edu.ua

Artem D. Ishchenko
Department of Intelligent
Technologies and Telecommunications
https://orcid.org/0009-0000-5161-7818
E-mail: artem.ischchenko@gmail.com
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
https://suitt.edu.ua

Vasyl M. Didenko
Department of Intelligent
Technologies and Telecommunications
https://orcid.org/0009-0008-9517-7963
E-mail: vahaod@gmail.com
State University of Intelligent Technologies and Telecommunications, Odesa, Ukraine
https://suitt.edu.ua

Energy-delay Aware Data Collection in Mobile Sink-based Wireless Sensor Networks with Optimized Visiting Points and Two-level Data Aggregation

Nadjib Benaouda[1], Boubakeur Moussaoui[1,2], and Oussama Senouci[1]

[1] *University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria,*
[2] *Mohammed Boudiaf University, M'sila, Algeria*

https://doi.org/10.26636/jtit.2026.3.2540

Submitted: 1 Mar. 2026, Accepted: 14 Aug. 2026

Abstract — This study addresses data collection challenges in wireless sensor networks (WSNs) employing mobile sinks (MSs). Although MSs improve flexibility and scalability, achieving balanced optimization of data collection delay and energy consumption remains challenging, as minimizing one objective typically comes at the expense of the other. To overcome this limitation, we propose the optimized grid-based data collection with mobile sink (ODCMS) approach. In ODCMS, the MS follows a predefined trajectory and visits strategically selected visiting points (VPs) to collect aggregated data. The number of VPs is minimized to reduce tour length and thus latency, while a two-level data aggregation scheme shortens transmission distances and lowers forwarding overhead, thereby decreasing energy consumption. Energy balancing is ensured through periodic rotation of aggregation roles to prevent early node depletion. The MS movement among VPs and the base station (BS) is modeled as a connected graph, and its trajectory is constructed as a Hamiltonian cycle using a backtracking algorithm. ODCMS also supports rapid reporting of urgent events to the BS prior to MS arrival. NS-3 simulations compared with EDEDA and EFDC demonstrate that ODCMS achieves shorter tour length and data collection delay, lower energy consumption, and longer network lifetime than other approaches.

Keywords — data aggregation, grid-based structure, mobile sink, path planning, visiting points, WSNs

1. Introduction

Wireless sensor networks (WSNs) comprise distributed autonomous nodes that monitor physical or environmental conditions and transmit collected data to a central base station (BS) or sink for processing. WSNs are fundamental to modern networks, forming the backbone of the Internet of Things (IoT), with applications in precision agriculture, industrial automation, healthcare, environmental monitoring, and smart cities [1]–[4].

Energy efficiency remains a major challenge in WSNs, as sensor nodes are typically battery-powered. In traditional architectures with static sinks, nodes located near the sink must handle excessive data forwarding, creating hotspot areas [5], [6]. This uneven load results in rapid energy depletion of these nodes, which consequently shortens the network lifetime.

Mobile sinks (MSs) have emerged as a promising solution. Depending on the application, they may take the form of unmanned aerial vehicles (UAVs) [7], [8], unmanned ground vehicles (UGVs) [9], autonomous underwater vehicles (AUVs) [10], or devices carried by humans or animals [11].

A MS can reduce energy consumption, balance energy usage, and extend network lifetime by collecting data from multiple locations [3], [5], [6], [11]. In agricultural WSNs, mobile drones replace fixed sinks to collect aggregated data from sensor nodes and reduce long-range transmissions [7], [12], [13]. Similarly, in forest fire monitoring, drone mobility enables efficient network-wide data collection, reducing energy consumption and reliance on multi-hop communication [14].

This work addresses the periodic collection of data by a MS from all WSN nodes. A straightforward approach requires the MS to visit every node, which results in excessively long collection times due to extensive travel. To overcome this, strategically selected locations, called visiting points (VPs), can be used, enabling the MS to gather data efficiently from nearby sensors.

Although this reduces the collection period, identifying the optimal set of VPs and designing an efficient MS route that simultaneously minimizes both delay and energy consumption remains a challenging task [15]. To reduce data collection delay, it is crucial to minimize the number of VPs required for full network coverage and to establish the optimal trajectory for the MS to visit each VP, starting and returning to the BS.

However, too few VPs may force sensor nodes to transmit data over long distances, either via direct long single-hop transmissions, where nodes increase transmission power to reach the MS, or multi-hop forwarding through intermediate nodes. In both cases, energy consumption rises sharply, accelerating network-wide energy depletion.

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

In response to these challenges, this paper proposes the optimized grid-based data collection with mobile sink (ODCMS) approach. Its primary goals are to minimize data collection delay, reduce energy consumption, and prevent energy imbalance among sensor nodes. ODCMS achieves this by carefully minimizing the number of VPs required for full network coverage, while employing a hierarchical two-level data aggregation scheme that effectively shortens transmission distances and limits forwarding load. Aggregation roles are periodically rotated to evenly balance energy usage across the network, thereby significantly extending its overall lifetime.

ODCMS organizes the sensor field into a virtual grid, partitioning the deployment area based on node communication ranges and region size. The grid dimensions are defined in accordance with design objectives to support the desired selection of VPs. Each grid cell has a designated grid cell head (GCH) that aggregates data from local nodes. At a higher level, one GCH in each quad of four cells is selected as the grid quad head (GQH), which further aggregates the data and prepares it for collection by the MS.

In addition, strategically chosen grid intersection points serve as VPs. The MS begins its tour at the BS and follows a predetermined trajectory visiting each VP exactly once. Its possible movements between VPs and the BS are modeled as a connected graph, and a valid data collection route is determined as a Hamiltonian cycle using a backtracking algorithm. At each VP, the MS gathers data from the four surrounding quads via a two-level aggregation process. This continues until all VPs are visited, after which the MS returns to the BS with the aggregated data.

To further enhance responsiveness, ODCMS also integrates a complementary mechanism to promptly deliver urgent event data directly to the BS via multi-hop communication, thereby avoiding delays associated with waiting for the MS's arrival.

The main contributions of this work are:

1) **Grid-based clustering** – the sensing field is divided into virtual grid cells, enabling structured clustering and efficient VP selection.
2) **Hierarchical aggregation** – each cell elects a GCH, and one GCH per quad becomes the GQH to deliver aggregated quad data to the MS, reducing transmission distances, forwarding load, and balancing energy via periodic role rotation.
3) **VP selection strategy** – minimizes the number of VPs required for full coverage and thus, reducing data collection delay.
4) **Trajectory planning** – models MS movements as a connected graph and computes a Hamiltonian cycle using a backtracking approach.
5) **Event-driven reporting** – supports urgent data delivery directly to the BS without waiting for the MS tour.
6) **Simulation-based evaluation** – ODCMS is implemented in NS-3 and compared with the state-of-the-art EDEDA [15] and EFDC [16] protocols across multiple performance metrics.

This article is structured as follows. Section 2 reviews related work and discusses their limitations. Section 3 presents the network model along with the key assumptions considered. Section 4 introduces ODCMS, while Section 5 provides its detailed description. Section 6 presents the performance evaluation, and finally, Section 7 concludes the article.

2. Related Works

This section reviews recent related studies, emphasizing indirect data collection through VPs rather than direct node-by-node collection methods [17]–[19].

The authors in [20] proposed VGRSS, a virtual grid-based method for selecting rendezvous points (RPs) and sojourn locations to improve energy efficiency and data collection in WSNs. The approach operates in two phases: the sensor field is partitioned into grid cells, then a grid cell coordinator (GCC) is elected from the cell members using a fuzzy inference system that accounts for residual energy and node centrality. Each GCC aggregates and stores data from associated nodes for the MS. To prevent premature node failure, GCCs' energy is monitored each round to determine if replacements are needed. The MS visits sojourn locations at grid intersections, following a trajectory constructed via the travelling salesman problem using Christofides heuristics [21].

In [15], the authors proposed EDEDA, an energy- and delay-efficient data collection approach based on a virtual grid. The sensing area is divided into grid cells, and a grid cell head (GCH) is elected from grid cell members (GCMs) based on proximity to the cell center and residual energy. A subset of cells is selected as visiting cells (VCs), allowing the MS to collect data from the GCHs of nine adjacent cells in one hop. The MS follows a Hamiltonian cycle starting and ending at the BS, while a GCH replacement strategy prevents premature node failure.

In [22], the authors proposed a virtual grid-based MS data collection method in which each cell elects a GCH. The MS follows paths along grid peripheries and diagonals, selecting one cell per path as a rendezvous grid cell (RGC). In each RGC, a rendezvous GCH (RGCH) broadcasts the MS location, enabling GCHs to adjust minimum-hop routes and efficiently forward aggregated data to the MS.

In [16], the authors proposed EFDC, an energy-efficient data-gathering protocol for WSNs with a MS. The approach employs hierarchical agglomerative clustering (HAC) to form clusters and select the optimal clustering level. CHs are chosen as the nodes closest to cluster centroids. The VPs for the MS are designated as the coverage intersection points (CIPs) located along the boundaries of the cluster communication ranges. The MS travels to these CIPs, halts for a specified duration to collect data, and then continues its trajectory using the traveling salesman problem with neighborhoods (TSPN) algorithm.

In [23], the authors proposed E2OSLMS, a method to determine optimal MS locations within clusters, ensuring balanced energy consumption among nodes and preventing energy

holes. The approach assumes a circular sensing area, which is divided into coronas, each containing multiple clusters. Cluster heads are elected based on their distance to the cluster center and residual energy. Four MS traverse a spiral trajectory through the designated clusters and return to the starting point. Within each cluster, the precise MS location is determined from member node positions using a fuzzy inference system, enabling efficient, balanced, and reliable data collection across the network.

In [24], an energy-aware path construction (EAPC) approach is proposed to establish an energy-efficient data-gathering path while enhancing the selection of data collection points (CPs). This construction is led under the constraint of a predetermined path length provided as input. The algorithm comprises three primary phases: initialization, selection of CPs, and path construction. A minimal spanning tree (MST) is established during the initialization phase to minimize the total path length. The collection point selection phase determines CPs using a benefit index that optimizes energy usage while minimizing the forwarding burden on sensor nodes. The most direct route between the BS and the CPs is finally generated during the path construction phase.

Article [25], introduced BIIE, an energy-efficient data collection algorithm for WSNs. The algorithm operates in three phases: initialization, path establishment, and local re-clustering. The network is first divided into clusters, with the maximum number determined using a dichotomy method under MS path length constraints. During path establishment, particle swarm optimization (PSO) identifies optimal Rendezvous Nodes (RNs) and the MS trajectory, minimizing energy consumption imbalance among nodes. Finally, local re-clustering dynamically adjusts cluster structures based on residual energy, dividing or merging clusters to ensure balanced energy usage, thereby enhancing network lifetime and reliability.

A hybrid data collection method combining rendezvous- and cluster-based approaches is presented in [26] to achieve energy efficiency while meeting delivery deadlines for critical packets. Rendezvous-based collection reduces energy by storing data at rendezvous nodes (RNs) until the MS arrives, whereas cluster-based collection forwards data via multi-hop paths, consuming more energy but reducing delay. The method operates in three phases: clustering nodes based on density and residual energy, selecting strategic points near cluster centers, and determining Steiner points that connect these points. PSO groups Steiner points, and MSs visit each to collect data, which is then transmitted to a central collection point.

An optimal RPs selection with MS trajectory construction algorithm is proposed in [27]. This approach, called ORPSTC, begins by constructing a minimum-cost spanning tree (MST) of the WSN graph. The MST is subsequently partitioned into several clusters, and a CH is selected within each cluster to serve as the RP. Upon identifying the RPs, a computational geometry approach is employed to determine the route of the MS linking these RPs. A RPs reselection process is devised to maintain equitable energy consumption among sensor nodes, with the consideration of virtual RPs (VRPs). VRPs are sensor nodes that can directly communicate with the MS.

In [28], an enhanced MS path planning method for homogeneous and heterogeneous WSNs is proposed, supporting single and multiple MSs. The sensing area is divided into equal regions, and nodes are clustered using the proposed stable election algorithm (SEA) to reduce frequent CH rotations.

In homogeneous networks, CHs are selected based on neighbor count, while in heterogeneous networks, residual energy prioritizes super-nodes. CH positions are reported to the BS to compute MS paths. Visiting points are determined via a minimum weighted vertex cover formulation, and MS trajectories are optimized using multi-objective evolutionary algorithms, including ACO, GA, and SA.

The authors of [29] outline the following method named SCTR to ascertain a minimum number of RPs. The WSN was represented as an undirected graph G. They subsequently determined the connected components of G through the application of breadth-first search (BFS) traversal.

The reasoning for this is to diminish the problem's size, hence decreasing the computing complexity of the solution. The larger connected components are then partitioned into manageable units (MUs) using the combined use of a heuristic and Tarjan's technique. Upon acquisition of these MUs, spectral clustering is employed to create clusters from the MUs. The centroid of each cluster is then identified as RP. The identified RPs are ultimately employed to construct the traveling salesman tour (TSP) for the MS utilizing Christofides' algorithm.

2.1. Discussion

Focusing on minimizing energy consumption and data collection latency, the reviewed articles can be grouped into three main classes, as outlined in Tab. 1.

- **Low/medium energy – medium/high latency:** Methods such as EDEDA, VGRSS, E2OSLMS, SCTR, EFDC, and [28] conserve energy by reducing multi-hop transmissions and control overhead but may increase latency due to more VPs and node waiting time. Energy consumption may also depend on cluster size, as larger clusters increase transmission distances between members and their CHs.
- **High energy – low latency:** Approaches such as TARA and ORPSTC lower data delivery delay by minimizing VPs or maintaining continuous shortest multi-hop paths, at the cost of increased energy from frequent forwarding and control messages.
- **Adaptive/variable solutions:** Methods like EAPC, BIIE, and HRCM balance energy and latency according to application requirements.

Despite existing proposals, jointly optimizing data collection delay and energy consumption remains challenging, as improving one often increases the other. Proposed approach tackles this trade-off by reducing the number of VPs to shorten delays and limiting long-distance and multi-hop transmissions to save energy, effectively balancing both objectives.

Tab. 1. Summary of the reviewed articles.

Method	Structure type	Multi-sink support	Path selection strategy	Energy consumption	Data acquisition latency
EDEDA [15]	Grid	No	Hamiltonian path planning strategy	Low	Medium
EFDC [16]	Cluster	No	Traveling salesman problem with neighborhoods (TSPN)	Low	Medium
VGRSS [20]	Grid	No	Christofides algorithm	Low	Medium
TARA [22]	Hybrid (grid & tree)	No	Computation-free path strategy	High	Low
E2OSLMS [23]	Ring	Yes	Computation-free path strategy	Low	Medium
EAPC [24]	Cluster	No	Convex polygon algorithm	Variable	Variable
BIIE [25]	Cluster	No	Particle swarm optimization	Variable	Variable
HRCM [26]	Cluster	Yes	Dijkstra shortest path algorithm	Variable	Variable
ORPSTC [27]	Cluster	No	Author-defined computational geometry approach	High	Low
Approach in [28]	Cluster	Yes	Multi-objective evolutionary algorithm (MOEA)	Low	Medium
SCTR [29]	Cluster	No	Christofides algorithm	Low	Medium

3. Network Model and Assumptions

This study adopts the following network model and assumptions:

- n stationary sensor nodes are uniformly and randomly deployed in a square area of side L.
- Each node has a maximum communication range R_c, forming a unit-disk graph (UDG), where two nodes i and j are directly connected if their Euclidean distance

$$d(i,j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

  satisfies $d(i,j) \leqslant R_c$.
- A single MS collects data from sensor nodes by visiting predefined VPs, starting and ending at a fixed BS, which stores the collected data.
- Each node knows its location and starts with equal residual energy.
- The MS can reliably communicate with nodes within range and has sufficient energy to complete its tour.
- The MS trajectory is unobstructed.

4. Overview of the Proposed Approach

ODCMS is guided by the following design principles. The network is partitioned using a $k \times k$ virtual grid, based on area size L and node range R_c. The value of k is required to be a multiple of four in order to support the desired selection of VPs. VPs are chosen at grid intersections, each covering sixteen adjacent cells (four quads), as shown in Fig. 1.

Each cell elects a GCH for local aggregation. Among the four GCHs in a quad, the one closest to the VP becomes the GQH, which aggregates data from the other GCHs and sends it to the MS. GCHs and GQHs are periodically re-elected to prevent early failures.

The MS tours the network from the BS, visiting each VP to collect aggregated data, then returns to the BS. The trajectory is modeled as a connected graph, and a Hamiltonian cycle is computed via backtracking. Urgent data are sent directly to the BS without waiting for the MS.

5. In-depth Description of the Proposed Approach

The proposed approach operates through seven key operations. The first step consists of designing the virtual grid. The second identifies the set of VPs, while the third determines how GCHs and GQHs are selected. The fourth operation

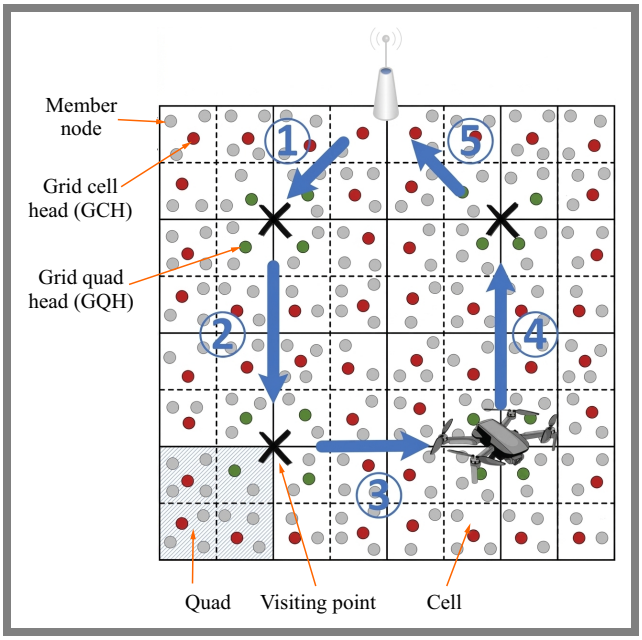


Fig. 1. Overview of ODCMS.

Tab. 2. List of messages used.

Message	Description
GCH_ELECTION	Initiates the election of GCHs/GQHs
GCH_ANNOUNCEMENT	Used by GCHs/GQHs to announce their roles
ASKING_DATA	Used by the MS to trigger data collection from nodes near a VP
DATA	Used to transmit data through the hierarchy from nodes to the MS
GCH_REELECTION	Used by GCHs/GQHs when they are unable to fulfill their designated roles
URGENT_EVENT	Used by sensor nodes to inform the BS of an emergency event

defines the trajectory of the MS, and the fifth explains the data-gathering process. The sixth describes the reelection mechanism for GCHs and GQHs. The seventh addresses the strategy for handling urgent events. These operations are elaborated further in the following sections.

The messages supporting the implementation of proposed solution, are listed in Tab. 2.

5.1. Virtual Grid Design

The virtual grid is created by partitioning the sensing area into square cells based on the nodes' communication range and the sensing area's dimensions.

The grid consists of k rows and k columns, forming a $k \times k$ square grid. The value of k is determined using the adjustment factor α:

$$\alpha = \frac{2\sqrt{2}L}{R_c} \,. \tag{1}$$

Given the value of α, the number k can be identified. It must be a multiple of four to allow the selection of the VPs as desired. It is defined as follows:

$$k = \begin{cases} 4, & \text{if } \alpha \leqslant 4 \\ 8, & \text{if } 4 < \alpha \leqslant 8 \\ 12, & \text{if } 8 < \alpha \leqslant 12 \\ 16, & \text{if } 12 < \alpha \leqslant 16 \\ \dots \end{cases} . \tag{2}$$

It is worth emphasizing that this grid formation ensures communication among nodes in neighboring cells.

To make possible the execution of the subsequent steps, the cell identifier $IdCell_i$ of every sensor node i needs to be calculated. This calculation is made using the coordinates (x_i, y_i) of i, according to the following formula:

$$IdCell_{\text{i}} = pk + q \,, \tag{3}$$

where $p = \lfloor \frac{y_i}{a} \rfloor$ and $q = \lfloor \frac{x_i}{a} \rfloor$, with a corresponding to the side length of a cell.

The value of a is calculated using the following formula:

$$a = \frac{L}{K} \,.$$

Figure 2 depicts the visual representation of an 8×8 virtual grid, showcasing the distinct identifiers assigned to each individual cell.

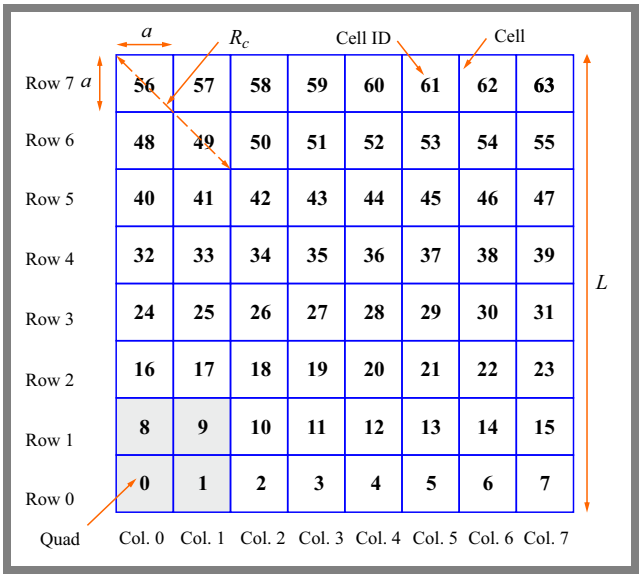


Fig. 2. Virtual grid design.

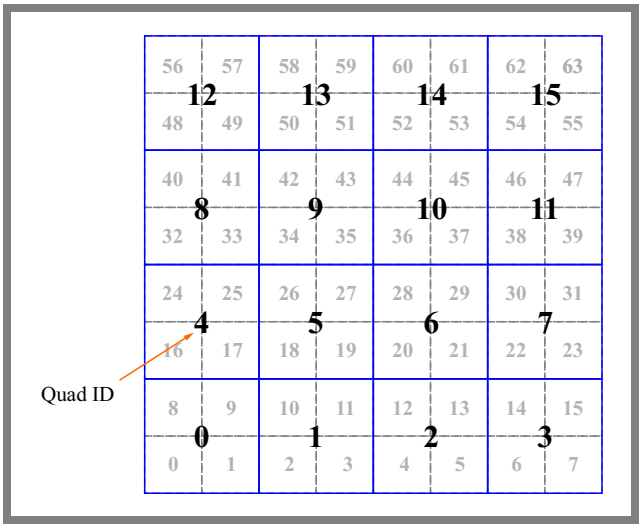


Fig. 3. Identification of the grid's quads.

In addition to cell identifiers, each sensor node i must know the identifier of the quad $IdQuad_i$ to which it belongs. Based on its coordinates, node i computes it as follows:

$$IdQuad_{\text{i}} = p' \frac{k}{2} + q' \,, \tag{4}$$

where $p' = \lfloor \frac{y_i}{2a} \rfloor$ and $q' = \lfloor \frac{x_i}{2a} \rfloor$.

Figure 3 illustrates the identifier assigned to each quad in an 8×8 grid.

5.2. Visiting Points Identification

Several specific intersection points between four neighboring cells are selected as VPs. Each VP is chosen to enable the MS to cover 16 adjacent cells, corresponding to four adjacent quads. The identification of these VPs is performed using Algorithm 1, which is based on the following formula:

$$m = 2a(2z + 1) \,, \tag{5}$$

where $z = 0, 1, \dots, k/4 - 1$.

Algorithm 1 Selection of VPs

Input: a, k

Output: φ: list of identified VPs coordinates

$\varphi \leftarrow \emptyset$
$i \leftarrow 0$
while $i < \frac{k}{4}$ **do**
  $x \leftarrow 2 \times a \times (2 \times i + 1)$
  $j \leftarrow 0$
  while $j < \frac{k}{4}$ **do**
    $y \leftarrow 2 \times a \times (2 \times j + 1)$
    $\varphi \leftarrow \varphi \cup \{(x, y)\}$
    $j \leftarrow j + 1$
  end while
  $i \leftarrow i + 1$
end while

Return φ

The possible values of m correspond to the potential X-intercept and Y-intercept values of the distinct coordinates of the determined VPs, which are henceforth denoted by the set φ. Figure 4a-b illustrates the coordinates of all VPs identified in 8×8 and 12×12 grids, respectively, by applying the Algorithm 1. For example, in a 8×8 grid ($k = 8$), m can take two values: $2a$ and $6a$ according, respectively, to the following values of z (i or j): 0 and 1. As a result, the different coordinates of the four identified VPs are as follows: $(2a, 2a)$, $(6a, 2a)$, $(2a, 6a)$ and $(6a, 6a)$.

5.3. GCHs and GQHs Election

Once the virtual grid is established, clusters naturally form, enabling the selection of GCHs and GQHs. Upon network deployment, each node broadcasts a `GCH_ELECTION` message containing its ID, coordinates, cell, and quad identifiers. Neighboring nodes record this information to maintain awareness of nearby nodes, including those within the same cell.

Within each cell, the node closest to the geometric center is elected as the GCH, responsible for local data collection and initial aggregation. In each quad, the GCH of the cell nearest the designated VP is elected as the GQH, serving as a rendezvous node that further aggregates quad data and delivers it to the MS.

This central positioning reduces transmission power requirements for cluster members and ensures GQHs remain near their associated VPs. Elected GCHs/GQHs broadcast a `GCH_ANNOUNCEMENT` to notify neighbors of their roles.

While GCHs can transmit directly to the MS, forwarding data through GQHs reduces transmission distances, aggregates correlated data, conserves energy, and lowers the relaying workload, thereby improving network efficiency.

5.4. Path Planning

The MS starts and ends its tour at the BS, visiting all VPs along a pre-established path. To determine this route, the BS models possible MS movements between VPs as a connected graph $G' = \{V', E'\}$, where V' includes all VPs and the BS, and E' represents feasible direct movements, as illustrated in

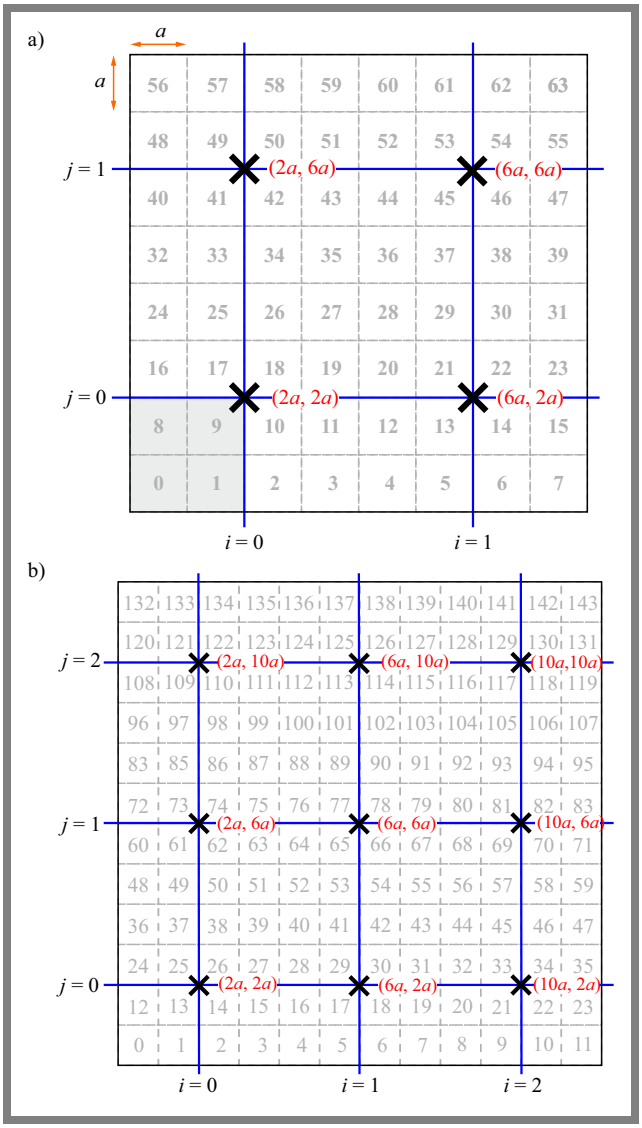


Fig. 4. VPs identification for various grid sizes: a) 8×8 grid and b) 12×12 grid.

Fig. 5, which shows the connected graph used to determine the MS's path in a 12×12 grid.

A valid Hamiltonian cycle, visiting each VP exactly once, is then identified using a backtracking algorithm [30], though other optimization methods can also be applied [3]. The algorithm incrementally builds the cycle by adding adjacent, unvisited vertices and backtracks when no valid extension exists, continuing until a complete cycle is found.

Although finding a Hamiltonian cycle is an NP-complete problem with factorial worst-case complexity, the computational overhead at the BS remains manageable in ODCMS. This is primarily because ODCMS reduces the number of VPs and restricts feasible movements to directly connected neighboring vertices in G', thereby limiting the search space explored by the backtracking algorithm. Consequently, minimizing the number of VPs not only reduces the MS tour length but also alleviates the computational burden of route construction.

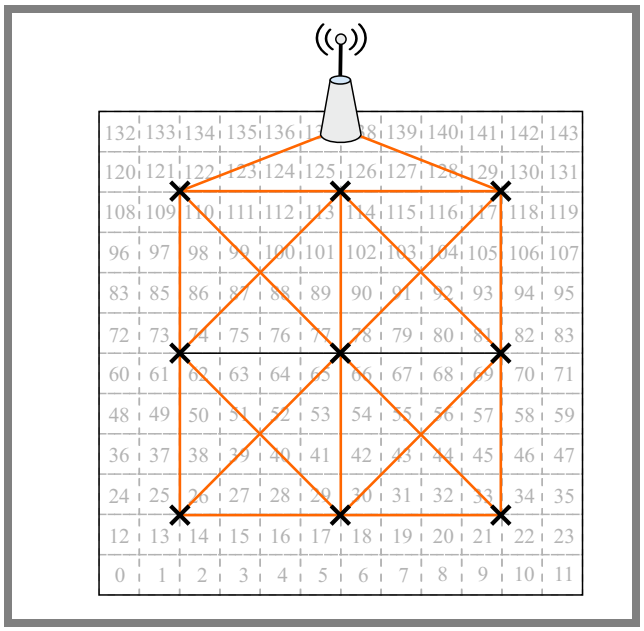


Fig. 5. Connected graph for determining the MS path in a 12×12 grid.

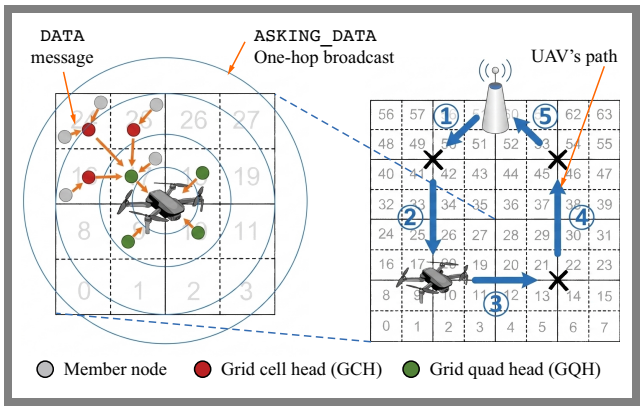


Fig. 6. Data collection process.

5.5. Data Collection

The MS periodically follows the predetermined Hamiltonian cycle, broadcasting an `ASKING_DATA` message at each VP to reach all nodes in the four associated quads. GCHs schedule `DATA` transmissions to their GQH after a short wait to gather member data, and GQHs subsequently forward aggregated data to the MS after a longer wait, ensuring complete collection.

Each `ASKING_DATA` message specifies the targeted cell IDs to limit requests to relevant nodes. At each VP, the MS collects data from the four neighboring GQHs, then moves to the next VP, repeating the process until returning to the BS.

Figure 6 illustrates this data collection procedure.

5.6. Re-election of GCHs and GQHs

GCHs and GQHs are periodically reelected to prevent premature failure and ensure continuous data collection. When the residual energy of a GCH or GQH drops below a predefined threshold $E_{\text{threshold}}$, the node initiates a reelection process by broadcasting a `GCH_REELECTION` message. The current GCH/GQH is excluded from candidacy, and a new leader is selected from among the nodes closest to the cell center that possess sufficient residual energy.

Tab. 3. Simulation parameters used to analyze the proposed system.

Parameter	Value
Sensing field dimensions $L \times L$	120 ×v120, 160 × 160, 200 × 200, 240 × 240, 280 × 280, 320 × 320 m^2
Number of sensor nodes	200 – 400
Location of BS	$(\frac{L}{2}, 0)$
Communication range R_c	50 m
Initial energy per node	1 J
Energy threshold $E_{\text{threshold}}$	40%
Data packet size	512 bytes
Mobile sink speed	10 m/s
Duration between sink tours T_{rest}	1 hour
Sink sojourn duration in a VP	10 s
Simulation duration	Until first node dies
E_{elec}	50 nJ/bit
E_{mp}	0.0013 pJ/bit/m^4
E_{Fs}	10 pJ/bit/m^2

5.7. Urgent Events Handling

In emergency scenarios, when a sensor node detects a measurement exceeding a predefined threshold, it encapsulates the measurement within an emergency notification message (`URGENT_EVENT`) and transmits it to the GCH of its respective cell. To mitigate false alarms arising from sensor faults or anomalies, the GCH validates the urgency of the event by requiring the reception of multiple corroborative `URGENT_EVENT` messages from distinct member nodes within the same cluster.

Upon validation, the GCH initiates a unicast transmission of the `URGENT_EVENT` message directly to the BS employing a geographic routing protocol, thereby bypassing the inherent latency associated with awaiting the MS arrival. This direct transmission strategy significantly reduces the end-to-end latency for critical event reporting within the network.

6. Performance Evaluation

6.1. Simulation Setup

To evaluate the proposed approach, ODCMS was implemented in NS-3, incorporating its message types, data structures, and protocol operations. For comparison, EDEDA [15] was also implemented, as it similarly employs a virtual grid and mobile sink, with the sink visiting selected visiting cells (VCs) to collect data from nearby nodes.

In addition, EFDC [16] was implemented, which employs the HAC algorithm for cluster formation and directs the MS through coverage intersection points (CIPs), serving as VPs for data collection.

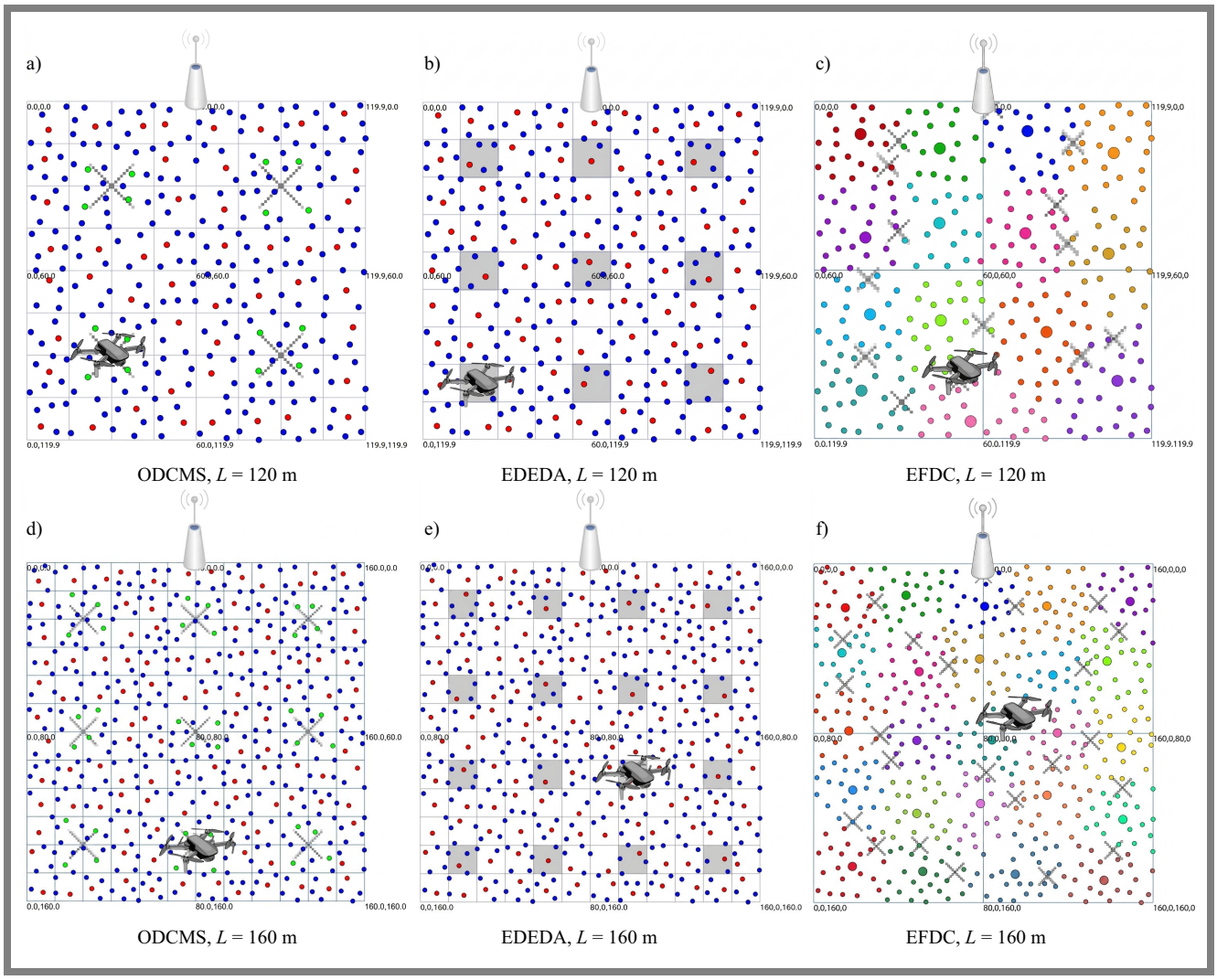


Fig. 7. NetAnim snapshots showing the configuration of the simulated network after the execution of ODCMS, EDEDA, and EFDC for different terrain side lengths.

Extensive simulations were conducted using the parameters in Tab. 3. The number of sensor nodes (200 – 400) was varied to evaluate energy consumption and network lifetime, while the terrain size L was adjusted to assess its impact on MS data collection delay. All other parameters were kept constant.

To reflect a realistic scenario, we considered a fire detection WSN assisted by a drone acting as the MS. Sensors were uniformly deployed, and the drone periodically collected data from designated VPs/VCs before returning to the BS. Data collection occurred hourly, and simulations ran until the first node died.

Before presenting the results, we briefly describe the adopted energy model and the evaluation metrics used for comparison.

6.2. Energy Consumption Model

To evaluate energy consumption, we adopted the same radio energy model as in [20]. The transmission energy for sending a p-bit message over a distance d, denoted by $E_{TX}(p,d)$, and the reception energy $E_{RX}(p)$ are computed as follows:

$$E_{TX}(p,d) = \begin{cases} pE_{\text{elec}} + pE_{\text{fs}}d^2, & d < d_0 \\ pE_{\text{elec}} + pE_{\text{mp}}d^4, & d \geqslant d_0 \end{cases}, \tag{6}$$

$$E_{RX}(p) = pE_{\text{elec}}\,, \tag{7}$$

where E_{elec} represents the electronics energy per bit, while E_{fs} and E_{mp} denote the free-space and multi-path amplification factors, respectively.

The threshold distance is defined as:

$$d_0 = \sqrt{\frac{E_{\text{fs}}}{E_{\text{mp}}}}\,. \tag{8}$$

Energy consumed in sensing and processing is neglected, since transmission dominates overall energy dissipation.

6.3. Comparison Metrics

The performance is evaluated using three primary metrics: total energy consumption, network lifetime, and data collection duration. We also report complementary metrics, including energy consumption per message type (`ASKING_DATA` and

DATA), MS tour length, and the number of VPs/VCs, to better explain variations in the main results and provide deeper insight into protocol behavior.

The total energy consumption represents the total energy consumed by all sensor nodes during the simulation. It is computed as the sum of the difference between the initial and residual energy of each node i:

$$E_{\text{total}} = \sum_{i=1}^{n} \left(E_{init_i} - E_{resid_i}\right) . \quad (9)$$

Network lifetime is defined as the period during which the network remains operational before performance degradation occurs, such as node death, disconnection, or coverage loss [31]. In this study, we adopt the commonly used definition: the time from the start of the simulation until the first sensor node depletes its energy.

Data collection duration is defined as the total time required for the MS to depart from the BS, visit all VPs to gather data, and return to the BS. It includes both travel time between VPs and the pause time spent collecting data at each stop. This duration can be expressed as follows:

$$D_{\text{collection}} = \sum_{i=1}^{|\varphi \cup \{\text{BS}\}|} \left(\frac{d(\text{SL}(i), \text{SL}(i+1))}{sink\ speed}\right) + \sum_{j=1}^{|\varphi|} Pt_j . \quad (10)$$

The stopping location i is represented by $\text{SL}(i)$, while the pause time or sink stay duration at VP j is denoted by Pt_j.

6.4. Results and Discussion

In the following, we present the simulation results using graphs and NetAnim visualizations, highlighting the performance of ODCMS, EDEDA and EFDC under varying network conditions. Each result represents the average of 10 runs with different random seeds to ensure reliability (Fig. 7).

To ensure fair comparison across scenarios with different durations, total energy consumption is normalized by dividing it by the simulation time (in seconds).

Total energy consumption increases with network density for all protocols, as more nodes generate greater communication traffic and overhead (Fig. 8a). ODCMS consistently consumes less energy than EDEDA. For 200 nodes, ODCMS consumes 9.30 $\times 10^{-5}$ J/s compared with 1.09 $\times 10^{-4}$ J/s for EDEDA, representing a reduction of approximately 14.68%. At 400 nodes, the corresponding values are 2.41 $\times 10^{-4}$ J/s and 2.83 $\times 10^{-4}$ J/s, yielding a reduction of approximately 14.84%.

EFDC exhibits the highest energy consumption, averaging 1.60 $\times 10^{-4}$ J/s at 200 nodes and 4.19 $\times 10^{-4}$ J/s at 400 nodes. This is primarily attributed to its larger clusters, which require member nodes to transmit data over longer distances. The greater number of visiting points in EFDC e.g., 14 at $L = 120$ m and 24 at $L = 160$ m, further increases overhearing of asking for data messages (Fig. 8b).

The impact of network density on the lifetime of ODCMS, EDEDA, and EFDC under identical conditions, with 200–400 nodes in a fixed 120×120 m area is presented in Fig. 8d.

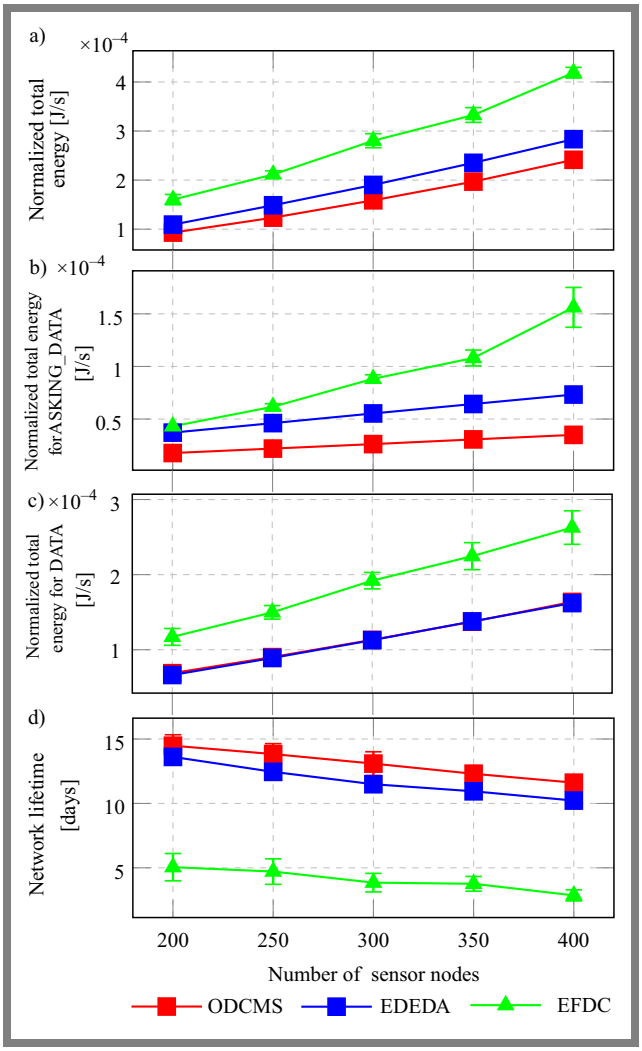


Fig. 8. Comparison of simulation results for ODCMS, EDEDA, and EFDC protocols across multiple scenarios (with standard deviation error bars).

Network lifetime decreases with node density due to increased communication activity and energy depletion. ODCMS consistently outperforms both protocols. At 200 nodes, it achieves 14.48 days, compared with 13.61 days for EDEDA ($\approx 6.4\%$ longer) and 5.05 days for EFDC ($\approx 186.73\%$ longer). At 400 nodes, ODCMS reaches 11.61 days, compared with 10.22 days for EDEDA ($\approx 13.6\%$ longer) and 2.86 days for EFDC ($\approx 305.94\%$ longer). This advantage stems from ODCMS's improved energy efficiency, while EFDC lacks CH rotation and incurs additional overhead from overhearing data and MS data-request messages.

Figure 9 illustrates the resulting impact on data collection duration for different values of L, while adjusting the number of sensor nodes to maintain a constant density of 2 nodes per 100 m^2. As L varies, the grid size ($k \times k$) changes accordingly, which in turn affects the number of VPs in ODCMS and VCs in EDEDA.

EDEDA and EFDC consistently require longer collection times than ODCMS, with EFDC exhibiting the highest delays. This advantage results from ODCMS minimizing the number of VPs and, consequently, the MS tour length (Fig. 9b). Table 4

Tab. 4. MS tour length, data collection delay, and parameters for ODCMS, EDEDA, and EFDC protocols, along with percentage reductions.

L [m]		120	160	200	240	280	320
Nodes		288	512	800	1152	1568	2048
ODCMS	$k \times k$	8 × 8	12 × 12	12 × 12	16 × 16	16 × 16	20 × 20
	VPs	4	9	9	16	16	25
	Tour [m]	264.85	590.10	737.63	1139.44	1329.35	1958.94
	Delay [min]	1.10	2.48	2.72	4.56	4.83	7.43
EDEDA	$k \times k$	9 × 9	12 × 12	12 × 12	15 × 15	18 × 18	21 × 21
	VCs	9	16	16	25	36	49
	Tour [m]	442.58	759.63	949.54	1469.20	1948.61	2661.71
	Delay [min]	2.24	3.93	4.25	6.62	9.25	12.6
EFDC	VPs	14	24	40	54	68	102
	Tour [m]	336.77	622.36	1023.68	1416.84	1928.43	2624.42
	Delay [min]	2.89	5.04	8.37	11.36	14.55	21.37
Red. (vs. EDEDA)	Tour [%]	40.16	22.32	22.32	22.44	31.78	26.40
	Delay [%]	50.89	36.90	36.00	31.12	47.78	41.03
Red. (vs. EFDC)	Tour [%]	21.36	5.18	27.94	19.58	31.07	25.36
	Delay [%]	61.94	50.79	67.50	59.86	66.80	65.23

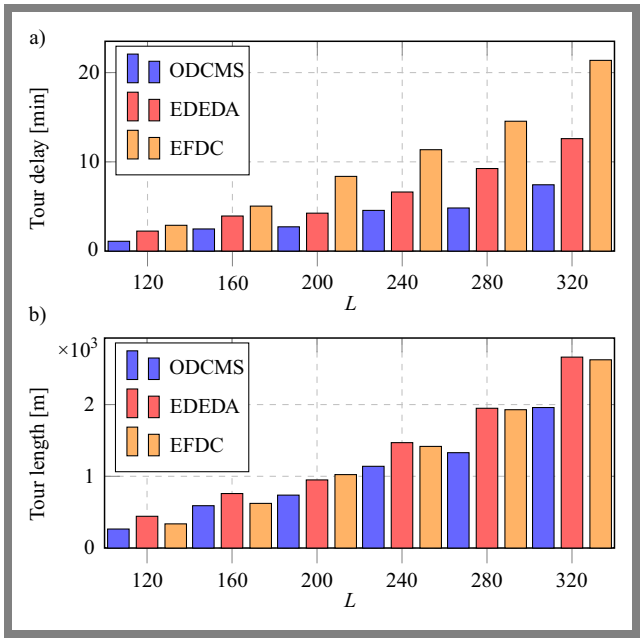


Fig. 9. Tour delay a) and tour length comparisons b) for ODCMS, EDEDA, and EFDC under different terrain dimensions.

confirms that ODCMS selects fewer VPs than both protocols for the same L, resulting in shorter tours and collection times. Although EFDC's tour lengths can be close to those of EDEDA at larger L values, its collection time remains substantially higher because it uses more VPs (14 – 102), leading to more MS stops.

In urgent events handling experiment, we validate ODCMS's urgent event handling by randomly selecting 20 sensor nodes, each generating a single urgent event during a 24-hour simulation, considering two protocol variants. The first variant, ODCMS-direct, sends urgent events immediately to the BS via multi-hop, while ODCMS-deferred stores events at GCHs and forwards them only when the MS collects data. System performance is measured by the end-to-end delay from event generation to BS reception. In ODCMS-direct, this reflects multi-hop forwarding via the GCH, while in ODCMS-deferred, it is dominated by the MS travel time to the VP and back to the BS. In ODCMS-deferred, the MS resting period T_{rest} strongly affects urgent event delay.

Figure 10a shows the CDF of latency for different T_{rest} values, indicating that longer rests lead to higher end-to-end delays, as events must wait for the MS to resume its tour. Figure 10b shows the CDF of latency for ODCMS-direct and ODCMS-deferred ($T_{rest} = 0$). ODCMS-direct exhibits millisecond delays, while ODCMS-deferred reaches minutes due to the MS traveling to VPs, retrieving data, and returning to the BS. These results highlight the necessity of dedicated mechanisms to handle time-critical events when MSs are employed for data collection.

7. Conclusions

This study addressed fixed-path data collection in WSNs using a MS. We proposed ODCMS, which reduces collection delay and energy consumption while extending network lifetime by minimizing VPs and mitigating costly multi-hop and long-distance transmissions via a two-level data aggregation scheme. ODCMS combines virtual grid-based clustering, selective VP selection, hierarchical aggregation, and backtracking-based MS path planning, with urgent events sent directly to the BS.

NS-3 simulations comparing ODCMS with EDEDA and EFDC show improved tour length, delay, energy efficiency, and lifetime. Future work will investigate more realistic

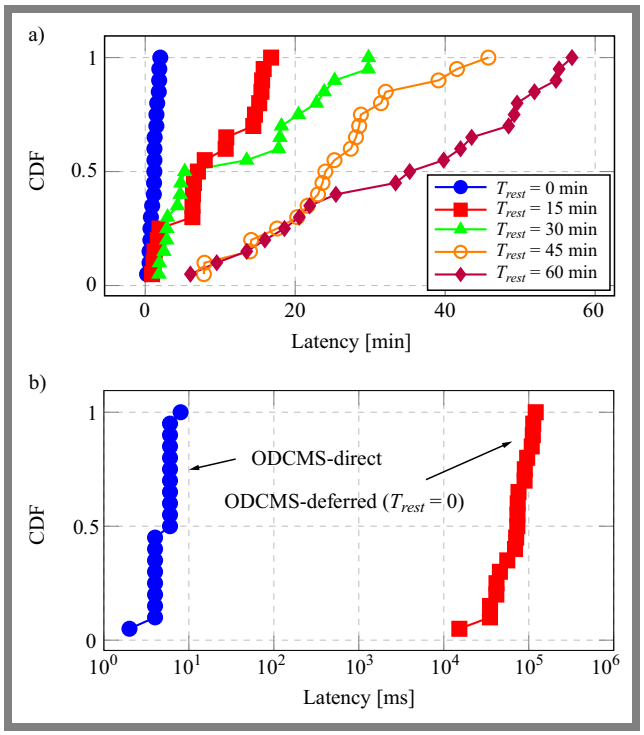


Fig. 10. CDF of latency under ODCMS-deferred for different values of T_{rest} a) and CDF of latency comparison between ODCMS-direct and ODCMS-deferred ($T_{rest} = 0$) b).

deployment scenarios, including obstacle-constrained MS mobility and non-uniform node distributions, as well as multiple MSs and path-length constraints for VP selection and route optimization.

References

[1] K. Gulati *et al.*, "A Review Paper on Wireless Sensor Network Techniques in Internet of Things (IoT)", *Materials Today: Proceedings*, vol. 51, pp. 161–165, 2022 (https://doi.org/10.1016/j.matpr.2021.05.067).

[2] D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of Wireless Sensor Networks: An Up-to-date Survey", *Applied System Innovation*, vol. 3, art. no. 14, 2020 (https://doi.org/10.3390/asi3010014).

[3] V. Agarwal, S. Tapaswi, and P. Chanak, "A Survey on Path Planning Techniques for Mobile Sink in IoT-enabled Wireless Sensor Networks", *Wireless Personal Communications*, vol. 119, pp. 211–238, 2021 (https://doi.org/10.1007/s11277-021-08204-w).

[4] N. Benaouda, "Efficient Routing for Delay-energy Tradeoff in Event-based Wireless Sensor Networks", *Journal of Telecommunications and Information Technology*, vol. 98, pp. 69–77, 2024 (https://doi.org/10.26636/jtit.2024.4.1833).

[5] N. Temene, C. Sergiou, C. Georgiou, and V. Vassiliou, "A Survey on Mobility in Wireless Sensor Networks", *Ad Hoc Networks*, vol. 125, art. no. 102726, 2022 (https://doi.org/10.1016/j.adhoc.2021.102726).

[6] S. Jain, R.K. Verma, K.K. Pattanaik, and A. Shukla, "A Survey on Event-driven and Query-driven Hierarchical Routing Protocols for Mobile Sink-based Wireless Sensor Networks", *Journal of Supercomputing*, vol. 78, pp. 11492–11538, 2022 (https://doi.org/10.1007/s11227-022-04327-4).

[7] P.K. Singh and A. Sharma, "An Intelligent WSN-UAV-based IoT Framework for Precision Agriculture Application", *Computers and Electrical Engineering*, vol. 100, art. no. 107912, 2022 (https://doi.org/10.1016/j.compeleceng.2022.107912).

[8] S.M. Bozorgi *et al.*, "An Impressive Clustering Based on the Fuzzy System for UAV-assisted IoT Wireless Networks", *Telecommunication Systems*, vol. 88, art. no. 79, 2025 (https://doi.org/10.1007/s11235-025-01307-w).

[9] M. Xia *et al.*, "WSN Lifetime Maximization: Effects of Energy-sharing and UGV Mobility", *China Communications*, vol. 22, pp. 236–253, 2025 (https://doi.org/10.23919/JCC.fa.2024-0431.202504).

[10] S. Kumari, P.K. Mishra, and V. Anand, "Coverage and Connectivity Aware Deployment Scheme for Autonomous Underwater Vehicles in Underwater Wireless Sensor Networks", *Wireless Personal Communications*, vol. 132, pp. 909–931, 2023 (https://doi.org/10.1007/s11277-023-10642-7).

[11] R. Hamidouche *et al.*, "Classical and Bio-inspired Mobility in Sensor Networks for IoT Applications", *Journal of Network and Computer Applications*, vol. 121, pp. 70–88, 2018 (https://doi.org/10.1016/j.jnca.2018.07.010).

[12] D. Popescu *et al.*, "Advanced UAV-WSN System for Intelligent Monitoring in Precision Agriculture", *Sensors*, vol. 20, art. no. 817, 2020 (https://doi.org/10.3390/s20030817).

[13] K. Medani, C. Gherbi, H. Mabed, and Z. Aliouat, "Energy-efficient Q-learning-based Path Planning for UAV-Aided Data Collection in Agricultural WSNs", *Internet of Things*, vol. 33, art. no. 101698, 2025 (https://doi.org/10.1016/j.iot.2025.101698).

[14] N. Abramov *et al.*, "Intelligent Methods for Forest Fire Detection Using Unmanned Aerial Vehicles", *Fire*, vol. 7, art. no. 89, 2024 (https://doi.org/10.3390/fire7030089).

[15] R.K. Verma and S. Jain, "Energy and Delay Efficient Data Acquisition in Wireless Sensor Networks by Selecting Optimal Visiting Points for Mobile Sink", *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 11671–11684, 2023 (https://doi.org/10.1007/s12652-022-03729-9).

[16] F. Hasan *et al.*, "Energy-efficient Data Collection in Mobile Sink-based Wireless Sensor Networks Using the Hierarchical Clustering Method", *2023 26th International Conference on Computer and Information Technology (ICCIT)*, Cox's Bazar, Bangladesh, 2023 (https://doi.org/10.1109/ICCIT60459.2023.10441076).

[17] A.A. Somasundara, A. Ramamoorthy, and M.B. Srivastava, "Mobile Element Scheduling with Dynamic Deadlines", *IEEE Transactions on Mobile Computing*, vol. 6, pp. 395–410, 2007 (https://doi.org/10.1109/TMC.2007.57).

[18] R. Sugihara and R.K. Gupta, "Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks", *IEEE Transactions on Mobile Computing*, vol. 9, pp. 127–139, 2009 (https://doi.org/10.1109/TMC.2009.113).

[19] B. Yuan, M. Orlowska, and S. Sadiq, "On the Optimal Robot Routing Problem in Wireless Sensor Networks", *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, pp. 1252–1261, 2007 (https://doi.org/10.1109/TKDE.2007.1062).

[20] A. Mehto, S. Tapaswi, and K.K. Pattanaik, "Virtual Grid-based Rendezvous Point and Sojourn Location Selection for Energy and Delay Efficient Data Acquisition in Wireless Sensor Networks with Mobile Sink", *Wireless Networks*, vol. 26, pp. 3763–3779, 2020 (https://doi.org/10.1007/s11276-020-02293-4).

[21] D.S. Johnson and L.A. McGeoch, "Experimental Analysis of Heuristics for the STSP", in *The Traveling Salesman Problem and Its Variations*, Springer, Boston, MA, pp. 369–443, 2007 (https://doi.org/10.1007/0-306-48213-4_9).

[22] A. Mehto, R.K. Verma, and S. Jain, "Efficient Trajectory Planning and Route Adjustment Strategy for Mobile Sink in WSN-assisted IoT", *2022 IEEE Region 10 Symposium (TENSYMP)*, Mumbai, India, 2022 (https://doi.org/10.1109/TENSYMP54529.2022.9864434).

[23] K. Pitchai, "Determination of Energy Efficient Optimal Sojourn Location of Mobile Sinks in Clustered Corona-based Wireless Sensor Networks", *Peer-to-Peer Networking and Applications*, vol. 15, pp. 1–12, 2022 (https://doi.org/10.1007/s12083-021-01224-0).

[24] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: Energy-aware Path Construction for Data Collection Using Mobile Sink in Wireless Sensor Networks", *IEEE Sensors Journal*, vol. 18, pp. 890–901, 2018 (https://doi.org/10.1109/JSEN.2017.2773119).

[25] X. Fu and X. He, "Energy-balanced Data Collection with Path-constrained Mobile Sink in Wireless Sensor Networks", *AEU - International Journal of Electronics and Communications*, vol. 127, art. no. 153504, 2020 (https://doi.org/10.1016/j.aeue.2020.153504).

[26] Y.M. Raghavendra and U.B. Mahadevaswamy, "Hybrid Rendezvous Clustering Model for Efficient Data Collection in Multi Sink Based Wireless Sensor Networks", *Wireless Personal Communications*, vol. 129, pp. 837–851, 2023 (https://doi.org/10.1007/s11277-022-10158-6).

[27] B. Gutam, P.K. Donta, C. Sekhar, and Y.-C. Hu, "Optimal Rendezvous Points Selection and Mobile Sink Trajectory Construction for Data Collection in WSNs", *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 7147–7158, 2021 (https://doi.org/10.1007/s12652-021-03566-2).

[28] B.R. Al-Kaseem, Z.K. Taha, S.W. Abdulmajeed, and H.S. Al-Raweshidy, "Optimized Energy-efficient Path Planning Strategy in WSN With Multiple Mobile Sinks", *IEEE Access*, vol. 9, pp. 82833–82847, 2021 (https://doi.org/10.1109/ACCESS.2021.3087086).

[29] R. Anwit, A. Tomar, and P.K. Jana, "Scheme for Tour Planning of Mobile Sink in Wireless Sensor Networks", *IET Communications*, vol. 14, pp. 430–439, 2020 (https://doi.org/10.1049/iet-com.2019.0613).

[30] J. Sleegers and D. van den Berg, "Backtracking Algorithms on the Hamiltonian Cycle Problem", *ArXiv*, 2021 (https://doi.org/10.48550/arXiv.2107.00314).

[31] F. Engmann *et al.*, "Prolonging the Lifetime of Wireless Sensor Networks: A Review of Current Techniques", *Wireless Communications and Mobile Computing*, vol. 2018, 2018 (https://doi.org/10.1155/2018/8035065).

Nadjib Benaouda, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-4361-9597
E-mail: nadjib.benaouda@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Boubakeur Moussaoui, Ph.D., Associate Professor
Laboratory of Informatics and its Applications of M'sila
https://orcid.org/0000-0003-4991-5044
E-mail: b.moussaoui@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz
Mohammed Boudiaf University, M'sila, Algeria
https://www.univ-msila.dz

Oussama Senouci, Ph.D., Associate Professor
Computer Science Department
https://orcid.org/0000-0002-5345-0713
E-mail: oussama.senouci@univ-bba.dz
University Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Harmonic Trust Spectrum Modeling for Cognitive Radio Intrusion Detection

S. Jeyaseelan[1], P.G.S. Velmurugan[1], G. Prabhakar[1], and J. Shanthi[2]

[1] *Thiagarajar College of Engineering, Madurai, India,*
[2] *Kamaraj College of Engineering and Technology, Virudhunagar, India*

https://doi.org/10.26636/jtit.2026.3.2747

Submitted: 23 Jul. 2026, Accepted: 12 Aug. 2026

Abstract — Cognitive radio networks (CRNs) are highly vulnerable to spectrum spoofing, PU emulation (PUE), and harmonic RF manipulation attacks, which significantly degrade dynamic spectrum access reliability and communication security. This paper proposes a harmonic trust spectrum modeling framework for cognitive radio intrusion detection using multi-domain RF intelligence and adaptive machine learning. The proposed system integrates harmonic spectrum analysis, trust oscillation modeling, spectral entropy analysis, wavelet decomposition, and power spectral density (PSD) characterization to detect anomalous RF behaviors. Three heterogeneous RF datasets were used for experimental evaluation: the RadioML 2016.10A modulation dataset, the Oracle RF fingerprinting IQ dataset, normal signals generated by GNU Radio, and malicious PUE RF signals. The proposed methodology extracts advanced harmonic trust features including spectral entropy, spectral flatness, harmonic trust instability index (HTII), and oscillatory trust divergence (OTD), which are subsequently processed using an adaptive XGBoost-based intrusion detection system. Experimental results demonstrate that malicious RF users exhibit significantly higher spectral instability, irregular harmonic distributions, elevated entropy behavior, and oscillatory trust divergence, compared to legitimate users. Furthermore, studies covering fast Fourier transform (FFT) harmonic spectra, PSD analysis, wavelet coefficient analysis, trust oscillation curves, and confusion matrix evaluations confirm the effectiveness of the proposed framework. The proposed harmonic trust spectrum model provides a computationally efficient and scalable solution for next-generation secure cognitive radio communications and RF cyber defense systems.

Keywords — cognitive radio networks, harmonic trust modeling, intrusion detection system, PU emulation, XGBoost

1. Introduction

Cognitive radio networks (CRNs) have emerged as a promising solution for overcoming spectrum scarcity and improving wireless spectrum utilization through intelligent and dynamic spectrum access mechanisms. Traditional fixed spectrum allocation policies often lead to inefficient spectrum usage, as a part of licensed frequency bands remains underutilized, while unlicensed users experience severe bandwidth limitations. Cognitive radio (CR) technology addresses this issue by allowing secondary users to opportunistically access vacant spectrum bands without causing interference to PUs (PU) [1], [2].

Spectrum sensing plays a critical role in CRN environments, since accurate detection of PU activity directly affects communication reliability, spectrum efficiency, and network security. Several spectrum detection techniques, such as energy detection, cyclostationary feature detection, matched filtering, and wavelet-based sensing have been extensively investigated recently [3], [4]. Among these approaches, deep learning and deep learning-based spectrum sensing techniques have gained significant attention due to their ability to learn complex RF patterns and improve detection performance in noisy and dynamic wireless environments [5]–[7]. Despite these advances, CRNs remain highly vulnerable to malicious attacks, particularly PU emulation (PUE), spectrum spoofing, and RF impersonation. In PUE attacks, malicious users mimic the characteristics of PUs to illegally occupy spectrum resources and disrupt normal communication behaviors. These attacks degrade the reliability of spectrum sensing, reduce spectrum utilization efficiency, and compromise the security of intelligent wireless communication systems [1], [3], [8]. Moreover, traditional energy-based sensing techniques are often insufficient for identifying sophisticated harmonic spoofing behaviors, as attackers are capable of imitating legitimate RF signatures with a high degree of accuracy.

Recent studies have explored advanced signal processing approaches including wavelet decomposition, empirical mode decomposition, compressive sensing, and cyclostationary analysis to improve spectrum sensing robustness [9], [10], [11]. Matched filter detection techniques with adaptive thresholds have also demonstrated improved sensing accuracy in dynamic wireless environments [2], [12]. However, existing approaches focus primarily on conventional spectrum occupancy detection and often neglect harmonic trust behavior, oscillatory RF instability, and spectral divergence patterns associated with malicious RF users.

To mitigate these limitations, this paper proposes a novel harmonic trust spectrum modeling (HTSM) framework for CR intrusion detection. The proposed approach integrates harmonic RF analysis, trust oscillation modeling, spectral entropy characterization, wavelet decomposition, and adaptive machine learning techniques to distinguish legitimate spectrum users from malicious PUE attackers.

The overall architecture of the proposed HTSM framework is illustrated in Fig. 1.

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

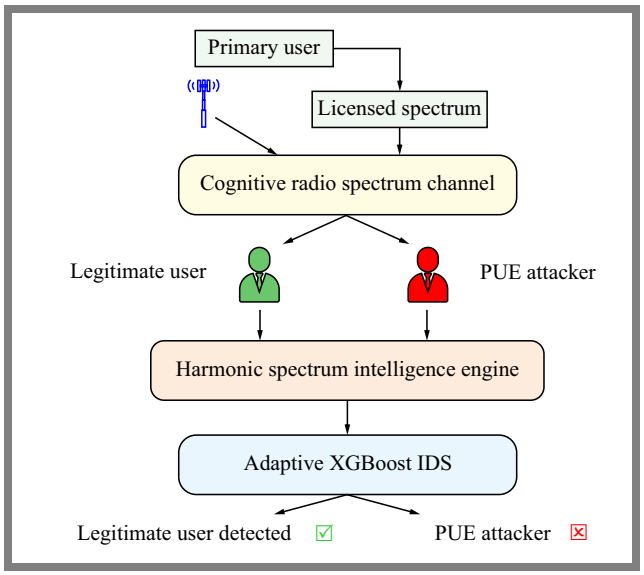


Fig. 1. System architecture of the proposed HTSM framework.

As shown in Fig. 1, the licensed spectrum is initially occupied by the PU, while both legitimate secondary users (SUs) and malicious PUE attackers attempt to access the CR channel. The observed RF transmissions are collected by a spectrum monitoring node and forwarded to the harmonic spectrum intelligence engine for spectral feature extraction. The extracted trust-aware harmonic features are subsequently processed by an adaptive XGBoost-based intrusion detection system to identify legitimate spectrum access and malicious spectrum spoofing activities.

The proposed framework introduces two novel metrics, i.e. harmonic trust instability index (HTII) and oscillatory trust divergence (OTD), for quantifying spectral instability and malicious oscillatory divergence in RF trust behavior. HTII measures the harmonic instability of RF spectral distributions, while OTD evaluates the divergence between legitimate and malicious trust harmonics across frequency-domain representations. These metrics enable an accurate characterization of spoofed RF behavior and enhance intrusion detection performance in CRN environments.

To validate the effectiveness of the proposed framework, experiments were conducted using three heterogeneous RF data sets. An experimental analysis focusing on fast Fourier transform (FFT), power spectral density (PSD), wavelet analysis, spectral entropy computation, trust oscillation modeling, and confusion matrix evaluation, was performed to assess the proposed framework. The extracted RF features were subsequently utilized to evaluate the effectiveness of the proposed intrusion detection methodology under various spectrum sensing scenarios.

An adaptive XGBoost-based intrusion detection model was developed using the extracted harmonic trust features to classify legitimate and malicious RF behaviors. Experimental results demonstrate that malicious RF users exhibit significantly higher spectral entropy, unstable harmonic distributions, elevated trust oscillations, and larger harmonic divergence compared to legitimate users. The proposed framework achieved effective RF anomaly detection performance while preserving computational complexity.

The remainder of this paper is organized as follows. Section 2 presents the related works. Section 3 presents the architecture of the proposed system. Section 4 presents the proposed harmonic trust spectrum modeling framework and feature extraction methodology. Section 5 presents the experimental results and performance evaluation. Finally, Section 6 concludes the article and discusses future research directions.

2. Related Works

The authors of [9] proposed a spectrum sensing framework based on random forest classification combined with wavelet transform and empirical mode decomposition. Their approach demonstrated improved sensing performance under varying channel conditions. The superior results were achieved by exploiting machine learning-based feature extraction techniques. Similarly, in [1] a comprehensive review of spectrum sensing in CR Internet of Things (CR-IoT) environments is presented and the challenges associated with reliable spectrum access, security threats, and intelligent sensing strategies are discussed.

The authors of [3] reviewed recent advances in spectrum sensing and highlighted the importance of robust sensing algorithms for future CR systems. In [4], a detailed review of conventional spectrum sensing approaches is provided including energy detection, matched filtering, and cyclostationary feature detection techniques. Deep learning-based spectrum sensing methods have also gained attention due to their ability to learn complex RF characteristics. Article [5] investigates deep learning classification techniques for CR spectrum sensing and demonstrates improved sensing accuracy in dynamic wireless environments.

Advanced signal processing techniques have been extensively explored to enhance spectrum sensing performance, e.g. in [10], [11], where compressive sensing and various radio spectrum monitoring approaches in CRNs are researched, emphasizing reduced sampling requirements and efficient spectrum monitoring. Paper [2] presents a comprehensive survey on spectrum management in CRNs and discusses spectrum sharing, sensing, and dynamic access strategies. Furthermore, in [12], a matched filter detection approach with adaptive threshold selection is proposed to improve sensing reliability under varying channel conditions, while in [13], the effectiveness of cyclostationary feature analysis is demonstrated to identify unique RF signatures in CR applications.

Recent studies have also explored machine learning-based intrusion detection approaches. The authors of [9] utilized the XGBoost algorithm combined with SHAP analysis for attack detection and explainable security analytics. In [12], the KNN and ANN-based methods are proposed to detect PU emulation attacks (PUEA) in CRNs is proposed. Similarly, in [13], a machine learning-based intrusion detection system is presented that is capable of identifying anomalous network behavior using supervised learning techniques.

Tab. 1. Comparison of existing CR security and spectrum sensing approaches.

Ref.	Method	Technique	Limitation
[1]	CR-IoT spectrum sensing and spectrum management survey	CR-IoT survey	No intrusion detection framework
[2]	Spectrum management survey for CRNs	Survey	No RF intrusion detection framework
[3]	Review of recent advances in spectrum sensing	Spectrum sensing	No trust-aware RF analysis
[4]	Comparative survey of conventional spectrum sensing methods	Survey	Limited security considerations
[5]	Deep learning-based RF signal classification for spectrum sensing	Deep learning	Limited attack detection capability
[6]	XGBoost and SHAP based attack detection framework	XGBoost along with SHAP	General IoT security application
[7]	Machine learning based intrusion detection system	Machine learning IDS	Not specifically designed for CRNs
[8]	KNN and ANN based PUE attack detection	KNN + ANN	Limited harmonic spectrum intelligence
[9]	Random forest with wavelet and EMD based spectrum sensing	RF + wavelet + EMD	No intrusion detection capability
[10]	Compressive spectrum sensing for CRNs	Compressive sensing	No intrusion detection support
[11]	Comprehensive review of spectrum sensing techniques	Spectrum sensing	No malicious user identification capability
[12]	Matched filter detection with adaptive thresholding	Matched filtering	Vulnerable to sophisticated spoofing attacks
[13]	Cyclostationary feature extraction for RF signature identification	Cyclostationary analysis	No trust behavior modeling

The comparison of existing CR security and spectrum sensing approaches is provided in Tab. 1

Although existing studies have demonstrated significant progress in spectrum sensing, RF signal classification, and intrusion detection, most approaches focus primarily on spectrum occupancy detection, signal classification, or conventional machine learning-based attack detection. Limited attention has been paid to harmonic trust behavior, oscillatory RF instability, spectral divergence analysis, and trust-aware harmonic modeling. To address these limitations, the proposed HTSM framework integrates harmonic spectrum intelligence, trust oscillation analysis, spectral entropy characterization, and adaptive XGBoost-based intrusion detection to provide enhanced protection against malicious RF activities and PUE attacks.

3. Proposed Model

Unlike conventional spectrum sensing approaches that mainly rely on energy detection or signal strength measurements, the proposed framework investigates harmonic patterns, spectral uncertainty, trust oscillations, and spectrum occupancy behavior to identify abnormal RF activities.

The overall framework begins with the acquisition of RF signals from the selected benchmark datasets. The collected RF signals are then processed through a spectrum monitoring stage, where both time- and frequency-domain characteristics are analyzed to capture the underlying behavior of spectrum users.

FFT is utilized to reveal the harmonic spectrum structure of the received signals, while PSD analysis provides information regarding signal energy distribution across the frequency spectrum. Spectral entropy is used to measure the randomness and uncertainty present within RF transmissions, whereas wavelet analysis captures localized spectral variations and transient behaviors that may indicate malicious activity. The combination of these techniques enables comprehensive multidomain RF signal characterization.

The extracted spectral information is forwarded to the harmonic spectrum intelligence engine which constitutes the main processing analytical component of the proposed framework. This module evaluates harmonic stability, spectral consistency, transmitter reliability, and spectrum occupancy behavior to construct a trust-aware representation of each RF transmission. By continuously monitoring these spectral attributes, the framework establishes a dynamic trust profile capable of distinguishing legitimate spectrum occupancy from suspicious RF activities.

To evaluate the intrusion detection capability, two metrics are introduced. The first is the harmonic trust instability index (HTII), which quantifies the degree of instability within the harmonic spectrum. Legitimate spectrum users generally exhibit stable harmonic structures, whereas malicious users often generate irregular spectral fluctuations due to spoofing behavior and transmission inconsistencies. Consequently, higher HTII values indicate increased harmonic instability and greater likelihood of malicious activity.

The second metric, i.e. oscillatory trust divergence (OTD), measures the deviation between trusted harmonic behavior and suspicious RF transmissions. This metric captures the spectral divergence observed between legitimate users and potential attackers by analyzing variations in harmonic energy distributions and trust oscillations. Larger OTD values indicate substantial deviations from expected spectrum behavior and serve as indicators of spectrum spoofing and PUE attacks.

The extracted features are combined to form a multidimensional feature representation of each spectrum user. These features are subsequently fed to an adaptive extreme gradient boosting (XGBoost)-based intrusion detection system (IDS). The classifier learns discriminative patterns associated with legitimate and malicious RF activities and performs automated classification of spectrum users.

The proposed HTSM framework establishes a unified relationship between harmonic spectrum intelligence, trust-based signal analysis, and machine learning-based intrusion detection. By introducing trust-oriented RF metrics and integrating them with advanced signal processing techniques, the framework provides an additional layer of spectrum security that complements traditional spectrum sensing approaches.

4. Experimental Setup and Methodology

The proposed implementation of the HTSM framework was performed using the Google Collaboratory environment, using Python libraries that include NumPy, SciPy, Scikit-Learn, PyWavelets, Matplotlib, and XGBoost. In addition, the GNU radio companion (GRC) was employed to generate custom RF datasets that represent legitimate and malicious spectrum users.

4.1. Data Set Acquisition and RF Signal Sources

To investigate the behavior of the RF spectrum in diverse communication scenarios, three different datasets were used. The first dataset, RadioML2016.10A, was used to analyze modulation-specific RF characteristics under varying channel conditions. This data set contains eleven modulation schemes, including BPSK, QPSK, 8PSK, PAM4, QAM16, QAM64, GFSK, CPFSK, WBFM, AM-DSB and AM-SSB, across signal-to-noise ratio (SNR) values ranging from –20 to +18 dB. The data set was used primarily for modulation-aware harmonic analysis, spectral entropy extraction, power spectral density estimation, and wavelet decomposition.

The second data set is a subset of the Oracle RF Fingerprinting dataset. It provides real-world RF fingerprints captured from multiple USRP X310 software-defined radio devices. Unlike its synthetic modulation counterparts, this dataset contains device-specific hardware impairments and transmitter fingerprints, making it suitable for trust-aware RF intelligence modeling. The dataset was used to investigate harmonic stability, RF reliability, transmitter-specific spectrum characteristics, and trust signal generation. IQ samples were processed as complex-valued RF signals as:

$$x(n) = I(n) + jQ(n)\,, \tag{1}$$

where $I(n)$ and $Q(n)$ represent the in-phase and quadrature components of the RF signal, respectively.

The third dataset was generated using the GNU radio companion to emulate both legitimate spectrum users and malicious PUE attackers under realistic RF conditions. Modulated communication signals such as BPSK and QPSK were utilized to represent practical wireless transmission behavior.

To simulate realistic wireless environments, channel impairments including Rayleigh fading, Doppler shift, additive white Gaussian noise (AWGN), and phase variations were implemented. The attacker behavior was designed to mimic legitimate transmissions while introducing controlled spectral distortions and power fluctuations, thereby representing sophisticated RF spoofing scenarios. Although a simplified signal model is used for initial validation, the proposed framework is further evaluated using real-world RF datasets, including RadioML 2016.10A and Oracle to ensure robustness under realistic wireless conditions.

4.2. RF Signal Preprocessing

The acquired IQ samples from all datasets were first converted into numerical arrays and preprocessed for further analysis. The RadioML dataset was loaded from the serialized dictionary format and individual modulation samples were extracted. The Oracle RF dataset was loaded as complex-valued IQ samples using a complex128 representation to preserve the information from the transmitter's fingerprint. The GNU radio generated datasets were loaded using 64 complex representations.

To support machine learning-based analysis, the RF signals were partitioned into fixed-size windows.

Let the RF sequence be represented as:

$$X = \{x_1, x_2, x_3, \ldots, x_N\}\,, \tag{2}$$

where N denotes the total number of IQ samples.

The signal was segmented into windows of length $L = 2048$ samples according to the following:

$$W_i = \{x_{iL}, x_{iL+1}, \ldots, x_{(i+1)L-1}\}\,, \tag{3}$$

where W_i denotes the i-th signal segment.

This segmentation process transforms long RF recordings into multiple independent observations, thereby increasing the number of training samples available for machine learning.

4.3. Time-domain RF Signal Analysis

Time-domain visualization was performed to examine the temporal characteristics of legitimate and malicious RF transmissions. The real component of each IQ signal was analyzed to identify waveform stability and oscillatory behavior. Normal users generally produce periodic and stable RF waveforms, whereas malicious users introduce irregular fluctuations and noise-induced distortions. Time-domain analysis provides an initial understanding of the behavior of RF signals before frequency-domain processing.

4.4. Frequency-domain Spectrum Analysis

Frequency domain analysis was conducted using the FFT, which converts the RF signal from the time domain into its spectral representation and reveals harmonic structures associated with legitimate and malicious transmissions. The discrete Fourier transform is defined as:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-\frac{j2\pi kn}{N}}\,, \tag{4}$$

where $x(n)$ is the input RF signal, N is the signal length, and $X(k)$ denotes the spectral representation.

FFT analysis enables visualization of harmonic peaks, spectral spreading, and frequency domain anomalies. The normal user signals generated in GNU Radio exhibited narrow harmonic

peaks, whereas the PUE attack signals produced broader spectral distributions due to the presence of AWGN.

4.5. Power Spectral Density Estimation

To evaluate spectral energy distribution, PSD estimation was performed using the Welch method. PSD provides information on energy concentration across frequency bands and is widely employed in CR spectrum sensing applications. PSD is calculated in the following manner:

$$PSD(f) = \frac{|X(f)|^2}{N} \,, \tag{5}$$

where $X(f)$ denotes the Fourier spectrum and N represents the signal length.

PSD analysis allows for comparison of normal and malicious spectrum occupancy behavior and assists in identifying abnormal RF energy patterns.

4.6. Wavelet-based Harmonic Analysis

Although FFT provides global spectral information, it lacks temporal localization capability. Therefore, discrete wavelet transform (DWT) analysis was employed to investigate localized spectral variations and transient RF disturbances. Daubechies wavelets (db4) were selected because of their effectiveness in capturing RF signal irregularities. The wavelet transform is expressed as:

$$W(a,b) = \int_{-\infty}^{\infty} x(t)\, \psi_{a,b}(t)\, dt \,, \tag{6}$$

where $\psi_{a,b}(t)$ denotes the scaled and shifted mother wavelet.

Wavelet coefficients were extracted from all RF signals and analyzed to distinguish stable harmonic behavior from malicious oscillatory activity.

4.7. Trust-aware Harmonic Spectrum Modeling

Trust is represented as a dynamic spectrum-dependent quantity derived from the characteristics of the RF signal. The trust function is formulated as:

$$T(t) = \alpha\, S(t) + \beta\, C(t) + \gamma\, R(t) + \delta\, O(t) \,, \tag{7}$$

where $S(t)$ denotes spectrum stability, $C(t)$ represents channel consistency, $R(t)$ corresponds to RF reliability, and $O(t)$ indicates spectrum occupancy behavior.

The weighting coefficients satisfy the following:

$$\alpha + \beta + \gamma + \delta = 1 \,, \tag{8}$$

and were empirically selected as:

$$T(t) = 0.3\, S + 0.2\, C + 0.3\, R + 0.2\, O \,. \tag{9}$$

The generated trust sequence forms a trust oscillation curve that reflects the stability of the behavior of the RF spectrum over time. Legitimate users generally exhibit smooth trust variations, whereas malicious users produce significant trust oscillations.

4.8. Feature Extraction and RF Intelligence Metrics

Several RF features were extracted from the processed signals to characterize spectral behavior and detect anomalies associated with the use of the malicious spectrum. The first feature is spectral entropy, which quantifies the uncertainty within the RF spectrum. Spectral entropy is computed as:

$$H = -\sum_{i=1}^{N} p_i \log_2(p_i) \,, \tag{10}$$

where:

$$p_i = \frac{P_i}{\sum\limits_{i=1}^{N} P_i} \,, \tag{11}$$

P_i represents the spectral power of the i-th frequency component. Higher entropy values indicate increased spectral randomness and potentially malicious behavior.

To quantify harmonic instability, a harmonic trust instability index (HTII) is proposed. HTII measures the variation of harmonic spectral energy by relating the variance of the FFT power spectrum to its mean power. This metric provides an indication of the stability of the received RF signal and helps distinguish legitimate users from malicious occupants of the spectrum. HTII is defined as:

$$HTII = \frac{\sigma_P^2}{\mu_P} \,, \tag{12}$$

where the mean FFT spectral power is computed as:

$$\mu_P = \frac{1}{N} \sum_{i=1}^{N} P(i) \,, \tag{13}$$

where σ_P^2 denotes the variance of the FFT power spectrum, μ_P represents the mean FFT spectral power, $P(i)$ denotes the FFT power of the i-th spectral component, and N is the total number of FFT spectral components.

Higher HTII values indicate greater harmonic instability, suggesting irregular RF spectral behavior and an increased likelihood of malicious spectrum occupancy, including PUE attacks.

The second proposed metric is the oscillatory trust divergence (OTD), which measures the deviation between the trust behavior of legitimate spectrum users and malicious PUE attackers. OTD quantifies the cumulative difference in trust values over a sequence of RF observations and is defined as:

$$OTD = \sum_{i=1}^{N} \left| T_{\mathrm{Normal}}(i) - T_{\mathrm{Attack}}(i) \right| \,, \tag{14}$$

where $T_{\mathrm{Normal}}(i)$ denotes the trust value of the legitimate spectrum user at the i-th observation, $T_{\mathrm{Attack}}(i)$ represents the trust value of the suspected PUE attacker, and N is the total number of analyzed RF observations.

Larger OTD values indicate greater divergence from normal spectrum behavior, thereby providing a strong indication of malicious spectrum occupancy and RF spoofing activities.

4.9. Machine Learning-based Intrusion Detection

The extracted RF intelligence features were assembled into feature vectors for intrusion detection. Each signal window generated a feature vector containing spectral entropy and HTII values. The resulting feature matrix is represented as:

$$X = \begin{bmatrix} H_1 & HTII_1 \\ H_2 & HTII_2 \\ \vdots & \vdots \\ H_n & HTII_n \end{bmatrix}, \tag{15}$$

where each row corresponds to an RF signal segment.

Class labels were assigned as:

$$y = \begin{cases} 0, & \text{normal user} \\ 1, & \text{PUE attacker} \end{cases}, \tag{16}$$

The data set was divided into training and testing subsets using a 70:30 ratio. A stratified train-test split was employed to prevent data leakage and all performance metrics were evaluated on previously unseen testing samples. XGBoost was selected as the classifier due to its robustness, computational efficiency, and ability to capture non-linear relationships within RF features.

4.10. Performance Evaluation

The effectiveness of the proposed HTSM framework was evaluated using well-known classification metrics and CR performance indicators:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{17}$$

$$Precision = \frac{TP}{TP + FP}, \tag{18}$$

$$Recall = \frac{TP}{TP + FN}, \tag{19}$$

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}. \tag{20}$$

To provide a more balanced evaluation, the Matthews correlation coefficient (MCC) was defined as:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{21}$$

The consistency was further evaluated using Cohen's kappa coefficient:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{22}$$

where p_o denotes observed agreement and p_e represents expected agreement.

For CR security evaluation, the probability of detection and the probability of false alarm were determined:

$$P_d = \frac{TP}{TP + FN}, \tag{23}$$

$$P_{fa} = \frac{FP}{FP + TN}. \tag{24}$$

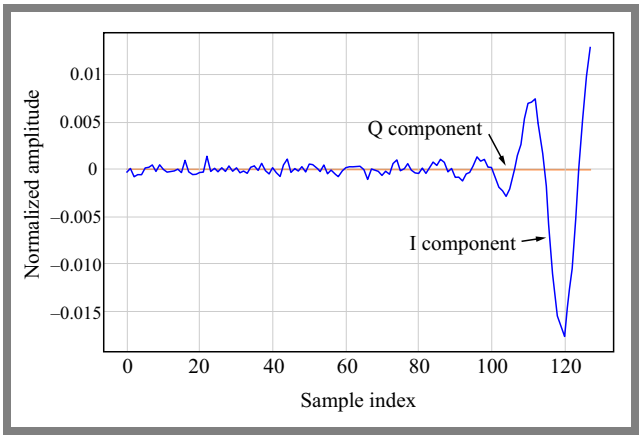


Fig. 2. Time-domain representation of the selected RadioML RF signal.

The generated experimental outputs include time-domain RF waveforms, FFT harmonic spectra, power spectral density plots, wavelet coefficient visualizations, trust oscillation curves, confusion matrices, and machine learning performance metrics.

5. Simulations Results

The evaluation was carried out using multiple heterogeneous RF datasets representing diverse spectrum sensing scenarios. The objective of the evaluation was to investigate harmonic spectrum behavior, trust oscillation characteristics, spectral entropy variations, and the effectiveness of the proposed HTII and OTD metrics.

5.1. RadioML2016.10A

This data set was used as the first benchmark dataset to validate the ability of the proposed framework to analyze the harmonic spectrum. The dataset contains multiple modulation schemes and SNR levels, allowing investigation for RF signal behavior under realistic communication conditions.

Figure 2 illustrates the temporal behavior of the selected RF signal. The waveform exhibits stable amplitude characteristics with limited fluctuations, indicating a communication environment for trust modeling.

The FFT spectrum shown in Fig. 3 shows dominant spectral components concentrated within specific frequency bins. Such a spectral concentration indicates efficient spectrum utilization and reduced harmonic instability.

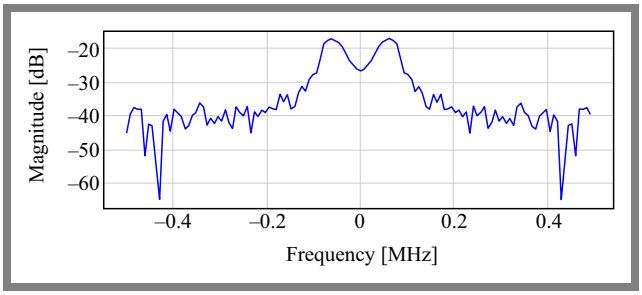


Fig. 3. FFT spectrum of the RadioML.

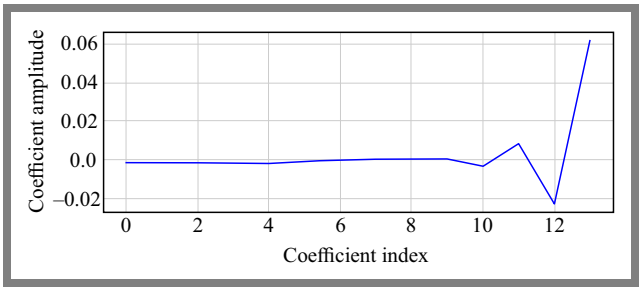


Fig. 4. Wavelet approximation coefficients obtained from RadioML.

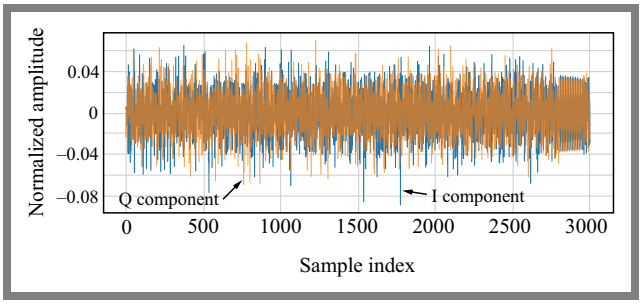


Fig. 5. Oracle RF fingerprint signal.

Figure 4 presents the wavelet approximation coefficients generated using the Daubechies wavelet transform. The smooth coefficient evolution indicates stable low-frequency harmonic structures that are characteristic of legitimate communication signals. The results obtained demonstrate low spectral randomness and stable harmonic characteristics, which form a baseline for the proposed HTSM framework.

5.2. Oracle RF Fingerprinting Dataset

Unlike RadioML, this dataset contains real-world transmitter fingerprints generated from physical wireless devices.

Figure 5 demonstrates the temporal variation of the Oracle RF signal. The observed fluctuations represent hardware-specific non-linearities and manufacturing imperfections that naturally occur in real transmitters.

The FFT presented in Fig. 6 exhibits spectral energy patterns associated with device-specific RF fingerprints. These spectral variations contribute significantly to the estimation of the trust.

Figure 7 illustrates the evolution of trust values over time. Variations in trust indicate changes in spectrum behavior and provide an effective mechanism for identifying abnormal RF activities.

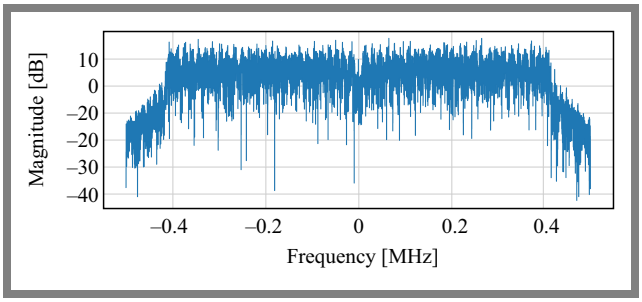


Fig. 6. FFT spectrum of the Oracle RF signal.

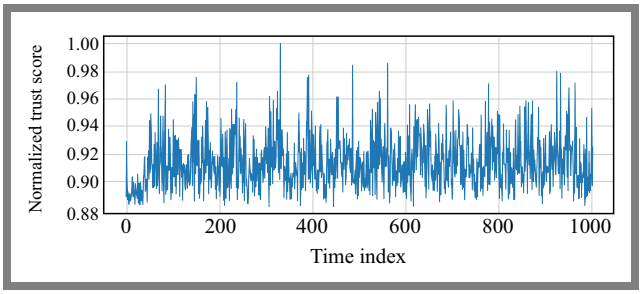


Fig. 7. Trust oscillation curve generated by the proposed HTSM framework.

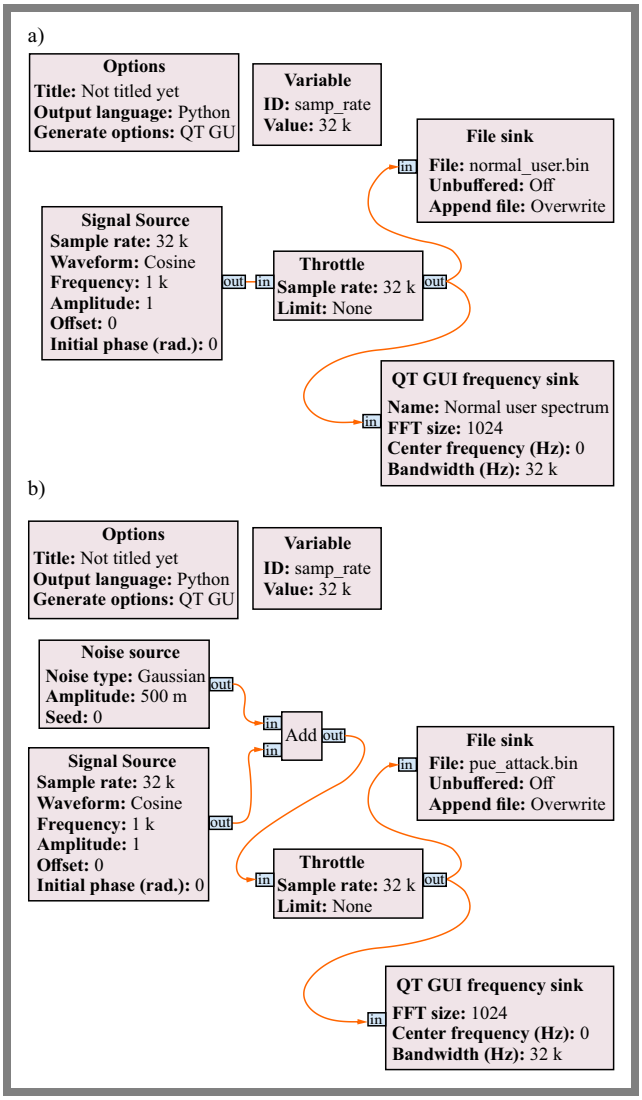


Fig. 8. GNU Radio flowgraph used to generate legitimate user signals a) and for PUE attack b).

5.3. GNU Radio Generated RF Dataset

To validate the proposed framework under controlled attack conditions, a custom RF dataset was generated using GNU Radio Companion. Two binary datasets were created: a legitimate spectrum user represented by `normal_user.bin` and a malicious attacker represented by `pue_attack.bin`. The generated datasets were subsequently analyzed in Google Colab.

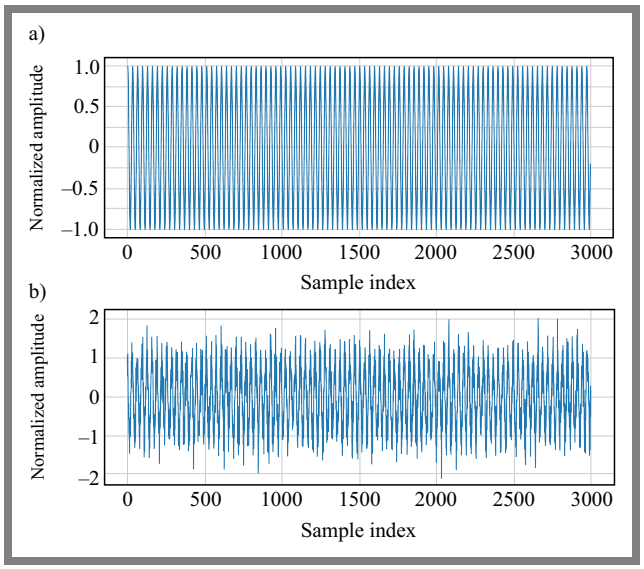


Fig. 9. Legitimate user RF signal a) and PUE attack RF signal b).

Figure 8a shows the signal generation architecture for the legitimate user employing modulated RF transmission schemes, while in Fig. 8b the PUE attack scenario generated by introducing controlled interference, channel impairments, and spectral distortions is presented to emulate malicious spectrum access behavior under realistic conditions. This process emulates an attacker trying to mimic a licensed spectrum occupant.

The legitimate signal shown in Fig. 9a exhibits highly periodic behavior with consistent amplitude characteristics. In contrast, Fig. 9b shows significant fluctuations introduced by malicious interference, resulting in higher spectral uncertainty and trust instability.

The most significant result is presented in Fig. 10. The PSD comparison clearly illustrates the difference between the occupancy of trusted and malicious spectrum. The legitimate user shows concentrated spectral energy, whereas the PUE attacker introduces broadband spectral spreading and increased harmonic distortion. The results confirm that the proposed HTSM framework successfully captures the spectral instability, trust oscillation behavior, and harmonic divergence introduced by malicious spectrum users.

Table 2 presents the classification performance of the proposed framework integrated with the XGBoost classifier on the GNU Radio-generated RF dataset. The dataset consisted of 331 555 feature instances extracted from both legitimate user transmissions and PUE attack signals.

Experimental results demonstrated an overall classification precision of 99.99%, with a precision of 100% and an F1 score of 99.51%, indicating highly reliable discrimination between legitimate and malicious spectrum activities. The probability of detection reached 99.03%, while the probability of false alarm remained at zero, confirming the capability of the proposed framework to identify the occupancy of spoofed spectrum without misclassifying legitimate users. Furthermore, the obtained MCC (0.9951) and Cohen kappa score

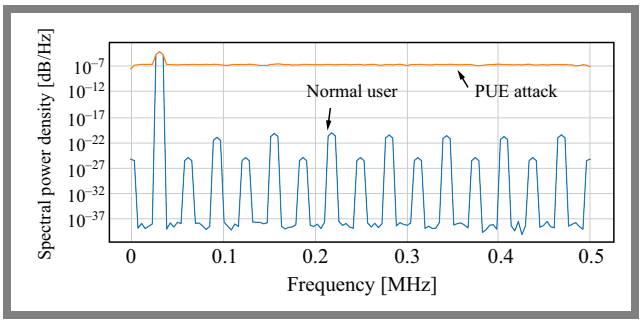


Fig. 10. PSD comparison between legitimate and PUE attack signals.

Tab. 2. Performance evaluation of the proposed framework using the GNU Radio RF dataset.

Performance metric	Value
Accuracy	99.9919%
Precision	100.0000%
Recall	99.0326%
F1-score	99.5139%
Matthews correlation coefficient	0.9951
Cohen kappa score	0.9951
Probability of detection	0.9903
Probability of false alarm	0.0000

(0.9951) indicate an excellent agreement between predicted and actual classes.

These results validate the effectiveness of the proposed HTII and OTD metrics in capturing trust-related spectral variations and demonstrate the suitability of harmonic trust intelligence for next-generation CR intrusion detection systems. It should be noted that the exceptionally high classification performance achieved by the proposed HTSM framework is influenced by the controlled GNU Radio attack generation environment used in this study. The generated PUE attack signals exhibit distinguishable spectral and trust-related characteristics, enabling effective separation between legitimate and malicious spectrum users. Although the results obtained demonstrate the effectiveness of the proposed HTII and OTD metrics, further experimental evaluation of the proposed framework is envisaged using real-world software-defined radio (SDR) platforms such as USRP and ADALM-Pluto devices, under real wireless channel conditions involving fading, interference, noise uncertainty, and mobility effects.

It is worth noting that the GNU Radio dataset incorporates realistic modulation schemes and channel effects. However, further validation using real-world SDR platforms remains an important direction for future work.

6. Conclusions

This paper proposed a harmonic trust spectrum modeling framework for intrusion detection in CRNs. The framework combines harmonic spectrum analysis, trust-aware signal modeling, and machine learning to identify malicious spec-

trum activities, including PUE attacks. The proposed approach was validated using multiple heterogeneous RF data sets under diverse spectrum sensing scenarios. Signal processing techniques such as FFT, PSD, wavelet analysis, and spectral entropy were employed to extract RF spectrum characteristics. Furthermore, two trust-oriented metrics, HTII and OTD, were introduced to quantify spectral trust variations and malicious oscillatory behavior.

The experimental results demonstrated clear distinctions between legitimate and malicious RF activities, with spoofed users exhibiting greater spectral entropy, greater harmonic instability, and increased trust divergence. The integration of these features with an IDS confirmed the effectiveness of the proposed framework for adaptive spectrum security.

Acknowledgments

The authors wish to thank the management, principal, and Head of the Department of Thiagarajar College of Engineering, Madurai, India, for their support and facilities provided for this research.

Data Availability

Data used in this study include publicly available RadioML2016.10A and Oracle RF Fingerprinting datasets, along with a GNU Radio generated RF dataset. The custom dataset is available from the corresponding author upon a reasonable request.

References

[1] A.A. Raji and T.O. Olwal, "Spectrum Sensing in Cognitive Radio Internet of Things: State-of-the-art, Applications, Challenges, and Future Prospects", *Journal of Sensor and Actuator Networks*, vol. 14, art. no. 109, 2025 (https://doi.org/10.3390/jsan14060109).

[2] I.F. Akyildiz, W.-Y. Lee, M.C. Vuran, and S. Mohanty, "A Survey on Spectrum Management in Cognitive Radio Networks", *IEEE Communications Magazine*, vol. 46, pp. 40–48, 2008 (https://doi.org/10.1109/MCOM.2008.4481339).

[3] A. Nasser *et al.*, "Spectrum Sensing for Cognitive Radio: Recent Advances and Future Challenge", *Sensors*, vol. 21, art. no. 2408, 2021 (https://doi.org/10.3390/s21072408).

[4] S. Samala, S. Mishra, and S.S. Singh, "Spectrum Sensing Techniques in Cognitive Radio Technology: A Review Paper", *Journal of Communications*, vol. 15, pp. 577–582, 2020 (https://doi.org/10.12720/jcm.15.7.577-582).

[5] S. Zheng *et al.*, "Spectrum Sensing Based on Deep Learning Classification for Cognitive Radios", *ArXiv*, 2019 (https://arxiv.org/abs/1909.06020).

[6] M.F. Habibi and P. Sukarno, "Analysis of Attack Detection on IoT Using XGBoost Algorithm and SHAP", *2025 International Conference on Information and Communication Technology (ICoICT)*, Bandung, Indonesia, 2025 (https://doi.org/10.1109/ICoICT66265.2025.11193104).

[7] A. Kiran *et al.*, "Intrusion Detection System Using Machine Learning", *2023 International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, 2023 (https://doi.org/10.1109/ICCCI56745.2023.10128363).

[8] M.A. Inamdar and H.V. Kumaraswamy, "Accurate Primary User Emulation Attack (PUEA) Detection in Cognitive Radio Network using KNN and ANN Classifier", *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2020 (https://doi.org/10.1109/ICOEI48184.2020.9143015).

[9] G.K. Gole, S. Morya, and R.K. Nagar, "Spectrum Sensing in Cognitive Radio Using Random Forest with Wavelet and Empirical Mode Decomposition", *International Journal of Environmental Services*, vol. 11, art. no. 24s, 2025.

[10] F. Salahdine, "Compressive Spectrum Sensing for Cognitive Radio Networks", *ArXiv*, 2018 (https://arxiv.org/abs/1802.03674).

[11] F. Salahdine, "Spectrum Sensing Techniques for Cognitive Radio Networks", *ArXiv*, 2017 (https://arxiv.org/abs/1710.02668).

[12] F. Salahdine, H. El Ghazi, N. Kaabouch, and W.F. Fihri, "Matched Filter Detection with Dynamic Threshold for Cognitive Radio Networks", *International Conference on Wireless Networks and Mobile Communications (WINCOM)*, Marrakech, Morocco, 2015 (https://doi.org/10.1109/WINCOM.2015.7381345).

[13] P.D. Sutton, K.E. Nolan, and L.E. Doyle, "Cyclostationary Signatures in Practical Cognitive Radio Applications", *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 13–24, 2008 (https://doi.org/10.1109/JSAC.2008.080103).

S. Jeyaseelan, M.Sc.
https://orcid.org/0009-0009-6556-0491
E-mail: jeyaseelanjeyas@gmail.com
Thiagarajar College of Engineering, Madurai, India
https://www.tce.edu

P.G.S. Velmurugan, Ph.D.
https://orcid.org/0000-0002-3507-9797
E-mail: pgsvels@tce.edu
Thiagarajar College of Engineering, Madurai, India
https://www.tce.edu

G. Prabhakar, Ph.D.
https://orcid.org/0000-0003-4760-7646
E-mail: gpece@tce.edu
Thiagarajar College of Engineering, Madurai, India
https://www.tce.edu

J. Shanthi, Ph.D.
https://orcid.org/0000-0002-7282-7854
E-mail: shanthiece@kamarajengg.edu.in
Kamaraj College of Engineering and Technology, Virudhunagar, India
https://www.kamarajengg.edu.in

Two-way Planar Array Radar with Diamond- and X-shaped Configurations for Sidelobe Suppression

Jafar Ramadhan Mohammed

Ninevah University, Mosul, Iraq

https://doi.org/10.26636/jtit.2026.3.2771

Submitted: 31 Jul. 2026, Accepted: 28 Aug. 2026

Abstract — The undesirable sidelobes in the pattern of a two-way radar antenna array is usually minimized by thinning several elements in the receive or transmit arrays. The thinning process is performed randomly on all array elements by using an optimization algorithm, with the computation time being significant due to a large number of combinations that need to be examined. In this paper, sidelobe suppression in a two-way array pattern is achieved by properly structuring the received array's elements into a diamond- or X-shape, such that its far-field pattern has exact null regions toward the directions of the sidelobe regions of the transmit array pattern. Then, multiplication of the null regions of one pattern with the sidelobe regions of another pattern results in a great reduction in the sidelobe levels of the resultant two-way array pattern. The transmit array structure was fixed as a standard rectangular shape, which has two dominant regions of the sidelobes along the horizontal and vertical lines on the array aperture. To obtain ultra-low sidelobes, the receive array structure should have nulls on these lines, and their sidelobe regions should be located on the diagonals of the array aperture. To meet these alignment conditions, the received array structure was designed as a diamond- or X-shaped structure. It was observed that the proposed configurations offer potential to suppress the sidelobe levels to more than –51 dB and the complexity ratio was lowered to 43%, with both values being much lower than the ones available in the literature.

Keywords — array antenna, sidelobe suppression, two-way radar antenna array

1. Introduction

Array tapers are usually used to reduce the undesirable sidelobe levels in radar antennas. It is a process of designing the weight vector for an antenna array by controlling the phase-only, amplitude-only, or amplitude and phase simultaneously in order to obtain a synthesized pattern with the required sidelobe level (SLL) [1]–[4]. In other approaches, two or more levels of weight vectors were also used to reduce SLL [5]–[7]. The main drawbacks of applying these methods include wider beamwidth, lower directivity, and poorer efficiency [8]–[10].

In the two-way radar antenna array, which is the product of a transmit array pattern and a receive array pattern, the SLL was reduced to more than –26 dB by aligning the first null direction of the receive array pattern in the same direction as the first sidelobe peak of the transmit array, while both arrays have the same number of elements and are excited uniformly [11]. This reduction in SLL can be further improved to more than –31 dB if the number of receive array elements is reduced to 71.4 % of the number of transmit array elements [12]. This reduction was achieved by thinning the edge elements of the receive array only.

Many other works used iterative-based stochastic optimization algorithms to randomly thin the elements of transmit or receive arrays [13]–[19]. However, all these iterative-based optimization algorithms need a huge number of iterations to examine all the possible combinations of the active and inactive elements in transmit and receive arrays. Thus, they are characterized as slow convergence algorithms, especially when applied to larger planar antenna arrays.

In [19], the receiving array in the two-way radar system was designed and analyzed with only two elements, while in [20], a plus-shaped coarray approach was used to simplify the design configurations of a 3D medical ultrasound system.

In this paper, several new designs structures for the receive array in a two-way radar system with novel X- and diamond-shaped structures are proposed. The unique shape, enabling their null regions to be placed at the exact positions with respect to the sidelobe regions of the transmit array patterns, is the decisive feature of the solution. The null regions of these receive array patterns are along the horizontal and vertical axes of the array aperture, while the sidelobe regions of the transmit array are along the diagonals. Thus, ultralow sidelobes can be obtained with the proposed structures.

Moreover, the number of the active elements in these proposed receive structures is much lower than that of the transmit array elements. In other words, all elements in the two-way radar system are transmitters, while only a small subset of elements arranged in a diamond or X shape are receivers. Thus, a great reduction in the complexity of the overall array was also achieved by adopting the proposed approach.

2. Problem Formulation

Consider two rectangular arrays. The first one consists of $T_x = N_t \times M_t$ elements and has the form of a uniform

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

transmit array, while the other consists of $R_x = N_r \times M_r$ elements and is a uniform receive array. The overall two-way array pattern is the product between the far-field patterns of the transmit and receive arrays and may be described as follows:

$$AF_{\text{Two-Way}}(u,v) = AF_T(u,v) \times AF_R(u,v)\,, \tag{1}$$

where:

$$AF_T(u,v) = \frac{1}{N_t \times M_t} \sum_{n=1}^{N_t} a_n e^{-j(n-1)(kd_{xt}u)} \times \sum_{m=1}^{M_t} a_m e^{-j(m-1)(kd_{yt}v)}\,, \tag{2}$$

$$AF_R(u,v) = \frac{1}{N_r \times M_r} \sum_{n=1}^{N_r} a_n e^{-j(n-1)(kd_{xr}u)} \times \sum_{m=1}^{M_r} a_m e^{-j(m-1)(kd_{yr}v)}\,, \tag{3}$$

where $u = \sin\theta\cos\phi$, $v = \sin\theta\sin\phi$, θ is the elevation angle, ϕ is the azimuth angle, $k = \frac{2\pi}{\lambda}$, λ is the wave length, and d_{xt}, d_{yt}, d_{xr}, and d_{yr} are the inter-element spacing along the x-axis and y-axis in the transmit and receive arrays, respectively.

Since both transmit and receive arrays are uniform, then a_n and a_m have unit amplitudes for all transmit and receive elements.

For a rectangular structure with $T_x = R_x$, the two-way array structure and the corresponding array patterns for both transmit and receive arrays are shown in Fig. 1. From the array contour pattern, there are only two obvious sidelobe regions, marked using green and yellow, along the horizontal and vertical axes of the array aperture, while the null regions, marked blue, are positioned symmetrically in four quadrants. Here, since the sidelobe regions of the transmit and receive arrays are exactly aligned with each other, the SLL of the resultant two-way planar array is equal to the peak sidelobe level of the uniform transmit pattern times the peak sidelobe level of the uniform receive pattern. In decibels, this is equivalent to –13.2 × 2 = –26.4 dB.

A better reduction in SLL can be obtained when $T_x > R_x$ by a factor of 1.4, i.e., the number of elements in the receive array equals 66.67% of the number of elements in the transmit array [11]. This helps change the direction of the first null of one array to match the direction of the first sidelobe of the other array. However, this alignment is only limited to the first null and first sidelobe peak; therefore, only the first sidelobe level in the resultant two-way array pattern was suppressed [12].

3. Proposed Two-way Array Structure

In the proposed planar array structures, the number of active elements in the receive array can be greatly reduced while achieving an ultra-low SLL in the resulting two-way array

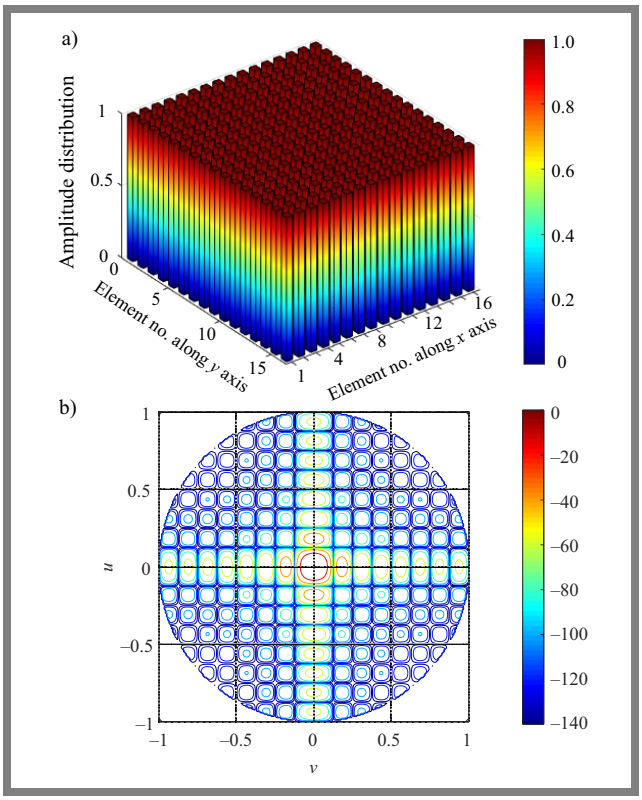


Fig. 1. Conventional two-way planar antenna array structure (16 × 16) with all elements transmuting and receiving simultaneously, and the relevant contour pattern.

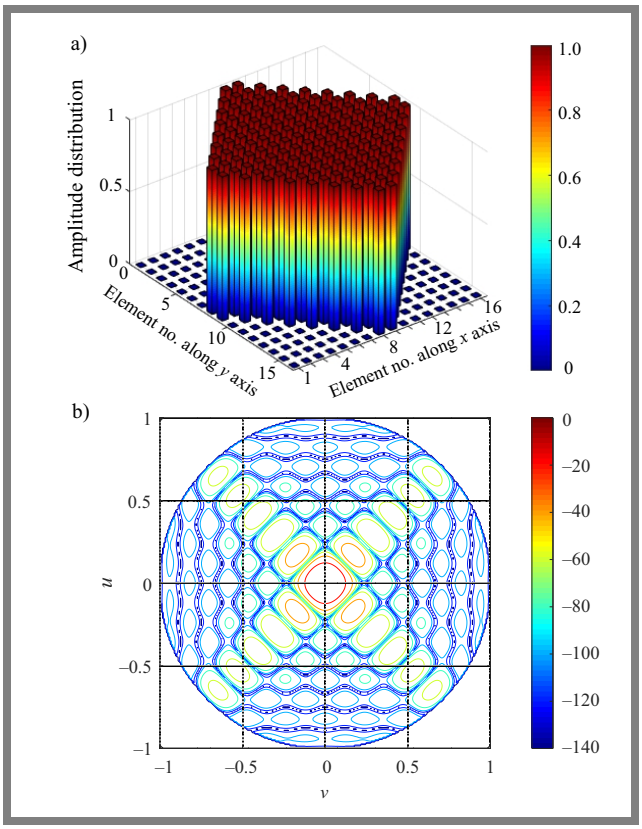


Fig. 2. Proposed diamond-shaped receive array structure and its corresponding contour pattern.

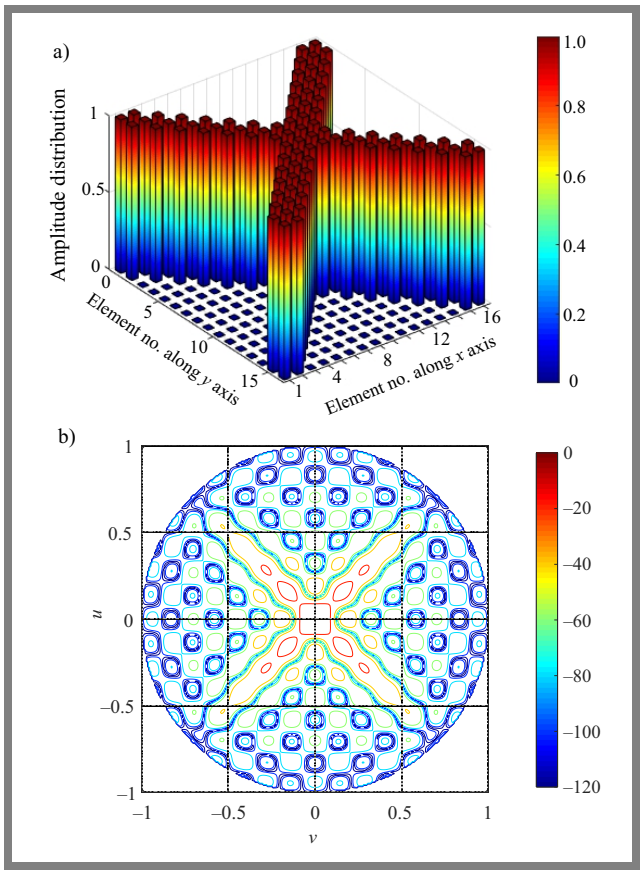


Fig. 3. Proposed X-shaped receive antenna array structure and its corresponding contour pattern.

pattern. The best SLL reduction may be obtained by aligning the null regions of the receive array pattern in the same positions as the sidelobe regions of the transmit array pattern. To ensure such an alignment between the null regions and the sidelobe regions, the structure of the receive array should be changed.

Below, several structures are proposed for a receive array whose null regions are in perfect alignment with the sidelobe regions of the transmit array pattern. The diamond-shaped structure is one of the best solutions that meets these alignment requirements (see Fig. 2). X-shaped structure is another solution that perfectly meets the alignment requirement, as shown in Fig. 3.

The number of active elements in these proposed receive array structures is much lower than that of the transmit array. Noticeably, all elements in the two-way radar system are transmits, while only a small subset of active X- or diamond-shaped elements are receiving. Thus, a significant reduction in the overall complexity of the array is achieved.

Additionally, a plus-shaped structure was considered as well. The receive array with this structure is shown in Fig. 4. The size and the shape of the array aperture in the case of the plus-shaped structure remains fixed, just as for the standard 16 × 16 square array shown in Fig. 1. The sidelobe regions of the transmit and receive arrays do not change and remain aligned with each other, just as shown in Fig. 1. Therefore,

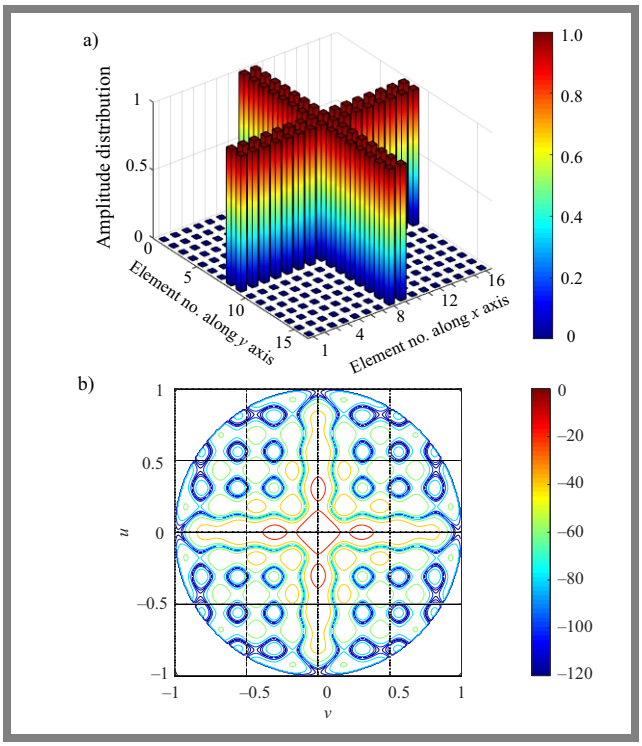


Fig. 4. Proposed plus-shaped receive array and its corresponding contour pattern.

the SLL of the resultant two-way plus-shaped planar array is reduced to only –26.4 dB.

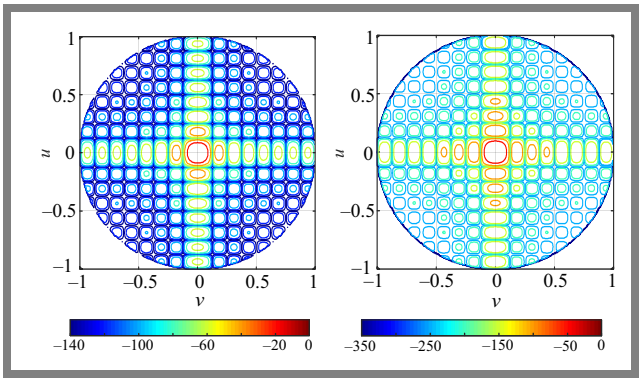


Fig. 5. Array patterns for the method from [11]: a) transmit and receive, and b) resultant two-way.

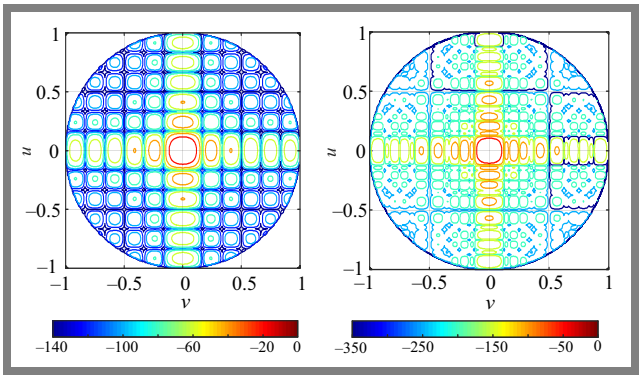


Fig. 6. Array patterns for the method proposed in [12]: a) receive and b) resultant two-way.

Tab. 1. Performance measures of the tested methods.

Method	Receive configuration	T_x	R_x	Complexity ratio [%]	SLL [dB]	Directivity [dB]
[11]	Square	256	256	100	–26.46	23.69
[12]	Square with thinned edges	256	142	55.47	–30.44	22.66
Proposed 1	Diamond	256	112	43.75	–51.00	23.16
Proposed 2	X-shaped	256	88	34.38	–47.47	20.93
Proposed 3	Plus-shaped	256	60	23.44	–26.46	23.69

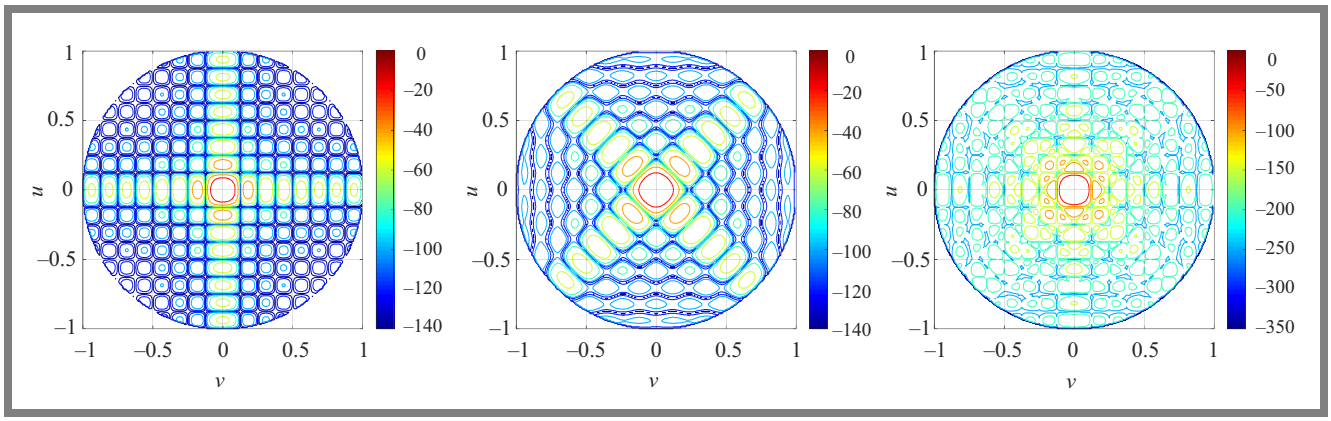


Fig. 7. Array patterns for the diamond-shaped structure: a) transmit, b) receive, and c) resultant two-way.

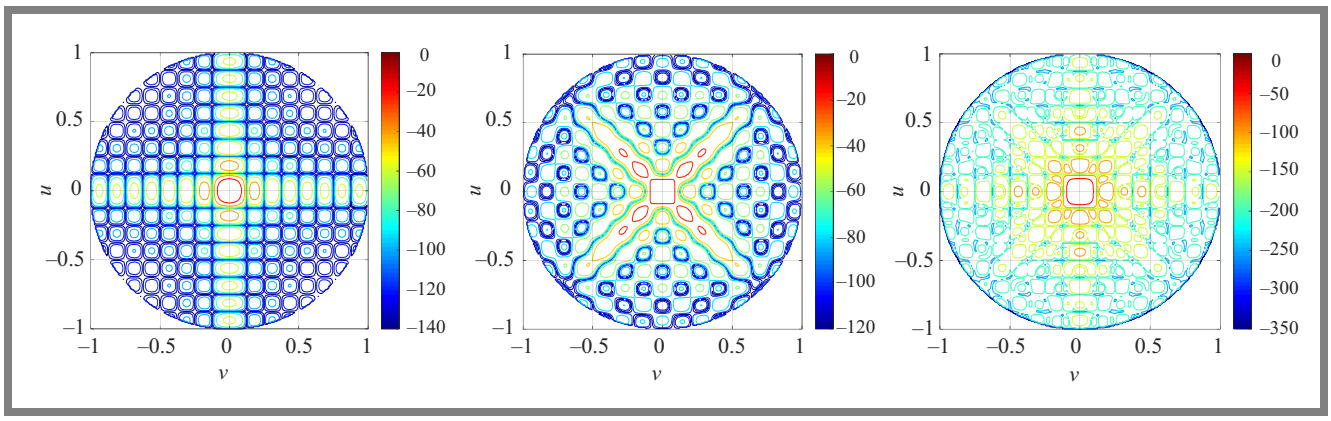


Fig. 8. Array patterns for the X-shaped antenna structure: a) transmit, b) receive, and c) resultant two-way.

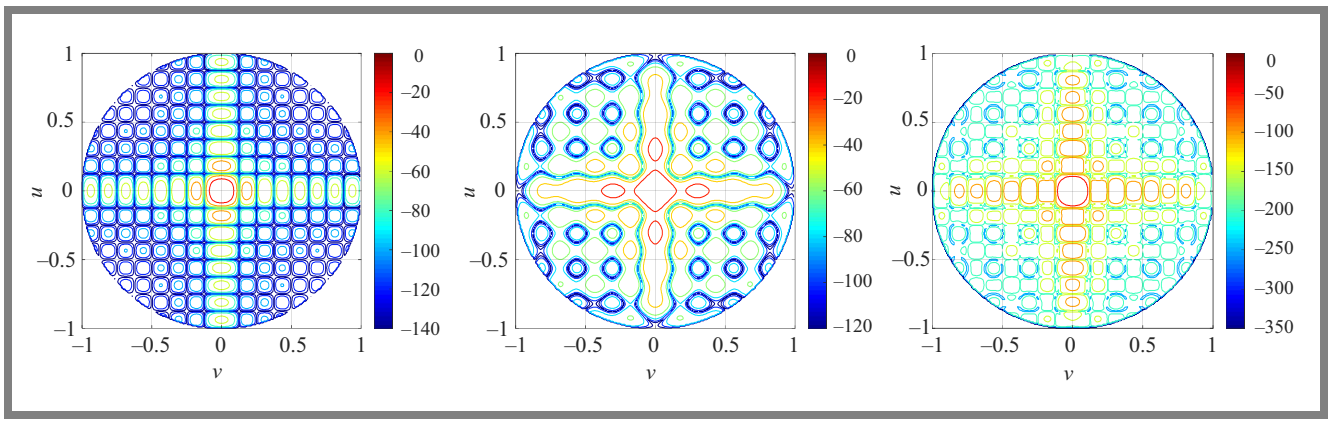


Fig. 9. Array patterns for the plus-shaped structure: a) transmit, b) receive, and c) resultant two-way.

3.1. Simulation Results

The effectiveness of the proposed two-way planar antenna array structures is demonstrated under several scenarios. The conventional two-way array method from [11] and the method described in [12] are also covered, for comparison.

In all the considered methods, all the elements in the transmit and receive arrays are uniformly excited with unit amplitudes and zero phases. The number of elements in the transmit arrays was $T_x = 16 \times 16$, and $d_{xt} = d_{yt} = 0.5\lambda$, whereas the number of elements in the receive arrays R_x was as given. For the comparison to be fair, the array patterns of the tested methods depicted in the following figures are normalized.

Two-way array directivity is the first performance measure of the tested methods. It is defined by:

$$Directivity|_{\text{dB}} = 10\ \log_{10} \frac{4\pi\, U_{Two-Way}(u,v)}{P_{rad}}\ , \qquad (4)$$

where

$$U_{Two-Way}(u,v) = \left[AF_{Two-Way}(u,v)\right]^2$$

is the radiation intensity of the resultant two-way array pattern given by Eq. (1) and P_{rad} is its radiated power.

Peak sidelobe level is the second performance measure. System complexity, defined as the ratio of the number of active elements in the receive and transmit arrays, is another important measure:

$$Complexity = \frac{R_x}{T_x} \times 100\ \ [\%]\ . \qquad (5)$$

The performance measures of the three methods [11], [12] and the proposed structures are shown in Tab. 1 and Figs. 5 – 9. According to this table, the SLLs of the X-shaped and diamond structures are the lowest. Directivity of the diamond-shaped structure is higher than that of the X structure.

However, the three complexity ratios of the proposed structures are lower than the same parameter of the conventional ones. The array patterns of the proposed X- and diamond-shaped structures shown in Figs. 7 and 8 indicate that the positions of the null regions in the receive array patterns are precisely aligned with the positions of the sidelobe regions in the transmit array patterns; thus, most of the sidelobes in the resultant two-way patterns were effectively suppressed.

The variations of SLL and complexity ratio of the diamond structure versus the diamond aperture size (i.e., the number of active receive elements R_x) is analyzed in Tab. 2. The deepest reduction in SLL occurs when $R_x = 112$ elements, while at this aperture size, the complexity ratio becomes 43.75%.

Tab. 2. Variations in SLL and complexity ratio for the diamond structure versus R_x.

Receive configuration	T_x	R_x	Complexity ratio [%]	SLL [dB]	Directivity [dB]
Diamond 1	256	144	56.25	–43.42	23.69
Diamond 2	256	112	43.75	–51.0	23.16
Diamond 3	256	84	32.81	–47.89	22.66
Diamond 4	256	60	23.44	–45.34	22.20
Diamond 5	256	40	15.63	–43.26	21.75
Diamond 6	256	24	9.38	–41.63	21.33

4. Conclusions

The simulation results confirm the effectiveness of the proposed two-way array structures with diamond- and X-shaped receive antenna arrays in obtaining ultralow sidelobe levels, reducing system complexity, and retaining undistorted directivities. The obtained values of the suppressed SLL, equaling –51.0 dB, and the complexity ratio of 43.75% (for the proposed diamond-shaped structure) are much better than those of conventional two-way methods. These values could be further improved when using smaller diamond sizes. Moreover, the proposed two-way planar structures may be enhanced further by including clustered configurations rather than elemental level counterparts.

References

[1] J.R. Mohammed, "An Optimum Side Lobe Reduction Method with Weight Perturbation", *Journal of Computational Electronics*, vol. 18, pp. 705–711, 2019 (https://doi.org/10.1007/s10825-019-01323-5).

[2] C.A. Balanis, *Antenna Theory: Analysis and Design*, 4th ed., John Wiley & Sons, Hoboken, USA, 1104 p., 2016 (ISBN: 9781118642061).

[3] J.R. Mohammed, "Unconventional Method for Antenna Array Synthesizing Based on Ascending Clustered Rings", *Progress in Electromagnetics Research Letters*, vol. 117, pp. 69–73, 2024 (https://doi.org/10.2528/PIERL23122201).

[4] J.R. Mohammed, "Minimization of Grating Lobes in Large Arrays Using Clustered Amplitude Tapers", *Progress In Electromagnetics Research C*, vol. 120, page 93–103, 2022 (https://doi.org/10.2528/PIERC22031706).

[5] J.N. Sahalos, "Design of Shared Aperture Radar Arrays with Low Sidelobe Level of the Two-way Array Factor", *IEEE Transactions on Antennas and Propagation*, vol. 68, pp. 5415–5420, 2020 (https://doi.org/10.1109/TAP.2020.2981735).

[6] R. Rajender *et al.*, "Design of Linear and Planar Arrays with Low Sidelobe Levels and High Directivity Using Two-way Array Factor", *IEEE Transactions on Antennas and Propagation*, vol. 72, pp. 4150–4160, 2024 (https://doi.org/10.1109/TAP.2024.3375966).

[7] T. Samaras and J. Sahalos, "A Systematic Study of low SLL Two-way Pattern in Shared Aperture Radar Arrays", *Progress In Electromagnetics Research C*, vol. 128, pp. 169–182, 2023 (https://doi.org/10.2528/PIERC22102003).

[8] K.H. Sayidmarie and J.R. Mohammed, "Performance of a Wide Angle and Wide Band Nulling Method for Phased Arrays", *Progress In Electromagnetics Research M*, vol. 33, pp. 239–249, 2013 (https://doi.org/10.2528/PIERM13100603).

[9] J.R. Mohammed, "Rectangular Grid Antennas with Various Boundary Square-rings Array", *Progress in Electromagnetics Research Letters*, vol. 96, pp. 27–36, 2021 (https://doi.org/10.2528/PIERL20112402).

[10] J.R. Mohammed and A.J. Abdulqader, "Array Pattern Restoration under Defective Elements", *Progress in Electromagnetics Research C*, vol. 123, pp. 17–26, 2022 (https://doi.org/10.2528/PIERC22061101).

[11] R.L. Haupt, "Lowering the Sidelobe Level of a Two-way Array Factor for an Array with Uniform Transmit and Uniform Receive Arrays", *IEEE Transactions on Antennas and Propagation*, vol. 67, pp. 4253–4256, 2019 (https://doi.org/10.1109/TAP.2019.2905932).

[12] R.L. Haupt, "Optimizing the Sidelobe Level of a Two-way Antenna Array Pattern by Thinning the Receive Aperture", *2018 International Conference on Radar (RADAR)*, Brisbane, Australia, 2018 (https://doi.org/10.1109/RADAR.2018.8557293).

[13] P. Rocca *et al.*, "Design of Modular Radar Array Antenna for Two-way Pattern Sidelobe Optimization", *2022 IEEE Radar Conference (RadarConf22)*, New York, USA, 2022 (https://doi.org/10.1109/RadarConf2248738.2022.9764163).

[14] T.N.F. Kaifas and J.N. Sahalos, "Low SLL Control of Two-way Pattern in Shared Circular Planar Radar Arrays", *IEEE Transactions on Antennas and Propagation*, vol. 72, pp. 2915–2920, 2024 (https://doi.org/10.1109/TAP.2024.3352270).

[15] R.L. Haupt, *Antenna Arrays: A Computational Approach*, John Wiley & Sons, Hoboken, USA, 534 p., 2010 (https://doi.org/10.1002/9780470937464).

[16] J.R. Mohammed, R. Hamdan, and A.J. Abdulkadeer, "Linear and Planar Array Pattern Nulling via Compressed Sensing", *Journal of Telecommunications and Information Technology*, vol. 85, pp. 50–55, 2021 (https://doi.org/10.26636/jtit.2021.152921).

[17] G.M Battaglia, A.F. Morabito, R. Palmeri, and T. Isernia, "Minimal-size 2-D Sources for Mask-constrained Synthesis of Shaped Patterns", *IEEE Open Journal of Antennas and Propagation*, vol. 7, pp. 128–139, 2025 (https://doi.org/10.1109/OJAP.2025.3637546).

[18] J.R. Mohammed, "A New Technique for Obtaining Wide-angular Nulling in the Sum and Difference Patterns of Monopulse Antenna", *IEEE Antennas and Wireless Propagation Letters*, vol. 11, pp. 1245–1248, 2012 (https://doi.org/10.1109/LAWP.2012.2224086).

[19] J.R. Mohammed, "Two-way Array Radar with Two-elements Receive Array and Improved Performance", *Progress in Electromagnetics Research C*, vol. 166, pp. 195–200, 2026 (https://doi.org/10.2528/PIERC25112601).

[20] M. Karaman, I.O. Wygant, Ö. Oralkan, and B.T. Khuri-Yakub, "Minimally Redundant 2-D Array Designs for 3-D Medical Ultrasound Imaging", *IEEE Transactions on Medical Imaging*, vol. 28, pp. 1051–1061, 2009 (https://doi.org/10.1109/TMI.2008.2010936).

Jafar Ramadhan Mohammed, Professor
College of Electronics Engineering
https://orcid.org/0000-0002-8278-6013
E-mail: jafar.mohammed@uoninevah.edu.iq
Ninevah University, Mosul, Iraq
https://uoninevah.edu.iq

An Adaptive Threshold Tracking Area Update and Selective Probabilistic Paging Scheme for Signaling Cost Reduction in 5G NR Networks

Kalpesh Popat[1], Divyakant Meva[1], and Hardik K. Molia[2]

[1]*Marwadi University, Rajkot, India,*
[2]*Government Engineering College, Bhavnagar, India*

https://doi.org/10.26636/jtit.2026.3.2701

Submitted: 3 Jul. 2026, Accepted: 26 Aug. 2026

Abstract — Tracking area update (TAU) and paging are two competing signaling-cost characteristics of 5G NR networks: a smaller registration area reduces paging cost but increases TAU signaling and vice versa. The majority of existing schemes optimize one parameter while fixing the other and apply a single network-wide parameterization to all users regardless of their heterogeneous call and mobility patterns. This paper proposes ATAU-SP, a scheme that manages both operations jointly, on a per user equipment (UE) basis: each UE maintains an online estimate of its call-to-mobility ratio (CMR) and triggers a TAU only after an adaptive movement threshold derived by minimizing a per UE cost function, with a dwell-time guard that suppresses boundary ping-pong updates. The network builds a residence-probability distribution from a learned Markov mobility model and pages cells in descending probability rounds up to a delay bound. We derive a closed-form analytical cost model and validate it as a provable upper bound against a custom discrete-event simulator. The results show that a well-tuned TAL scheme is the lowest-cost baseline over most of the moderate CMRs, ATAU-SP-ML attains the lowest cost at high CMR, and plain ATAU-SP is best only at very low CMR. Nonetheless, ATAU-SP variants dominate the classical static, distance-, movement-, and time-based schemes throughout, and both ATAU-SP and TAL require materially less machinery per UE state and network side than a full TAL list distribution subsystem.

Keywords — 5G NR, mobility management, paging, signaling cost, tracking area update

1. Introduction

In an idle or inactive radio state, the user equipment (UE) is not in direct communication with the network, so the network maintains only an approximate, area-level record of the UE's location. Two control plane procedures keep this record usable: the first, tracking area update (TAU), is initiated by the UE when it moves into a tracking area that is not included in its list of currently registered tracking area identifiers (TAI). The other, i.e., paging, is initiated by the network when downlink traffic arrives for an idle UE, as the access and mobility management function (AMF) broadcasts a paging message across the cells of the UE's registration area until the UE responds. Together, these procedures constitute location management.

If a UE registers its location frequently, the network knows its position precisely, and paging is cheap, but TAU signaling is high. If registration is infrequent, TAU signaling falls, but the network must page a larger area to locate the UE. Because both procedures consume scarce control plane and radio resources and, on the UE side, battery energy, minimizing their combined cost is a long-standing objective that has only gained in importance as network density increases and massive numbers of low-power devices join the network [1]–[3].

Despite previous work, three limitations still exist. First, most schemes treat the two operations separately: they fix the registration rule and then optimize paging, or vice versa. Second, schemes typically apply a single network-wide parameterization to all UEs, even though real populations are highly heterogeneous: a stationary metering device, a pedestrian, and a vehicle differ by orders of magnitude in both mobility rate and call frequency, so a parameter that is optimal on average is suboptimal for most individual UEs. Surveys of 5G mobility management identify adaptive, per-UE, prediction-assisted location management as an open direction [1].

A third limitation is that per-UE adaptive schemes have rarely been benchmarked against per-UE tracking area list (TAL) schemes or against learning-based next-cell predictors under a common cost model. Both are per UE, so the comparison is not obvious.

This paper addresses these limitations with ATAU-SP – a joint, adaptive, per-UE location management scheme, and provides an empirical comparison against the two baseline families identified above.

The scientific contribution is as follows:

- Jointly adaptive scheme. ATAU-SP couples an adaptive registration with a probability-ordered paging rule, so that the registration threshold and the paging search are optimized together against a single signaling cost objective.

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

- Per-UE adaptive movement threshold with a formally characterized estimator. Each UE estimates its own call-to-mobility ratio (CMR) using an exponentially weighted moving average (EWMA) and sets its TAU threshold by minimizing a closed-form per-UE cost function. This paper adds a convergence analysis of the EWMA estimator – a bias half-life and a steady-state noise floor, both in closed form – and quantifies how estimation error propagates to the threshold through the cube-root relationship.
- Selective probabilistic paging under a delay bound, with an explicit grouping rule and an entropy-based benchmark. Upon the arrival of a call, AMF builds a residence probability distribution over candidate cells and pages them in descending probability rounds, subject to a hard limit D_{max} on the number of rounds. This article explicitly states the probability-balanced grouping rule and compares the realized mean delay against an information-theoretic lower bound.
- Two additional baselines and comparison. We implement and simulate a dynamic, frequency-based TAL scheme and a lightweight second-order Markov predictor ATAU-SP-ML, alongside the original four baselines, and report where each of them wins, rather than only where ATAU-SP wins.
- Analytical model, simulation validation, and new sensitivity analysis. We develop a closed-form per-UE cost expression, validate it as a provable upper bound against a custom discrete event simulator (DES), and add a cost-ratio sensitivity sweep, a dwell-time-guard ablation, a Markov matrix learning convergence study, and a UE-energy proxy grounded in a published TAU power measurement.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 presents a model of the system and the formulation of the problem. Section 4 describes the ATAU-SP algorithm, including TAL and ATAU-SP-ML. Section 5 derives the analytical cost model. Section 6 describes the simulation setup. Section 7 reports and discusses the results. Section 8 concludes the article.

2. Related Works

2.1. Registration Strategies (Location Update)

Registration strategies fall into those of the static and dynamic category. Static strategies group cells into fixed location or tracking areas. A UE registers only when it crosses an area boundary, and the network pages the entire area. The area size is a network-wide design parameter, which makes static schemes simple but not capable of adapting to the behavior of individual UEs. Dynamic strategies make the registration decision specific to the UE. The three canonical dynamic rules are distance-based, movement-based, and time-based.

Research shows that each rule has an optimal threshold that depends on the ratio between call arrival rate and mobility rate, and that the distance-based rule is generally the most efficient of the three but the hardest to implement, as it requires the UE to compute distance in cells [1], [4], [5].

More recent work targets the 5G tracking-area structure directly. Dynamic TAL schemes assign each UE a personalized list of tracking areas to balance TAU and paging overhead, and clustering-based methods group cells into tracking areas according to observed intercell mobility, so that frequently traversed boundaries fall inside a tracking area rather than across one [6], [7].

2.2. Paging Strategies

Paging strategies range from blanket parallel paging, which pages all cells of the registration area simultaneously to minimize delay at maximum signaling cost, to sequential paging, which pages cells one group at a time to minimize signaling cost at the expense of delay. Selective and probabilistic paging methods order the search by the estimated probability that the UE resides in each cell.

Searching in descending order of probability minimizes the expected number of cells paged. Delay-bounded variants partition the area into a fixed number of paging rounds, so that the worst-case delay is controlled while still exploiting the probability ordering [5], [8], [9].

2.3. Mobility Prediction for Location Management

Mobility prediction is the foundation of the residence probabilities that selective paging and adaptive registration require. Markov models are widely used because they rely only on information already available to the network, such as handover and cell-residence history. Their low complexity is a contributing factor as well.

Weighted and user-classified Markov models improve precision by accounting for the heterogeneity of individual movement patterns rather than relying on a single aggregate model [10]. Classical movement-based location-update and selective-paging schemes have also been studied as joint location-management mechanisms [11].

Machine learning approaches, including hidden Markov models and neural sequence models, can capture a longer-range structure at a higher computational cost [5], [12], [13]. Recent work adopted a first-order Markov model as a lightweight predictor and was cited, but did not implement a learning-based alternative.

2.4. Optimization-based and Learning-based Mobility Management

A parallel line of research formulates cell tracking area or reporting planning as an NP-hard combinatorial optimization problem and solves it with metaheuristics such as genetic algorithms, particle swarm optimization, and related methods [14], [15], or with learning-based controllers that adapt the configuration to traffic [16], [17].

Comprehensive surveys of 5G and beyond mobility management catalog these directions and highlight adaptive, prediction-assisted, and energy-aware location management as priorities, particularly for ultra-dense deployments and large device populations [1], [3], [19], [20].

Machine learning for handover and mobility optimization reaches similar conclusions about the value of adaptation [12], [13]. Independently of location management specifically, large-scale studies of real human mobility using call-detail record (CDR) and GPS data report that individuals repeatedly return to a small number of frequently visited locations and that a large share of individual trajectories is theoretically predictable [21], [22].

3. System Model and Problem Formulation

ATAU-SP differs from the above in that it adapts both the registration decision and the paging search jointly and per-UE, against a single cost objective, while remaining lightweight enough for network-side implementation.

The classical static, distance-based, movement-based, and time-based schemes, as well as blanket paging, all arise as special cases of ATAU-SP, which lets us argue dominance rather than incomparable difference for that family of baselines.

Against the two per-UE baselines – a dynamic personalized TAL and a higher-order Markov predictor – ATAU-SP does not dominate uniformly; their relative performance depends on the operating regime.

3.1. Network and Mobility Model

Service area is modeled as a set of N hexagonal cells $C = \{c_1, \ldots, c_N\}$, each served by one gNB, with a fixed inter-site distance (Fig. 1). Cells are grouped into tracking areas (Fig. 2).

The number of cells per tracking area s is a configurable parameter. We focus on idle mode UEs, whose location is known to the network at tracking area granularity. The UE population is U.

For a UE u, cell residence times are independent and identically distributed with a mean $1/\lambda_m(u)$, so that the cell crossing (mobility) rate is $\lambda_m(u)$. The next sessions (calls) arrive as a Poisson process with a rate of $\lambda_c(u)$ [18]. The per-UE call-to-mobility ratio is defined as:

$$CMR(u) = \frac{\lambda_c(u)}{\lambda_m(u)} \,. \tag{1}$$

A low CMR denotes a highly mobile UE that is rarely called (favoring infrequent registration), and a high CMR denotes a frequently called, slowly moving UE (favoring frequent registration).

The movement of UE at the cell level is described by a first-order Markov chain with a transition matrix

$$P = [p_{ij}] \,,$$

where P_{ij} is the probability that an UE in cell c_i next enters cell c_j. The matrix is estimated from observed intercell transitions, optionally weighted by UE class to reflect heterogeneous movement patterns [10].

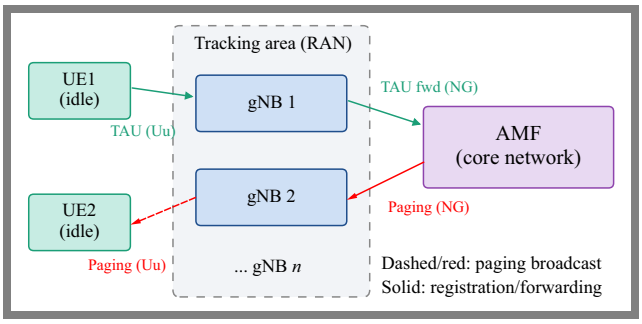


Fig. 1. 5G NR location management model. An idle UE registers via TAU over the Uu and NG interfaces. On a call, the AMF triggers paging across the registration area.

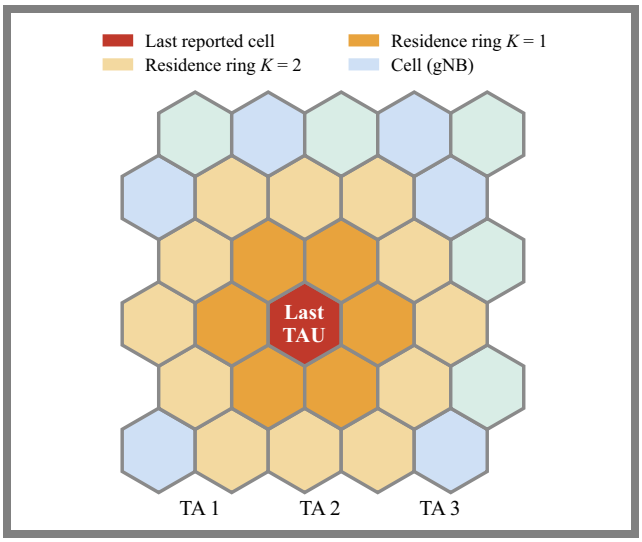


Fig. 2. Hexagonal cells grouped into tracking areas. After up to K crossings since the last TAU, the UE lies within a residence neighborhood of K rings around the last reported cell.

3.2. Cost Model

Let C_u be the signaling cost of one TAU and C_p the signaling cost of paging one cell. Over a long observation window, the per-UE signaling cost rate is the sum of a registration term and a paging term:

$$\Phi(u) = C_u \cdot R(u) + C_p \cdot \lambda_c(u) \cdot M(u) \,, \tag{2}$$

where $R(u)$ is the UE's TAU rate (updates per unit time), and $M(u)$ is the expected number of cells paged per incoming call. The first term is the registration cost, and the second is the paging cost, since each call triggers a search over $M(u)$ cells on average. Both $R(u)$ and $M(u)$ depend on the registration and paging rules.

The network-wide objective is to minimize total cost subject to a paging delay constraint:

$$\begin{gathered} \text{minimize} \sum_{u \in U} \Phi(u) \text{ subject to} \\ D(u) \leqslant D_{max} \text{ and } P_{succ}(u) \geqslant \rho \end{gathered} \,, \tag{3}$$

where $D(u)$ is the delay in paging in rounds, D_{max} is the maximum allowed number of paging rounds, $P_{succ}(u)$ is the probability that the UE is found within the delay bound, and ρ is target success probability.

Because UEs are independent in this model, the network objective decomposes into independent per-UE problems, which is what makes per-UE adaptation both tractable and beneficial – a property shared by the TAL and ATAU-SP-ML baselines introduced below, both of which are also per-UE schemes.

4. Proposed ATAU-SP Algorithm and Baseline Extensions

ATAU-SP consists of two coupled mechanisms: an adaptive-threshold registration rule executed at the UE, and a selective probabilistic paging rule executed at the network. Both are driven by quantities that are already observable in a 5G NR system, namely cell-crossing events, call arrivals, and inter-cell transition statistics.

This section also specifies the dwell-time-guard hysteresis rule, the probability-balanced grouping procedure and introduces the two new baselines: a dynamic TAL scheme and a second-order Markov predictor ATAU-SP-ML.

4.1. Adaptive-threshold Tracking-area Update

Each UE maintains an estimate of its CMR using an EWMA of its observed cell-crossing and call-arrival intervals:

$$\dot{m}(u) = \alpha \cdot \tau_m + (1-\alpha) \cdot \dot{m}(u) \,,$$

$$\dot{c}(u) = \alpha \cdot \tau_c + (1-\alpha) \cdot \dot{c}(u) \,,$$

where τ_m and τ_c are the most recently observed intercrossing and inter-call intervals, giving the following:

$$\hat{C}MR(u) = \frac{\dot{m}(u)}{\dot{c}(u)} \,.$$

The UE counts cell crossings since its last TAU. When the count reaches an adaptive threshold $K(u)$, it issues a TAU and resets the count. A dwell-time guard suppresses a TAU if the crossing is an immediate return to the previously visited cell within one crossing interval. If the newly entered cell is equal to the cell the UE occupied two crossings ago, the crossing is not counted toward k and no TAU is triggered.

The threshold $K(u)$ is set by minimizing the per-UE cost and is recomputed whenever $\hat{C}MR(u)$ changes by more than a fixed tolerance 15%. Setting $K(u)$ to a fixed constant recovers the classical movement-based scheme, and keying the count to the hop distance from the last registered cell recovers the distance-based scheme. ATAU-SP makes the threshold state-dependent and per-UE instead.

4.2. Selective Probabilistic Paging

When a call arrives for an idle UE, the AMF computes a residence probability vector over the candidate cells in the UE registration area (Fig. 3). Starting from the last reported cell and the crossings elapsed since the last TAU, the network propagates the learned Markov chain to obtain, for each candidate cell c_j, a residence probability π_j. Candidate cells are sorted in descending order π_j and partitioned into D_{max} paging rounds at most.

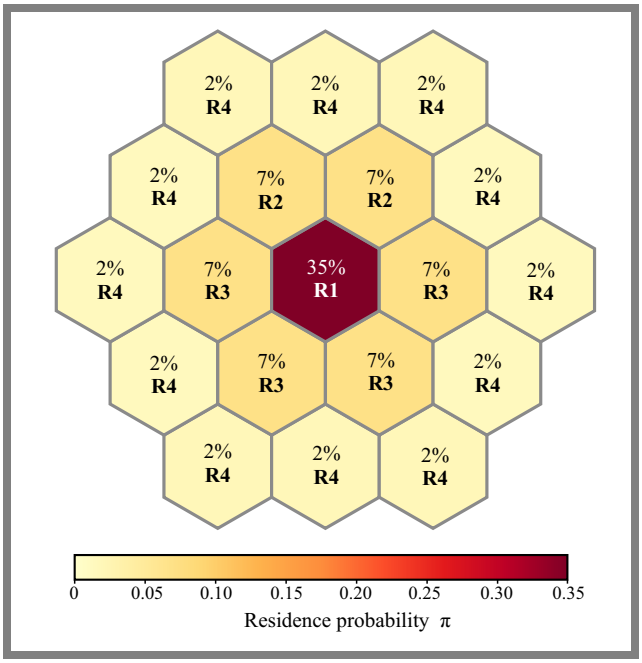


Fig. 3. Selective probabilistic paging. Candidate cells are shaded by residence probability and paged in descending probability rounds $(\mathrm{R1}, \mathrm{R2}, \ldots)$ up to the delay bound D_{max}, reducing the expected number of cells paged relative to blanket paging.

The AMF pages the first round. If the UE does not respond, it pages the second round and so on, stopping as soon as the UE responds or the bound D_{max} is reached. Setting $D_{max} = 1$ with all cells in a single round recovers blanket parallel paging, allowing one cell per round recovers maximum likelihood sequential paging.

The structure of the algorithm is as follows (Fig. 4).

Procedure 1 – Adaptive registration (runs on each UE).

1) Initialize crossing counter $k \leftarrow 0$ and CMR estimate from a warm-up window.
2) In each cell crossing, update the mobility rate EWMA. For setting the dwell-time guard, if the newly entered cell equals the cell occupied two crossings ago, i.e., the UE has bounced back to where it just came from, do not increment k and return without further action.
3) Otherwise increment k. If $k \geqslant K(u)$, issue a TAU carrying the current cell, then reset $k \leftarrow 0$.
4) On each incoming call, update the call rate EWMA, recompute $\hat{C}MR(u)$ and, if it has changed by more than 15% since the last recomputation, recompute $K(u)$.

Procedure 2 – Network-side selective paging (runs at the AMF).

5) On a call for idle UE u, retrieve the last reported cell and the crossing count since the last TAU, propagate the Markov chain that many steps to obtain residence probabilities π over the candidate cell set $A = ring_K$(last reported cell). The set of cells that can be reached in most K hops

$$|A| = f(K).$$

6) Probability-balanced grouping. Sort candidate cells by descending π. Partition the sorted list into D_{max} consec-

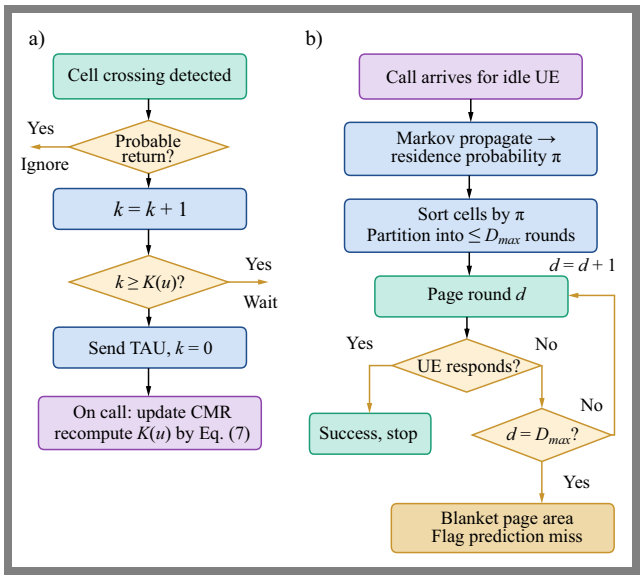


Fig. 4. ATAU-SP control flow scheme: a) adaptive registration procedure with dwell-time guarding and online threshold recomputation, and b) selective probabilistic paging procedure on the network side with descending probability rounds and a blanket paging fallback.

utive groups $g_1, \ldots, g_{D_{max}}$ of as equal a size as possible, i.e.

$$|g_j| = \lceil |A|/D_{max} \rceil$$

for the first $|A|$ mod D_{max} groups and $\lfloor |A|/D_{max} \rfloor$ thereafter, so that

$$\sum_j |g_j| = |A|.$$

This is the instantiation of the "probability-balanced grouping" by Eq. (5), where g_j denotes the number of cells assigned to paging round j according to this rule.

7) For round $d = 1$ to D_{max}, page the cells of round d, i.e.,

$$\sum_{j \leqslant d} g_j$$

reduces to exactly g_d being paged at step d, having not paged earlier rounds cells again. If the UE responds, record success and stop. The cumulative number of cells grouped by round d is

$$\sum_{j \leqslant d} |g_j|,$$

which is the $M(K)$ term in Eq. (5).

8) If there is no response after D_{max} rounds, fall back to blanket paging of the full registration area and flag a prediction miss for model update.

Procedure 1 costs $O(1)$ per event, apart from threshold recomputation, which is a one-dimensional minimization with a closed-form solution. Procedure 2 costs $O(|A| \log |A|)$ per call for sorting $|A|$ candidate cells of the registration area, plus the Markov propagation, which is sparse because the transition matrix has non-zero entries only between adjacent cells. Both are lightweight enough for online use.

4.3. TAL Scheme

We implement a dynamic TAL scheme in the style of [7]. Each UE is assigned a personalized list of cells (its TAL) rather than a network-fixed tracking area. The network maintains an online, per-UE visit frequency table over the cells. Whenever the UE crosses into a cell outside its current TAL, it issues a TAU. In this scheme, the TAU also implicitly requests a list refresh, so TAU and list update signaling are counted together at the same C_u cost, giving TAL no advantage over ATAU-SP's registration cost accounting.

Once the visit frequency table for the local neighborhood has accumulated enough observations, the network rebuilds the TAL as the UE's most frequently visited cells within an expanded radius of the new anchor cell, up to a size budget matched to $f(K_{radius})$ for a radius parameter tuned relative to the same movement-threshold scale. Before the table has warmed up, the list returns to a simple geometric neighborhood. Paging, when a call arrives, pages the UE's entire current TAL at once.

Note that our synthetic mobility models are memoryless or weakly correlated random walks without repeated visitation structure, for example, with no modeled "home" and "work" anchor cells, so a frequency-based list under these traces converges to a geometric neighborhood of the recently visited cells.

4.4. ATAU-SP-ML, a Second-order Markov Predictor

We implement a second-order (bigram) Markov model that conditions the next predicted cell based on the current cell and the cell visited immediately before it, learned online from the same crossing events already available to the first-order model, with a one-cell-of-history requirement that can be piggybacked on existing TAU signaling.

ATAU-SP equipped with this predictor is referred to as ATAU-SP-ML. It uses the same adaptive threshold registration and round-based paging as ATAU-SP, differing only in how the residence-probability vector π is computed.

5. Analytical Cost Model

We derive the per-UE cost as a function of the registration threshold K and then obtain the optimal threshold. Between two successive calls, the expected number of cell crossings is equal to the ratio of the mobility rate to the call rate

$$\frac{\lambda_m}{\lambda_c} = \frac{1}{CMR}.$$

With a movement threshold K, the UE issues one TAU every K crossings, so the expected number of TAUs per call cycle is

$$\frac{1}{CMR \cdot K},$$

and the registration cost per call cycle is (Fig. 5):

$$\Phi_{reg}(K) = C_u \cdot \frac{1}{CMR \cdot K}. \quad (4)$$

On the paging side, after up to K crossings since the last TAU, the UE lies within a neighborhood of the last reported

cell whose size grows with K. For a hexagonal layout, the number of cells within K rings is

$$f(K) = 3K(K+1) + 1 \, .$$

Under blanket paging, the paging cost per call is $C_p \cdot f(K)$. Under ATAU-SP selective paging, ordering cells by residence probability and applying the probability-balanced grouping of Sec. 4.2 reduces the expected number of cells paged to:

$$M(K) = \sum_{d=1}^{D_{max}} \left(\sum_{j \leqslant D} |g_j| \right) \cdot q_d \, , \tag{5}$$

where $|g_j|$ is the number of cells assigned to paging round j by the equal-count partition of procedure 2, step 6, and q_d is the probability that the UE is first located in round d.

The cells are searched in descending probability order, $M(K) \leqslant f(K)$, with equality only when the residence distribution is uniform. By the rearrangement inequality, no other ordering of the candidate cells yields a smaller expected search, so descending probability paging is optimal for a given partition into rounds.

Combining the cost two terms, the per call cycle is:

$$\Phi(K) = \frac{C_u}{CMR \cdot K} + C_p \cdot M(K) \, . \tag{6}$$

The first term decreases in K, while the second increases in K, so a unique minimizer exists. $M(K)$ is approximated by a quadratic in K, since $f(K)$ is quadratic, and selective paging scales it by a near-constant factor over the operating range. Setting the derivative to zero gives an optimal threshold of the form:

$$K^* = \kappa \cdot \left(\frac{C_u}{C_p \cdot CMR} \right)^{\frac{1}{3}} . \tag{7}$$

Equation (7) is the analytical basis for the adaptive rule: K^* increases as CMR decreases, so highly mobile, rarely called UEs register less often, while frequently called UEs register more often (Fig. 6). ATAU-SP substitutes the online estimate $CMR(u)$ in Eq. (7), clip K^* to the integer range admissible for the size of the registration area and applies it on the per-UE basis. The constant κ is set by a numerical sweep of the fixed K cost simulated at the $CMR = 1$, $C_u : C_p = 10 : 1$, which locates the true empirical minimum at $K = 2$ and gives $\kappa = 0.93$.

5.1. Convergence of the CMR Estimator

Both $\dot{m}(u)$ and $\dot{c}(u)$ are EWMA estimates of the mean of an exponentially distributed inter-event time. Two standard properties of the exponential smoothing estimator characterize its behavior.

The first is bias decay from a cold start. If the estimator is initialized at a value that differs from the true rate by a factor of β, the relative bias after n observed events decays as $(1-\alpha)^n$. Therefore, the bias half-life is:

$$t_{1/2} = \frac{\ln 2}{-\ln(1-\alpha)} \, ,$$

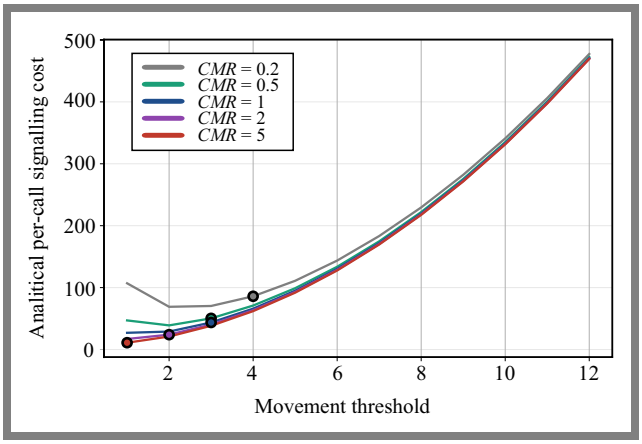


Fig. 5. Analytical cost per call of signaling cost $\Phi(K)$ vs. movement threshold K for several CMRs, evaluated from Eq. (6) with $f(K) = 3K(K+1)+1$ and $C_u : C_p = 20$. The cost is convex in K; circles mark the optimal threshold K^*, using $\kappa = 0.93$.

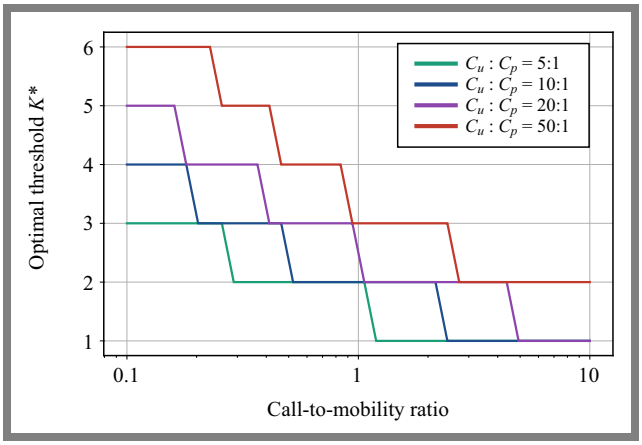


Fig. 6. Analytical optimal threshold K^* vs. call-to-mobility ratio for several cost ratios, from Eqs. (6)–(7).

events and the number of events needed to shed 90% of the initial bias is:

$$n_{90\%} = \frac{\ln 0.1}{\ln(1-\alpha)} \, .$$

For $\alpha = 0.1$, these give $t_{1/2} \approx 6.6$ events and $n_{90\%} \approx 21.9$ events – i.e., a warm-up window of roughly 20 crossing/call events per UE is sufficient for the cold start bias to become negligible.

The second is steady-state noise floor. Because the smoothed quantity is itself a noisy per-event sample rather than a converging average, EWMA retains an irreducible steady-state variance even after the bias has decayed. For an exponential inter-event time $CV = 1$, the steady state relative error of the EWMA estimate is approximately:

$$\sqrt{\frac{\alpha}{2-\alpha}} \, ,$$

combining the mobility- and call-rate estimates via the ratio $\hat{C}MR = \dot{m}/\dot{c}$ roughly doubles this value in the worst case. We verified both properties with a synthetic Monte Carlo test (300 trials, 5× mismatched cold start): the measured steady-state error was 18%, 26–30%, and 37–42% for α = 0.05,

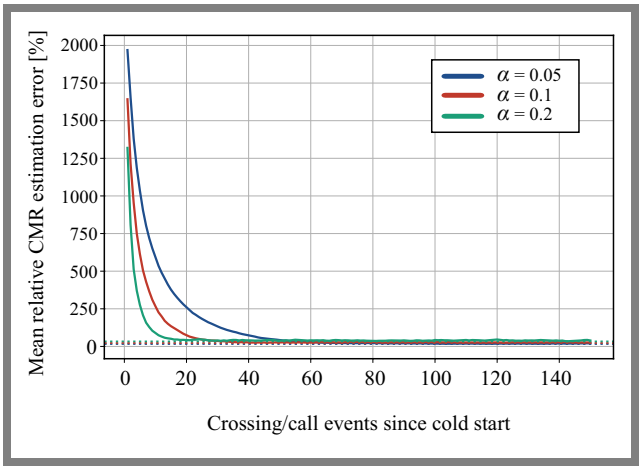


Fig. 7. EWMA-CMR convergence: mean relative estimation error vs. number of observed events, for three decay rates α, starting from a 5× mismatched cold start. The dotted lines mark the closed-form steady-state noise floor.

0.1, and 0.2 respectively, consistent with the closed-form floor.

Figure 7 plots the measured convergence curves. This smaller α gives a lower noise floor (more precise threshold selection) but is slower to track a genuinely time-varying CMR, e.g., a UE that changes its mobility pattern, while larger α tracks changes faster at the cost of noisier, more frequent threshold recomputation. We use α = 0.1 throughout, balancing a sub-30-event warm-up against a moderate 26 – 30% steady-state floor.

Consider the error propagation to the threshold from Eq. (7), a first-order propagation gives relative error in $K^* \approx 1/3\times$ relative error in $\hat{C}MR$. Therefore, the cube root relationship is self-damping: even a 26 – 30% steady-state CMR estimation error at α = 0.1 propagates to an error of approximately 9 – 10% only in K^*, and because K^* is subsequently clipped to a small integer range, the realized threshold values are often completely unaffected by estimation noise.

5.2. Paging Delay Bound and Theoretic Comparison

For a residence distribution π over $|A| = f(K)$ candidate cells with Shannon entropy $H(\pi)$, a standard argument for grouped sequential search states that no grouping into D_{max} rounds can achieve the expected number of rounds below:

$$\frac{H(\pi)}{\log_2(|A|/D_{max})},$$

where $|A|/D_{max}$ is the average branching factor of a round. This is an approximate bound in our setting, but it gives a basic floor to compare against. Using the simulator, the mean entropy of the residence distribution at the moment of call $CMR = 1, \sim 18$ cells/TA is $H(\pi) \approx 3.45$ bits on a mean candidate-set size of 14.2 cells. With $D_{max} = 3$, this gives a lower bound of approximately 1.54 rounds. The simulated mean delay at $D_{max} = 3$ is 1.76 rounds, about 14% above this bound, indicating that the simple equal-count grouping used by ATAU-SP operates reasonably close to, but measurably short of, the entropy-optimal grouping, which would require unequal-size rounds rather than the equal-count rule currently used.

5.3. Quadratic Approximation Error and Its Propagation

Fixing K and running the simulator gives the true simulated $M(K)$ at each integer K. Fitting the least squares-optimal scalar γ in the surrogate

$$M_{quad}(K) = \gamma \times f(K)$$

against these measured points shows that the point-wise fit is loose. At $CMR = 1$ the relative error is

$$\frac{|M(K) - M - quad(K)|}{M(K)}$$

and ranges from 3.5% near $K = 5$ to 76.6% at $K = 1$, averaging 39.8% across $K = 1 - 6$. This is consistent with, and helps explain, the 46.5% average envelope gap reported in Sec. 7.9. The quadratic surrogate is a genuinely loose local approximation to the true selective-paging cost, not merely a small correction.

Whether this loose pointwise fit changes the resulting threshold choice is a separate question, and the answer is operating-point dependent. Comparison of the argmin of Eq. (6) against the argmin of the quadratic surrogate at $CMR = 1$ and $CMR = 5$, the two curves select the same integer $K^* = 2$ and $K^* = 1$ respectively. Despite the large pointwise $M(K)$ error, the resulting threshold and cost are unaffected – the registration term's $1/K$ divergence at small K dominates the argmin decision at these points, and the fit error is not large enough to shift it.

At $CMR = 0.5$, however, the quadratic surrogate does select a different threshold from the true optimum ($K^*_{quad} = 3$ vs. $K^*_{exact} = 2$), and using the surrogate threshold in practice would incur a measurable, if modest, cost penalty of 2.3% relative to the true optimum. Therefore, the quadratic approximation's threshold selection error is real but small where it occurs. We observed a value of 2.3% at most across the three CMR values tested. It does not occur at every operating point and the closed form remains useful for the qualitative CMR dependence of Eq. (7).

6. Simulation Setup

ATAU-SP and six baselines were implemented in a custom discrete-event simulator written in Python (vectorized with NumPy). The simulator models location management at the signaling level: cells are nodes of a hexagonal adjacency graph, the UE's serving cell is updated by a direction-persistent random walk on that graph calibrated per mobility model, while cell crossings and call arrivals are generated as independent Poisson processes with per-UE rates $\lambda_m(u), \lambda_c(u)$ [23].

The logic runs the registration procedure of Sec. 4.2; a central function plays the role of the AMF, holding each UE's last reported cell, learning the Markov transition matrix (or matrices, for ATAU-SP-ML) from observed crossings, exe-

Tab. 1. Simulation parameters.

Parameter	Value
Simulator	Custom Python (NumPy) discrete-event, signaling-level
Topology	37 hexagonal cells, one gNB per cell
Inter-site distance	500 m
Cells per tracking area	5, 7, 12, 18, 37 (operating point ≈ 18)
Number of UEs	100 (idle mode), heterogeneous CMR (log-normal, $\sigma = 0.6$)
Mobility models	Random waypoint; Gauss-Markov (heading persistence $\kappa = 0.75$); RPGM (5 groups of 20)
UE speed	1 – 30 m/s (default 10)
Call arrivals	Poisson, per-UE rate set to the UE's target CMR × mobility rate
CMR	0.1 – 10 (nominal, population mean)
Cost ratio $C_u : C_p$	10:1 nominal; swept 5:1, 10:1, 20:1, 50:1
Paging delay bound D_{max}	1 – 5 (default 3)
Markov model order	ATAU-SP and bigram ATAU-SP-ML
TAL list size budget	$f(radius)$, $radius = \max(1, \lfloor K/2 \rfloor)$, frequency-ranked
K^* calibration constant κ	0.93
Simulation time / step	1200 s / event-driven (Poisson inter-event sampling), 200 s warm-up
Replications	15 (main sweeps), 8 – 12 (secondary sweeps), 95% CIs reported

cuting the paging procedure, and tallying signaling statistics. TAU and paging are modeled as counted control events.

Correlated mobility is generated at the cell-graph level rather than by simulating continuous positions. At each mobility event, the UE's heading persists with probability $\kappa = 0$ for random waypoint, $\kappa = 0.75$ for Gauss-Markov, and a two-level group/member persistence model for RPGM (group-anchor heading persistence 0.85, member-to-anchor attraction 0.55), so that the sequences of visited cells among group members exhibit correlation.

This is a coarser abstraction than simulating continuous UE positions and is explicitly stated as a modeling simplification. Such an approach preserves the qualitative ordering of the prediction difficulty across the three models (RWP is the hardest to predict, RPGM the easiest).

All schemes are implemented in the same module and evaluated under identical topologies, mobility traces, and call traces, so that differences are attributable to the schemes alone:

- Static registration area + blanket paging – the 3GPP-style reference: register at the boundary crossing, page the whole area at once.

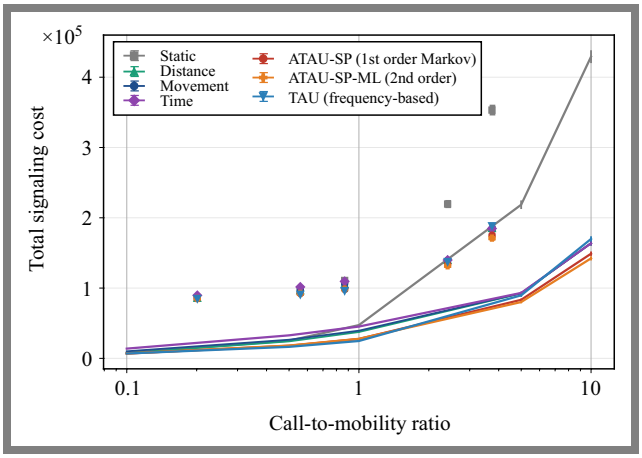


Fig. 8. Total signaling cost vs. CMR for all seven schemes (random waypoint, ≈ 18 cells/TA), including the TAL and ATAU-SP-ML baselines.

- Distance-based + blanket paging – register after a threshold hop distance from the last registered cell.
- Movement-based + blanket paging – register after a fixed number of cell crossings.
- Time-based + blanket paging – register after a fixed time interval.
- ATAU-SP (proposed) – adaptive per-UE threshold with first-order Markov selective probabilistic paging.
- ATAU-SP-ML – identical registration and round-based paging as ATAU-SP, using the second-order (bigram) Markov predictor.
- TAL – dynamic, frequency-based personalized tracking area list with blanket paging over the current list.

The performance metrics are as follows:

- TAU rate (updates per UE per unit time) and total registration signaling.
- Mean cells paged per call (paging signaling).
- Total signaling cost under a sweep of the cost ratio $C_u : C_p$.
- Paging success probability within the delay bound and mean paging delay (rounds), compared against an entropy-based lower bound.
- Convergence of the CMR estimator.
- Dwell-time-guard contribution, isolated by ablation.
- UE energy proxy (idle-mode monitoring plus TAU transmissions), grounded in a published per-TAU power measurement.

7. Results and Discussion

All results below are produced using parameters listed in Tab. 1.

7.1. Total Signaling Cost versus CMR

Comparison against the four classical baselines (static, distance, movement, and time) shows that ATAU-SP outperforms all four at every CMR tested. At $CMR = 1$, ATAU-SP

Tab. 2. Total signaling cost (normalized units) vs CMR, random waypoint, ≈ 18 cells/TA (mean over 15 seeds). The lowest values per row are shown in bold.

CMR	Static	Distance	Movement	Time	ATAU-SP	ATAU-SP-ML	TAL
0.1	7812	7492	10 034	13 865	**6808**	6947	6909
0.5	25 137	24 181	26 492	32 739	17 930	18 222	**16 382**
1.0	47 329	37 503	39 361	45 252	27 578	27 815	**24 708**
5.0	218 594	92 313	92 669	93 105	83 298	**80 128**	89 683
10.0	429 221	163 830	163 904	163 746	148 915	**142 275**	169 898

Tab. 3. TAU rate and mean cells paged per call at $CMR = 1$ (random waypoint, ≈ 18 cells/TA). The lowest total cost is shown in bold.

Scheme	TAU rate [UE/s]	Cells paged / call	Total cost
Static	0.00378	17.58	44 608
Distance	0.00572	13.11	36 181
Movement	0.00786	12.98	38 009
Time	0.01391	13.03	44 174
ATAU-SP	0.00786	8.07	26 608
ATAU-SP-ML	0.00786	8.19	26 875
TAL	0.00849	6.68	**24 016**

reduces signalling cost by 26.5% compared with the best classical dynamic baseline (distance-based) and by 41.8% compared with static registration, corresponding to an average reduction of approximately 34.1% across these two reference schemes (Fig. 8)..

TAL is the lowest cost scheme from $CMR = 0.5$ through $CMR = 1$ (8.6 – 10.4% below plain ATAU-SP), plain ATAU-SP is the cheapest only at the lowest CMR tested (0.1), where paging is rarely invoked and TAL's list refresh overhead is not yet amortized, and ATAU-SP-ML becomes the cheapest at high CMR (5 and 10), by 3.8% and 4.5% over plain ATAU-SP and by 10.7% and 16.3% over TAL, respectively.

7.2. Registration and Paging Overhead Decomposition

The decomposition shows that the per-UE schemes outperform the classical baselines and why TAL edges out ATAU-SP at this operating point (Tab. 3). Static registers rarely (once per 264 UE-seconds) the largest area (17.6 cells/call).

The classical dynamic schemes cut paging to roughly 13 cells/call at the cost of more frequent registration. ATAU-SP and ATAU-SP-ML have the same TAU rate as movement-based approaches, since both use the same adaptive-threshold registration rule, while cutting paging even further, to 8.1 to 8.2 cells/call, via probability-ordered rounds.

TAL achieves the lowest paging cost of all: 6.68 cells/call, since it bounds the search to a small personalized list rather than a K-ring neighborhood at a marginally higher TAU rate (0.00849 vs. 0.00786/UE/s) than the movement threshold schemes, and the net effect is the lowest total cost specified in this table.

This demonstrates that a well-tuned personalized list can outperform threshold-based adaptive registration under conditions where paging cost dominates.

7.3. Effect of the Registration-area Size

The crossover between static and the per-UE schemes now falls between 7 and 12 cells/TA, below which static is cheaper and above which every per-UE scheme (ATAU-SP, ATAU-SP-ML, and TAL alike) is cheaper (Fig. 9).

At the dense deployment operating point of 18 cells/TA, TAL is 46.8% below static and ATAU-SP is 40.9% below static. At 37 cells/TA, the gaps widen to 72.7% and 69.4% respectively.

Since 5G deployments trend toward larger tracking-area configurations to limit TAU signaling from massive device populations, the large-area regime where per-UE schemes win is the operationally relevant one, and this conclusion is robust to the choice of per-UE scheme.

7.4. Effect of Mobility Model

Qualitative ranking is consistent across random waypoint, Gauss-Markov, and RPGM approaches: TAL is the lowest in all three, closely followed by ATAU-SP and ATAU-SP-ML, with classical dynamic baselines being 30 – 34% higher and static being 42 – 46% higher (Fig. 10).

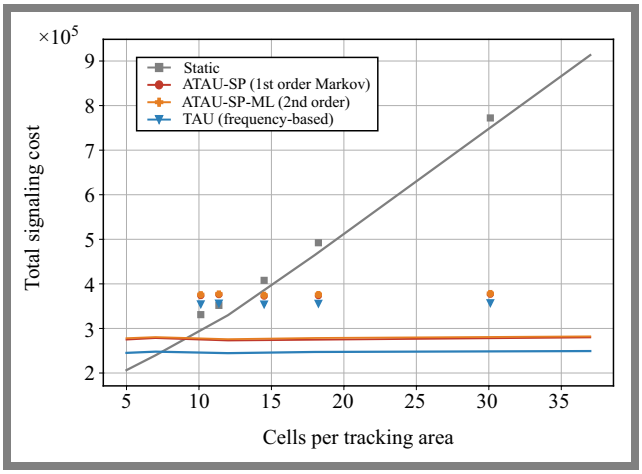


Fig. 9. Total cost vs. size of the tracking area for static, ATAU-SP, ATAU-SP-ML, and TAL (random waypoint, $CMR = 1$).

Tab. 4. Total signaling cost versus cost ratio for $CMR = 1$, random waypoint, ≈ 18 cells/TA. The lowest values per row are shown in bold.

$C_u : C_p$	Static	Distance	Movement	Time	ATAU-SP	ATAU-SP-ML	TAL
5:1	44 488	26 791	27 438	28 983	21 546	21 286	**20 383**
10:1	46 456	37 516	39 520	45 599	27 409	27 697	**24 575**
20:1	50 488	47 612	52 526	68 312	35 789	36 344	**32 760**
50:1	59 324	63 129	76 479	123 718	51 171	**51 963**	51 279

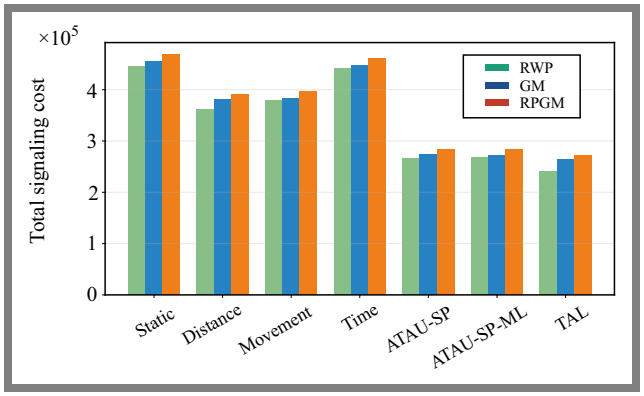


Fig. 10. Total cost under the three mobility models for all seven schemes ($CMR = 1, \approx 18$ cells/TA).

The second-order predictor's advantage is primarily a high-CMR effect in this simulator, not primarily a mobility-correlation effect.

7.5. Effect of UE Speed

As UE speed increases from 1 to 30 m/s, cell-crossing frequency and, hence, signaling cost increase for every scheme, but the relative ranking is stable throughout the swept range (Fig. 11). TAL is the lowest at every speed (by 19.4% over plain ATAU-SP at 1 m/s, narrowing to 2.5% at 30 m/s as list-refresh overhead grows with crossing frequency), with ATAU-SP and ATAU-SP-ML following close behind and remaining consistently 26 to 33% below the best classical dynamic baseline.

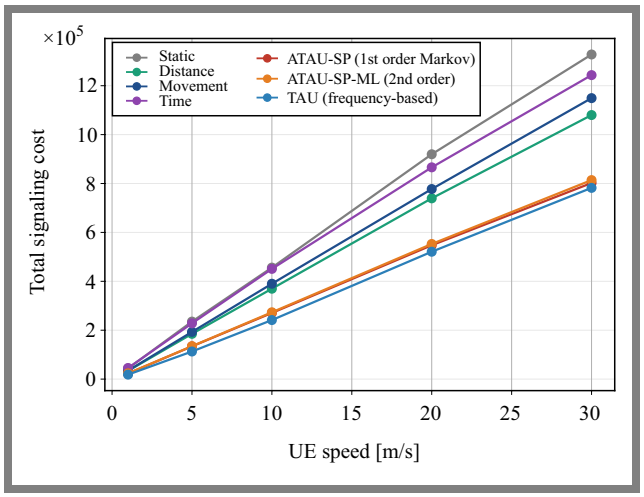


Fig. 11. Total signaling cost vs. UE speed for all seven schemes (random waypoint, $CMR = 1$).

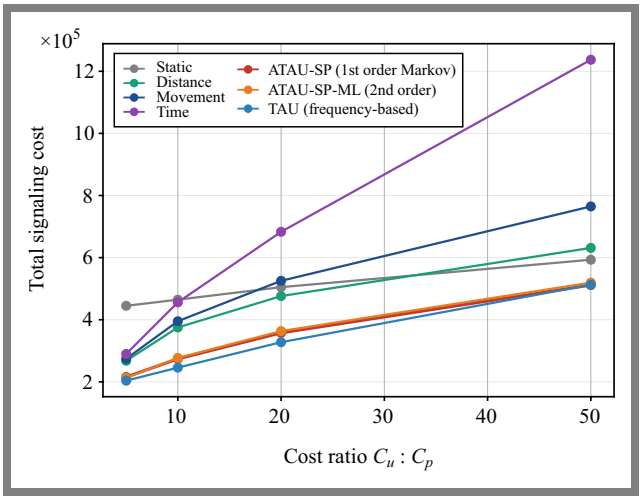


Fig. 12. Total cost vs. cost ratio.

All per-UE schemes remain far below their static and time-based counterparts across the full pedestrian-to-vehicular range. TAL remains the lowest-cost per-UE scheme throughout the swept speed range, while ATAU-SP and ATAU-SP-ML follow closely behind and remain consistently 26–33% below the best classical dynamic baseline.

7.6. Sensitivity to the Registration/Paging Cost Ratio

Table 4 and Fig. 12 show that TAL is the lowest-cost scheme at 5:1, 10:1, and 20:1, but its margin over ATAU-SP narrows sharply as the ratio grows: from 5.4% at 5:1 to essentially a statistical tie at 50:1, where registration cost dominates so heavily that all per-UE schemes with similar TAU rates converge. At 50:1, plain ATAU-SP is in fact marginally the lowest (51 171 vs. TAL's 51 279, a 0.2% difference well within simulation noise).

Classical dynamic baselines (distance, movement, time) degrade badly at high-cost ratios. the time-based approach, is 2.4× more expensive than ATAU-SP at 50:1, since its fixed time interval does not adapt to the increased penalty for registering.

7.7. Paging Delay Bound and an Entropy-based Comparison

Figure 13 illustrates that sweeping D_{max} from 1 to 5 traces the delay-cost trade-off, with the mean delay rises from 1.00 round (blanket paging) to 2.65 rounds, while total cost falls from 39 153 at $D_{max} = 1$ to 26 984 at $D_{max} = 3$, with a further small reduction to 26 038 at $D_{max} = 4$, before

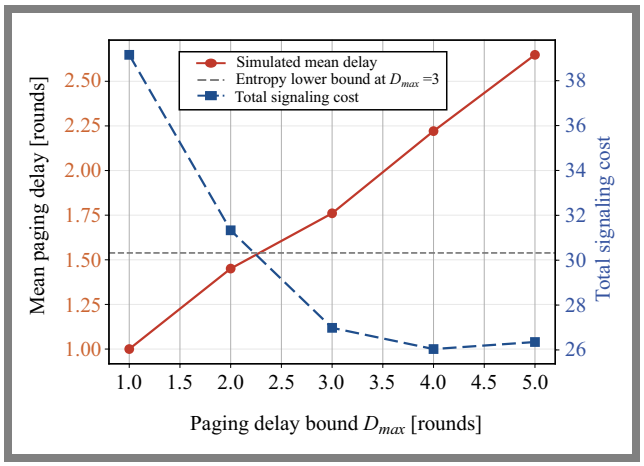


Fig. 13. ATAU-SP total cost and mean paging delay vs. D_{max} delay bound.

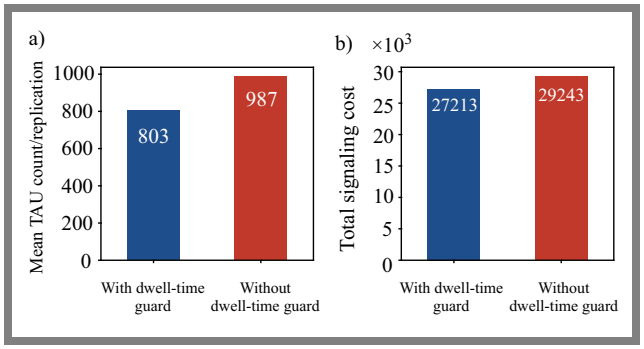


Fig. 14. Dwell-time-guard ablation: TAU count a) and total cost with and without the guard b).

a marginal increase to 26 352 at $D_{max} = 5$. This confirms the choice of $D_{max} = 3$ as a reasonable default on empirical grounds.

The lower bound for $D_{max} = 3$ is approximately 1.54 rounds versus a simulated value of 1.76 rounds – the simple equal-count probability-balanced grouping used by ATAU-SP is within about 14% of the theoretical floor.

7.8. Algorithm Parameter Analysis

The dwell-time guard and the online learning of the Markov transition matrix were previously described only qualitatively. Now, we isolate each by ablation/instrumentation.

At $CMR = 1$, random waypoint, 18 cells/TA, running ATAU-SP with the guard disabled increases the mean TAU count per replication from 803.3 to 987.5 (+22.9%) and increases the total cost from 27 213 to 29 243 (+7.5%). The guard suppresses a mean of 407.8 immediate boundary return crossings per replication that would otherwise each have counted toward the movement threshold and, over a sufficient number of them, triggered spurious TAUs (Fig. 14).

7.9. Model Validation

The simulator's selective, probability-ordered paging achieves $M(K) \leqslant f(K)$ by design, so the analytical curve is expected to sit above the simulated curve at every point (Fig. 15). It does so at the five CMR values tested, with an average gap

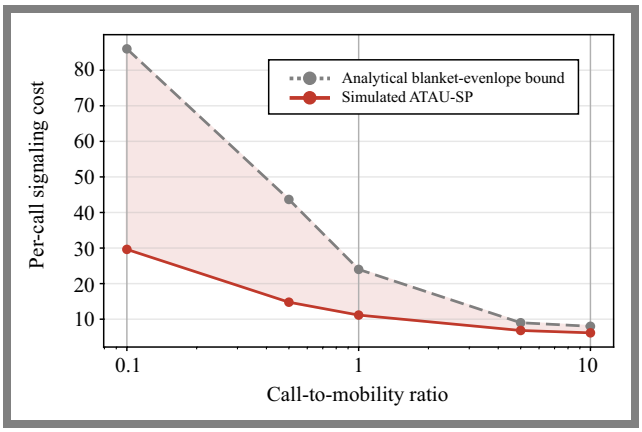


Fig. 15. Analytical blanket envelope bound.

Tab. 5. Total signaling cost: static partitions ($CMR = 1$, RWP). The lowest value per row is shown in bold.

Cells/TA	Static (geometric)	Static (clustered)	ATAU-SP	TAL
5	**20 494**	20 544	27 126	24 470
12	32 587	33 239	27 126	**24 470**
18	45 762	37 377	27 126	**24 470**
37	88 262	88 262	27 126	**24 470**

of 46.5% (ranging from 23.0% in $CMR = 10$ to 66.2% at $CMR = 0.5$). We report this as an envelope gap, not a prediction error.

Equation (6) is validated as a correct, non-trivial upper bound and the gap size quantifies how much selective, learned-probability paging saves over the blanket bound. Because the underlying UE-side and AMF-side operations are $O(1)$ and $O(|A| \log |A|)$, total signaling still scales nearly linearly with the number of UEs, independently of this correction.

To assess whether ATAU-SP is lightweight enough for online use in energy terms, we built a simple, transparently assumption-laden proxy rather than a full energy simulator.

According to [21], a single TAU procedure drains approximately 10 mW of UE battery power, assuming a NAS+RRC exchange duration of approximately 0.5 s. This gives an energy cost of approximately 5 mJ per TAU.

Under this proxy, the relative consumption of UE energy across the schemes tracks their TAU rate and is marginally higher (0.00849/UE/s, ≈ 4.2 µJ/UE/s), while that of the time-based approach is markedly worse (0.01391/UE/s, ≈ 7.0 µJ/UE/s, roughly 77% more TAU energy than ATAU-SP). Note, this is a coarse, explicitly caveated proxy, not a validated power model.

At a dense deployment operating point (18 cells/TA), flow-aware clustering reduces the static approach's cost by 18.3% relative to the geometric partition (37 377 vs. 45 762), confirming that clustering does capture a real, exploitable structure (Tab. 5).

It does not, however, close the gap to the per-UE schemes: clustered static remains 34.5% above TAL and 27.4% above ATAU-SP at this operating point. At 12 cells/TA, the clustered

Tab. 6. Energy-aware threshold selection vs. the network-cost-optimal threshold ($CMR = 1$, RWP, ≈ 18 cells/TA).

Scheme	ATAU-SP (network cost optimal K)	ATAU-SP, energy-aware K
TAU rate [UE/s]	0.00781	0.00438
Network cost	26 659	33 716
UE energy proxy [mJ/UE]	39.07	21.91

partition is marginally worse than the geometric one (33 239 vs. 32 587, a 2.0% increase) – at small tracking-area sizes, the greedy flow-based merge does not reliably outperform simple geometric contiguity, and we report this small negative result rather than only the favorable ones.

At 5 and 37 cells/TA, the two partitions are effectively identical. Overall, clustering offers a genuine improvement over the static baseline, but does not change this paper's central finding that per-UE adaptive schemes (ATAU-SP, ATAU-SP-ML, TAL) dominate any single network-wide partition.

The energy-aware variant cuts the UE energy proxy by 43.9% (39.07 to 21.91 mJ/UE) at the expense of a 26.5% increase in network signaling cost (26 659 to 33 716 – Tab. 6). This is a quantified trade-off rather than a free win, and it is the trade-off that an energy-aware scheme is supposed to make.

8. Conclusions

This paper presented ATAU-SP – a joint, adaptive, per-UE location management scheme for 5G NR that couples an adaptive-threshold registration rule with selective probabilistic paging under a delay bound and with two baselines: a dynamic, frequency-based tracking-area-list (TAL) scheme and a lightweight second-order Markov predictor (ATAU-SP-ML), together with a cost-ratio sensitivity sweep, a formal CMR-estimator convergence analysis, a dwell-time-guard ablation, an entropy-based comparison for the paging delay bound, and a UE energy proxy.

ATAU-SP and ATAU-SP-ML outperform the four classical baselines (static, distance, movement, and time) at every operating point tested. At the nominal operating point, ATAU-SP reduces total signalling cost by approximately 26.5% compared with the best classical dynamic baseline (distance-based) and by 40.4% compared with static registration, while ATAU-SP-ML achieves reductions of approximately 25.7% and 39.8%, respectively. These improvements remain consistent across the mobility-model, speed, area-size, and cost-ratio sweeps. Against the newly implemented TAL baseline, however, ATAU-SP is not uniformly the best: a well-tuned dynamic TAL scheme is the lowest-cost option across the majority of the moderate-to-high call-to-mobility range and across the full swept cost-ratio range. ATAU-SP-ML overtakes it at high CMR (5 – 10), while plain ATAU-SP is the cheapest at very low CMR only.

Future work will focus on improving paging-round grouping and making K^*calibration operating-point-specific or fully data-driven, eliminating the need for a fixed constant κ. We will validate the model using real cell-level carrier mobility traces, including behavioral revisitation patterns, and reassess TAL under such realistic mobility. We will also develop a validated UE energy model that accounts for the state-maintenance costs of ATAU-SP-ML bigram counts and the TAL frequency table, and replace the assumed 0.1 mJ paging-response energy with a measured value. Further work will examine self-similar and Markov-modulated call arrivals, broader CMR-heterogeneity distributions, and beam-level paging in directional NR.

Acknowledgments

The full discrete-event simulator, all driver scripts, result files, and figures are publicly available at https://github.com/kapopat1980/atau.

References

[1] S.M.A. Zaidi, M. Manalastas, H. Farooq, and A. Imran, "Mobility Management in Emerging Ultra-dense Cellular Networks: A Survey, Outlook, and Future Research Directions", *IEEE Access*, vol. 8, pp. 183505–183533, 2020 (https://doi.org/10.1109/ACCESS.2020.3027258).

[2] A. Sufyan *et al.*, "From 5G to Beyond 5G: A Comprehensive Survey of Wireless Network Evolution, Challenges, and Promising Technologies", *Electronics*, vol. 12, art. no. 2200, 2023 (https://doi.org/10.3390/electronics12102200).

[3] Y. Ullah *et al.*, "A Survey on Handover and Mobility Management in 5G HetNets: Current State, Challenges, and Future Directions", *Sensors*, vol. 23, art. no. 5081, 2023 (https://doi.org/10.3390/s23115081).

[4] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile Users: To Update or Not to Update?", *Wireless Networks*, vol. 1, pp. 175–185, 1995 (https://doi.org/10.1007/BF01202540).

[5] A.G. Chekol and M.S. Fufa, "A Survey on Next Location Prediction Techniques, Applications, and Challenges", *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, art. no. 29, 2022 (https://doi.org/10.1186/s13638-022-02114-6).

[6] H. Jang and J. Baek, "Optimal TAL-based Registration with Cell-based Central Policy in Mobile Cellular Networks: A Semi-Markov Process Approach", *The Journal of Supercomputing*, vol. 77, pp. 9172–9189, 2021 (https://doi.org/10.1007/s11227-021-03624-8).

[7] M. Bagaa, T. Taleb, and A. Ksentini, "Efficient Tracking Area Management Framework for 5G Networks", *IEEE Transactions on Wireless Communications*, vol. 15, pp. 4117–4131, 2016 (https://doi.org/10.1109/TWC.2016.2535217).

[8] W.-K. Choi and J.-Y. Pyun, "Machine Learning-based Paging Enhancement in 5G Network", *Applied Sciences*, vol. 12, art. no. 9555, 2022 (https://doi.org/10.3390/app12199555).

[9] C. Rose and R. Yates, "Minimizing the Average Cost of Paging Under Delay Constraints", *Wireless Networks*, vol. 1, pp. 211–219, 1995 (https://doi.org/10.1007/BF01202543).

[10] M. Yan *et al.*, "Mobility Prediction Using a Weighted Markov Model Based on Mobile User Classification", *Sensors*, vol. 21, art. no. 1740, 2021 (https://doi.org/10.3390/s21051740).

[11] I.F. Akyildiz, J.S.M. Ho, and Y.-B. Lin, "Movement-based Location Update and Selective Paging for PCS Networks", *IEEE/ACM Transactions on Networking*, vol. 4, pp. 629–638, 1996 (https://doi.org/10.1109/90.532871).

[12] S.K. Thillaigovindhan *et al.*, "A Comprehensive Survey on Machine Learning Methods for Handover Optimization in 5G Networks", *Electronics*, vol. 13, art. no. 3223, 2024 (https://doi.org/10.3390/electronics13163223).

[13] M. S. Mollel *et al.*, "A Survey of Machine Learning Applications to Handover Management in 5G and Beyond", *IEEE Access*, vol. 9, pp. 45770–45802, 2021 (https://doi.org/10.1109/ACCESS.2021.3067503).

[14] B. Sapkota *et al.*, "5G Network Deployment Planning Using Meta-heuristic Approaches", *Telecom*, vol. 5, pp. 588–608, 2024 (https://doi.org/10.3390/telecom5030030).

[15] B.B. Haile, E. Mutafungwa, and J. Hämäläinen, "A Data-driven Multiobjective Optimization Framework for Hyperdense 5G Network Planning", *IEEE Access*, vol. 8, pp. 169423–169443, 2020 (https://doi.org/10.1109/ACCESS.2020.3023452).

[16] M. Al-Khalidi, R. Al-Zaidi, and M. Hammoudeh, "Network Mobility Management Challenges, Directions, and Solutions: An Architectural Perspective", *Electronics*, vol. 11, art. no. 2696, 2022 (https://doi.org/10.3390/electronics11172696).

[17] H. Riaz, S. Öztürk, and A. Çalhan, "A Robust Handover Optimization Based on Velocity-aware Fuzzy Logic in 5G Ultra-dense Small Cell HetNets", *Electronics*, vol. 13, art. no. 3349, 2024 (https://doi.org/10.3390/electronics13173349).

[18] J. Navarro-Ortiz *et al.*, "A Survey on 5G Usage Scenarios and Traffic Models", *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 905–929, 2020 (https://doi.org/10.1109/COMST.2020.2971781).

[19] A. Haghrah, M.P. Abdollahi, H. Azarhava, and J.M. Niya, "A Survey on the Handover Management in 5G-NR Cellular Networks: Aspects, Approaches and Challenges", *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, art. no. 52, 2023 (https://doi.org/10.1186/s13638-023-02261-4).

[20] I.F. Akyildiz *et al.*, "Mobility Management in Next-generation Wireless Systems", *Proceedings of the IEEE*, vol. 87, pp. 1347–1384, 1999 (https://doi.org/10.1109/5.775420).

[21] A.A.R. Alsaeedy and E.K.P. Chong, "Tracking Area Update and Paging in 5G Networks: A Survey of Problems and Solutions", *Mobile Networks and Applications*, vol. 24, pp. 578–595, 2019 (https://doi.org/10.1007/s11036-018-1160-6).

[22] M.C. González, C.A. Hidalgo, and A.-L. Barabási, "Understanding Individual Human Mobility Patterns", *Nature*, vol. 453, pp. 779–782, 2008 (https://doi.org/10.1038/nature06958).

[23] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communications and Mobile Computing*, vol. 2, pp. 483–502, 2002 (https://doi.org/10.1002/wcm.72).

[24] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of Predictability in Human Mobility", *Science*, vol. 327, pp. 1018–1021, 2010 (https://doi.org/10.1126/science.1177170).

Kalpesh Popat, Ph.D.
Faculty of Computer Applications
https://orcid.org/0000-0002-3818-3717
E-mail: kapopat@gmail.com
Marwadi University, Rajkot, India
https://www.marwadiuniversity.ac.in

Divyakant Meva, Ph.D.
Faculty of Computer Applications
https://orcid.org/0000-0001-8804-7337
E-mail: divyakant.meva@marwadieducation.edu.in
Marwadi University, Rajkot, India
https://www.marwadiuniversity.ac.in

Hardik K. Molia, Ph.D.
Computer Engineering Department
https://orcid.org/0000-0002-1962-7255
E-mail: hardik.molia@gmail.com
Government Engineering College, Bhavnagar, India
https://gecbhavnagar.ac.in

Explainable Network Traffic Classification Using XGBoost and SHAP: Interpreting Feature Contributions Across Traffic Classes

Boubakar Seddik Deghem, Lahcene Aid, and Kadda Mostefaoui

University of Tiaret, Tiaret, Algeria

https://doi.org/10.26636/jtit.2026.3.2724

Submitted: 13 Jul. 2026, Accepted: 31 Aug. 2026

Abstract — Network traffic classification has become an essential component of modern network management, cyber security, and quality-of-service provisioning. The widespread adoption of encryption technologies has reduced the effectiveness of traditional traffic identification techniques, leading to the use of ML approaches based on flow-level characteristics. Although several machine learning and deep learning models were evaluated, the black-box nature complicates the practical deployment in security critical environments. This research proposes a framework for ML-based network traffic classification of explainable artificial intelligence (XAI). The CIC-Darknet2020 dataset is divided into four classes: non-TOR, non-VPN, TOR and VPN. Additionally, machine learning algorithms, including ID3, k-nearest neighbors (KNN), random forest, CatBoost, XGBoost, and LSTM are used as a deep learning approach. The evaluation is carried out through stratified split of trains and 10-fold cross-validation, while XGBoost is classified for explainability analysis. By ensuring model transparency, SHAP (SHapley Additive exPlanations) identifies the most influential features contributing to classification predictions. Furthermore, a novel category SHAP analysis is introduced by grouping higher-level behavioral categories, including temporal, statistical, rate-based, and TCP-related features. The results revealed that temporal traffic characteristics and transport layer behavioral features influence classification outcomes, particularly for encrypted traffic classes. Moreover, the framework demonstrates that high classification performance and model interpretability are achievable simultaneously within a category-level study that enhances the transparency, trustworthiness, and practical applicability of machine learning-based network traffic classification systems.

Keywords — deep learning, explainable AI, machine learning, network traffic classification

1. Introduction

The rapid proliferation of anonymity-preserving technologies such as Tor [1] and virtual private networks (VPN) services [2] has transformed the significance of modern network communication. Although these technologies are being investigated to improve user privacy and protect against surveillance by encrypting data and masking traffic, they also present network monitoring and security enforcement-related challenges.

Recent reports indicate that the overall percentage of web encryption via the HTTPS and TLS protocols will stabilize between 94.6% and 95.7% globally [3], with TLS 1.3 becoming the dominant protocol, accounting for more than 72.8% of encrypted traffic. This shift is accompanied by a stable HTTP/3 (QUIC-based over UDP) adoption rate equaling 19.8% to 21.1% of total web requests due to operational challenges affecting throughput on wired networks, while HTTP/2 maintains a 52.3% share. As far as emerging security trends are concerned, the use of postquantum cryptography (PQC) algorithms such as X25519MLKEM768 on the browser side has increased to 55.7% to protect against "collect now, decompose later" attacks [4]. While encryption enhances user privacy, it has also led to a security vulnerability. Cybersecurity reports [5] show that more than 86% of threats and malware are now hidden inside encrypted tunnels, while one in three organizations is unable to scan encrypted traffic internally due to privacy and performance considerations.

As encrypted traffic increasingly dominates network environments, traffic identification methods based on port numbers and deep packet inspection (DPI) have become less effective. Since payload contents are no longer accessible, modern traffic analysis systems must depend on behavioral and statistical characteristics, such as packet interarrival times, flow durations, packet size distributions, traffic rates, and detonate patterns. Consequently, automatic network traffic classification (ANTC) has emerged as an important component of network management, cybersecurity, quality of service provisioning, and anomaly detection. Precise traffic identification enables network administrators to optimize resource allocation, enforce security policies, detect malicious activities, and improve network performance.

To cope with these limitations, researchers have increasingly adopted machine learning (ML) and deep learning (DL) techniques capable of learning distinctive patterns directly from traffic statistics and flow-level characteristics. In the past, numerous studies have investigated ML-based classifiers,

This work is licensed under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.
For more information, see https://creativecommons.org/licenses/by/4.0/

including random forest, support vector machines, gradient boosting, and XGBoost. More recently, DL architectures such as convolutional neural networks (CNNs), long short-term memory networks (LSTMs), gated recurrent units (GRUs), and hybrid models have demonstrated promising performance across diverse traffic environments. These approaches have mainly advanced the state of high classification accuracy on several benchmark datasets.

Most classification systems operate as black-box models, making it difficult for network administrators and security analysts to understand the factors that influence the predictions of the model. Furthermore, although explainable artificial intelligence (XAI) techniques have recently attracted increasing attention, most studies focus on individual feature importance while skipping the analysis of broader traffic feature categories, such as temporal, statistical, rate-based, and protocol-related characteristics. Consequently, the behavioral patterns on which traffic classification predictions are based remain insufficiently understood, particularly in encrypted traffic environments.

In such a context, this study addresses the problem of classifying encrypted network traffic, focusing on distinguishing between non-TOR, non-VPN, TOR, and VPN classes. The proposed framework is designed to classify multiple traffic categories while providing transparent and interpretable explanations of the model predictions. Consequently, this can be regarded as a significant issue in the field of cybersecurity in light of the increasing use of encrypted applications and the difficulty of analyzing them through traditional methods. Furthermore, this study proposes an explainable network traffic classification framework based on XGBoost and SHAP. The study also addresses the challenges related to ML models limited interpretability by using the category-SHAP method.

By revealing how feature categories influence classification predictions, the framework facilitates more effective feature selection, simultaneously improving reliability, interpretability, and performance of network traffic classification systems.

2. Related Work

In recent years, the classification of network traffic has attracted considerable interest due to its importance in the proves of managing networks, ensuring the required QoS, and guaranteeing cybersecurity. Earlier methods used manual feature engineering and traditional ML techniques to perform network traffic classification. Decision trees were effective in sorting multiple types of network traffic using both categorical and continuous features, thus allowing the identification of complex traffic patterns and malicious behaviors [6]. Similarly, interpretable probabilistic approaches based on Markov chains and hidden Markov models were investigated for traffic characterization and prediction. These methods offered precise information on traffic behavior while achieving competitive performance compared to advanced ML techniques [7].

As modern communication becomes more complex and encryption becomes more widespread, researchers have increasingly moved towards using deep learning for identifying Internet traffic. In [3], the authors used deep learning to find distinctive characteristics of raw Internet traffic data, by taking advantage of the earlier studies. They proposed a traffic identification framework based on deep learning methods that reduced the dependence on manually engineered features by automating extraction relevant to the characteristics of the traffic network flows. The authors of [4] developed an end-to-end encrypted traffic classification framework specifically implementing 1D-CNNs, which resulted in substantial performance improvements on the ISCX VPN non-VPN datasets without dependence on hand-crafted features.

Further advances were achieved with the deep packet approach, proposed in [8], which combines stacked autoencoders and CNNs to automatically learn traffic representations from raw packet data. The framework achieves high classification performance for both application identification and traffic categorization tasks, including encrypted, VPN, and non-VPN.

More recently, lightweight deep learning architectures based on depth-wise separable convolutions (DSCs) have been introduced to reduce computational complexity while maintaining high classification performance. Such models have achieved state-of-the-art accuracy in benchmark datasets and demonstrated suitability for resource-constrained environments, including IoT and edge computing [9].

Despite the remarkable classification performance achieved by ML and deep learning models, most existing approaches operate as black-box systems, providing limited insight into the rationale behind their predictions. Consequently, explainable artificial intelligence (XAI) has appeared as a promising research direction to improve transparency, trustworthiness, and interpretability of network traffic analysis [10]. XAI techniques can be widely classified into model-specific and model-agnostic approaches, connected to local and global scales operating countless popular methods including SHAP [11], LIME [12], integrated gradients [13], layer-wise relevance propagation (LRP) [14], and occlusion analysis [15].

To address the black-box nature of deep learning models, researchers used XAI techniques to interpret multimodal traffic classification systems and generate global explanations of classifier behavior [16]. Similarly, explainable anomaly detection frameworks have combined ML models with SHAP explanations to improve transparency in encrypted traffic analysis, achieving high detection performance on data sets such as CIC-Darknet2020 while identifying the most influential traffic characteristics [17].

More recently, in [18], the effectiveness of SHAP in encrypted traffic analysis was demonstrated by identifying critical traffic characteristics such as packet size, inter-arrival time, and TLS-related attributes, while maintaining detection accuracy above 99%.

Beyond traffic classification, XAI has also been applied to cybersecurity domains such as IoT anomaly detection and log-based anomaly analysis. In IoT environments, SHAP-

Tab. 1. Comparative analysis with existing approaches.

Ref.	Classification	Encrypted traffic	SHAP	Class exp.	Feature category	Behavioral interp.
[9]	✓	✓	✓	✓	×	×
[16]	✓	✓	✓	×	×	×
[17]	×	✓	✓	✓	×	×
[18]	✓	✓	✓	✓	×	×
Proposed	✓	✓	✓	✓	✓	✓

based frameworks have been used to provide explanations for anomaly detection decisions, thus enhancing trust in AI-driven security systems [19]. Similarly, the SXAD framework integrated SHAP into the logarithmic anomaly detection process, providing local and global explanations while also achieving near-perfect detection performance, and enabling root-cause analysis of abnormal events [20].

Recent studies focusing on XAI improved the interpretability of network traffic analysis systems; most existing approaches focus primarily on individual feature importance or model-level explanations. Adequately, understanding of traffic behavior through higher-level semantic feature groups has received little attention. Consequently, there remains a need for explainability frameworks capable of providing behavioral interpretations of network traffic by analyzing the contribution of feature categories rather than isolated features. This research gap is the main focus of this work, which introduces a category-based SHAP analysis framework to explain encrypted traffic classification decisions from a behavioral perspective.

As summarized in Tab. 1, existing studies have integrated SHAP into network traffic analysis to provide explanations and class-level interpretations. However, most of these works focus on the importance of individual features without investigating the contribution of higher-level behavioral feature groups. Adequately, no previous study has introduced a category-based SHAP framework for encrypted traffic classification that explains model predictions through semantic feature categories. Therefore, this work deals with existing explainability approaches by providing both feature-level and category-level interpretations, enabling a more comprehensive understanding of network traffic behavior.

3. Proposed Framework

The framework's structure comprises four main stages: data preprocessing, feature engineering, model training, and explainability analysis using SHAP-based methods (Fig. 1).

3.1. Data Preprocessing and Feature Preparation

This study uses the CIC-Darknet2020 dataset [21], which is widely used for encrypted and darknet traffic analysis, and provides realistic network traffic traces collected from various communication scenarios. To create a representative dataset, we merge our previously generated data sets (ISCXTor2016 [22] and ISCXVPN2016 [23]) and combine the respective VPN and TOR with the corresponding darknet categories.

Each network flow is represented by a set of statistical and behavioral features extracted using CICFlowMeter. These features describe various characteristics of network traffic, including flow duration, packet length statistics, interarrival times, packet rates, TCP flag information, and other flow-level attributes. Since the data set relies on flow-based features rather than payload inspection, it is particularly suitable for analyzing encrypted traffic while preserving user privacy.

For the purposes of this study, the classification of the original data was retained, comprising four main categories of traffic: non-TOR, non-VPN, TOR, and VPN.

The CIC-Darknet2020 dataset was subjected to a preprocessing stage to ensure data quality and suitability for ML-based traffic classification. First, records containing missing values were removed from the dataset. Subsequently, the infinite values resulting from flow-level calculations were replaced with missing values and eliminated to avoid numerical inconsistencies during the model training phase.

To prevent information leakage and improve model generalization, non-informative identifiers such as Src IP, Dst IP, Src Port, Dst Port and Flow ID were excluded from the feature set. Categorical attributes were transformed into numerical representations using label encoding, while numerical flow-based features were retained in their original form (Fig. 2). As a result, each network flow was represented by a set of statistical and behavioral characteristics describing temporal behavior, packet size distributions, traffic rates, and TCP-related dynamics, enabling encrypted traffic classification without relying on packet payload inspection.

3.2. Traffic Modeling

To classify encrypted network traffic, six algorithms were evaluated: ID3, k-nearest neighbors (KNN), random forest, CatBoost, XGBoost, and LSTM as a deep learning approach. Due to their capabilities, the first five models are conventional ML algorithms designed for tabular flow level data, while LSTM was selected to capture long-term temporal dependencies and sequential patterns in network traffic flows. Such an approach renders the model well prepared in terms of temporal characteristics, encrypted traffic, and distinguishing between different traffic classes by identifying the most effective model, whereas random forest was included as an ensemble tree-based baseline to provide a comparison with

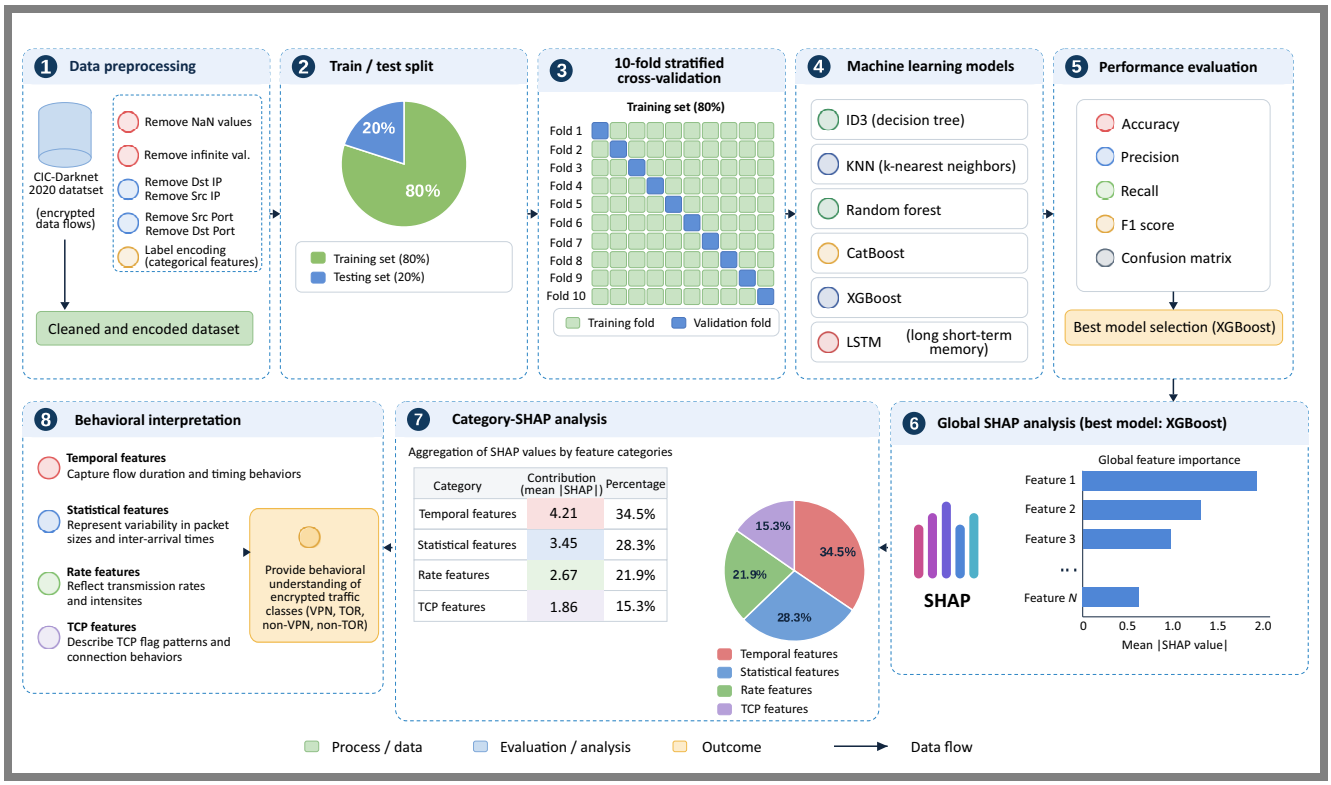


Fig. 1. Overview of the proposed framework.

the gradient-boosting approaches, namely CatBoost and XGBoost.

The extreme gradient boosting algorithm is widely used as a model in multiclass classification, due to its high proficiency in dealing with tabular data and its ability to model nonlinear feature relationships. The model is developed using 300 decision trees to guarantee incremental learning and improve performance. To avoid overfitting, the maximum depth of each tree is limited to 8 levels, achieving balance between learning ability and avoiding overfitting. In addition, a 0.05 learning rate is adopted to ensure gradual and stable updates during the training process.

To increase the generalizability of the model, subsampling is used at 80%, so that each tree is trained on a random

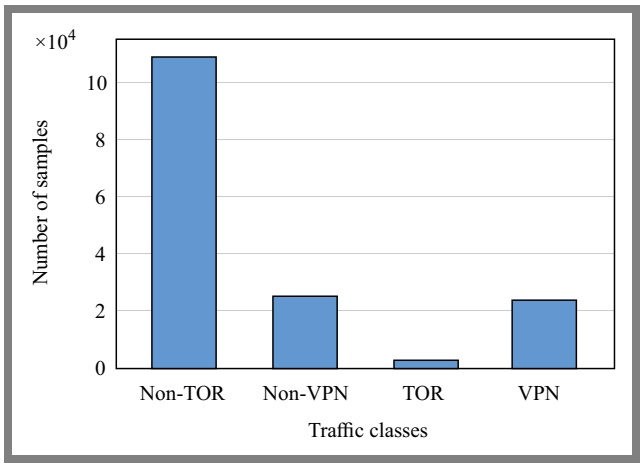


Fig. 2. Distribution of traffic categories in the Darknet 2020 dataset.

sample of training data to select a subset of features during the construction of each tree, reducing correlation between the trees and improving performance. Moreover, the softmax function is adopted to directly retrieve the predicted class using the multiclass loss function to monitor the learning quality during the training phase, considering that the issue is mainly about the classification of four different categories. To guarantee reproducibility of the results, the random seed is set at a parameter value of 42.

To evaluate the generalization capacity of the model's generalization ability and its capability to reduce the impact of data splitting, a k-fold stratified cross-validation strategy was applied to the training set in such a way that the training data is divided to ten equal components while preserving the exact same relative class distribution in each component [24].

In each training course, 9 components are used to train the model and 1 component is relied upon for the verification of results. The choice of $k = 10$ provides a balance between computational efficiency and reliable performance estimation [25]. The process repeats itself ten times in such a way that each component is used only once for verification purposes. In the end, the average classification accuracy is calculated across all folds, providing a more reliable evaluation of the model's performance.

Each model was trained to classify network flows into traffic categories: non-TOR, non-VPN, TOR, and VPN, using flow-level statistical and behavioral features extracted from network traffic. After comparing the classification performance of all candidate models using standard evaluation metrics, the best-performing model was selected and then used for the SHAP-

based explainability analysis to investigate the contribution of individual features and feature categories to the classification process.

3.3. Feature Learning and Selection via SHAP

To move beyond black-box prediction, the Shapley additive explanations (SHAP) approach is integrated to quantify feature contributions to model predictions. First, a global SHAP analysis is performed to classify all features according to their importance. Instead of selecting an arbitrary number of characteristics, a cumulative importance strategy is applied, retaining only those that contribute to 95% of the total explanatory power.

This results in a reduced and meaningful subset of discriminative features, ensuring both efficiency and interpretability.

3.4. Category-based Explainability Analysis

To provide a higher-level semantic interpretation, selected features are grouped into four behavioral categories: temporal, statistical, rate, and window-based features. Then, category-SHAP is applied to quantify the contribution of each category to the prediction of each traffic class. This step enables the transition from feature-level explanation to behavior level understanding, allowing to interpret the influence of different traffic dynamics on classification decisions in VPN and TOR environments.

To obtain the category-SHAP values, the individual SHAP contributions of features belonging to the same category are aggregated.

Let ϕ_i denote the SHAP value of feature i, and C_k represent the k-th feature category containing a set of features $\{i_1, i_2, \ldots, i_m\}$. The category-SHAP value for category C_k is computed as the sum of the absolute SHAP values of all features belonging to that category as:

$$\phi(C_k) = \sum_{i \in C_k} \left|\phi_i\right|, \tag{1}$$

where $\phi(C_k)$ represents the magnitude of the overall contribution of category C_k to the model prediction, and $|\phi_i|$ measures the magnitude of the contribution of each feature regardless of its direction (positive or negative impact).

To enable a comparison between different categories, we aggregated the values from Eq. (1) which are normalized according to:

$$\Phi_{\text{Norm}}(C_k) = \frac{\phi(C_k)}{\sum_{j=1}^{K} \phi(C_j)}, \tag{2}$$

where K is the total number of feature categories.

The resulting normalized category-SHAP values represent the relative importance of each category in the final prediction. Algorithm 1 illustrates the proposed prediction method.

Algorithm 1 Category-SHAP aggregation and normalization

Require: SHAP values $\boldsymbol{\phi} = \phi_1, \phi_2, \ldots, \phi_N$ and feature categories $\mathcal{C} = C_1, C_2, \ldots, C_K$
Ensure: Normalized category-SHAP values $\Phi_{\text{Norm}}(C_1), \ldots, \Phi_{\text{Norm}}(C_K)$

for $k = 1$ to K **do**
  $\phi(C_k) \leftarrow 0$
  for each feature $i \in C_k$ **do**
    $\phi(C_k) \leftarrow \phi(C_k) + |\phi_i|$
  end for ▷ Eq. (1)
end for
$S \leftarrow \sum_{j=1}^{K} \phi(C_j)$
for $k = 1$ to K **do**
  if $S > 0$ **then**
    $\Phi_{\text{Norm}}(C_k) \leftarrow \frac{\phi(C_k)}{\sum_{j=1}^{K} \phi(C_j)}$ ▷ Eq. (2)
  else
    $\Phi_{\text{Norm}}(C_k) \leftarrow 0$
  end if
end for
return $\Phi_{\text{Norm}}(C_1), \ldots, \Phi_{\text{Norm}}(C_K)$

4. Results and Discussion

This section presents the experimental findings of the proposed framework and discusses both classification performance and explainability insights derived from the SHAP-based analysis.

4.1. Classification Performance

The XGBoost classifier achieved strong performance in distinguishing between the four traffic categories (non-TOR, non-VPN, TOR, and VPN). Using an 80/20 stratified split and 10-fold cross-validation, the model demonstrated stable accuracy across folds, confirming its robustness on imbalanced and high-dimensional network traffic data.

The confusion matrix further indicates that most misclassifications occur between closely related encrypted traffic types (e.g., VPN vs. non-VPN), which is expected due to similar flow-level statistical behavior. Overall, the model achieves high predictive capacity, confirming that flow-based features extracted from CICFlowMeter are sufficient to model encrypted traffic patterns without relying on payload inspection.

Figure 3 presents the confusion matrix of the XGBoost classifier in the CIC-Darknet2020 dataset. Most samples are correctly classified, as indicated by the strong concentration of values along the main diagonal. The model achieved excellent recognition of non-TOR (22 068 samples), non-VPN (4 629 samples), TOR (257 samples) and VPN (4 318 samples).

The majority of misclassifications occurred between VPN and non-VPN, suggesting that these classes share similar statistical characteristics despite the presence of VPN tunneling. On the contrary, very limited confusion was observed between TOR and VPN, indicating that the model successfully

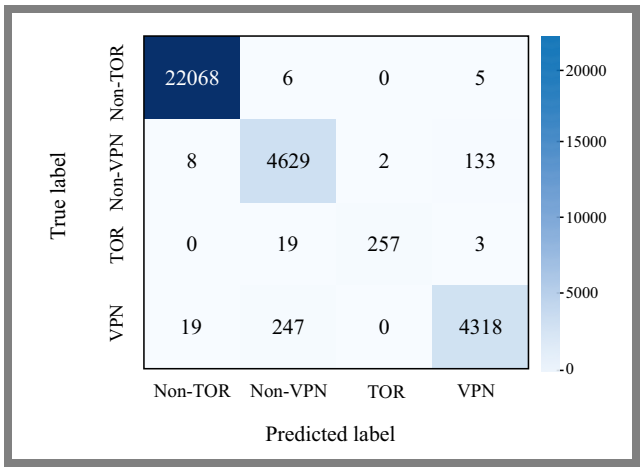


Fig. 3. Confusion matrix of the XGBoost model.

Tab. 2. Performance comparison of different models for encryption traffic classification.

Model	Accuracy	Precision	Recall	F1 score
ID3	0.9824	0.9508	0.9546	0.9526
KNN	0.9690	0.9687	0.9690	0.9688
Random forest	0.9814	0.9633	0.9471	0.9549
CatBoost	0.9837	0.9731	0.9493	0.9606
LSTM	0.9638	0.9348	0.9100	0.9217
XGBoost	0.9861	0.9862	0.9861	0.9860

captures the distinctive behavioral patterns of TOR communications. Overall, the confusion matrix confirms the effectiveness of XGBoost in discriminating between encrypted and unencrypted traffic categories using flow-level features.

Table 2 summarizes the performance of the evaluated models. XGBoost achieved the highest overall performance, with an accuracy of 0.9861 and an F1 score of 0.9860. The random forest achieved an F1 score of 0.9549, providing a competitive ensemble tree baseline but remaining below the performance benchmark of the gradient boosting models. LSTM achieved an F1 score of 0.9217, which was lower than the performance of the evaluated tree-based models for the considered flow level representation. The strong performance of XGBoost demonstrates its ability to effectively capture complex traffic patterns and distinguish between classes. Consequently, XGBoost was selected as the reference model for subsequent SHAP and category-SHAP explainability analyzes.

The receiver operating characteristic (ROC) curves of the four proposed models for the selected traffic types are shown in Fig. 4. All of the ROC curves are situated preferentially at the top left corner of the diagram, exhibiting an impressive balance between true positive rate (TPR) and false positive rate (FPR) throughout different threshold levels. The AUC indexes of the ROC curves are quite impressive with 1.0000 for the non-TOR, 0.9984 for the non-VPN, 0.9998 for the TOR, and 0.9985 for the VPN.

These values indicate the great discriminative power of the model, which allows it to effectively classify each traffic type. Moreover, the considerable distance of the ROC curves from the diagonal line of the random classifier would be an additional proof of the model's superiority to random guessing. Lastly, the high values of AUC mean that the model is both reliable, robust, and offers a high level of generalization in terms of its classification capability.

4.2. Global SHAP Interpretation

To improve interpretability, SHAP analysis was applied to quantify the contribution of each feature to the model's predictions.

The global SHAP results revealed that the model is primarily driven by a small subset of dominant features. The cumulative SHAP analysis showed that only 35 features are required to explain 95% of the model's predictions, significantly reducing the feature space while preserving interpretability, As shown in Fig. 5, of all features, *Idle Max* was identified as the most influential variable, followed by *FWD Init Win Bytes*, *Bwd Init Win Bytes* and *Idle Mean*. These results indicate that temporal behavior and TCP window-related characteristics are the most discriminative factors in encrypted traffic classification, while traditional packet-level features contribute less significantly.

4.3. Category-SHAP Analysis

For a better understanding of the impact of the features, the extracted SHAP features were classified into four semantic groups. The classification success shows that the temporal characteristics always exerted the greatest impact on the entire traffic class. It is evident from Fig. 6 that the rate features were the second most effective group after the temporal features for TCP and non-TOR. Statistical features have shown moderate and steady results when it comes to their contribution. Meanwhile, the window features were able to exert the least substantial impact in general, although they still played a role in distinguishing the classes.

The class-level analysis showed that while VPN relied on temporal features the most, non-TOR was influenced the most by all feature groups. Overall, the suggested model does not rely on payloads much when it comes to classification.

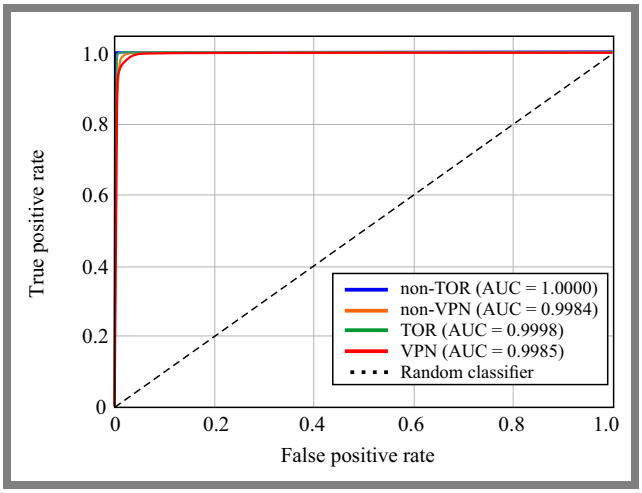


Fig. 4. ROC curves and corresponding AUC values for four types of traffic.

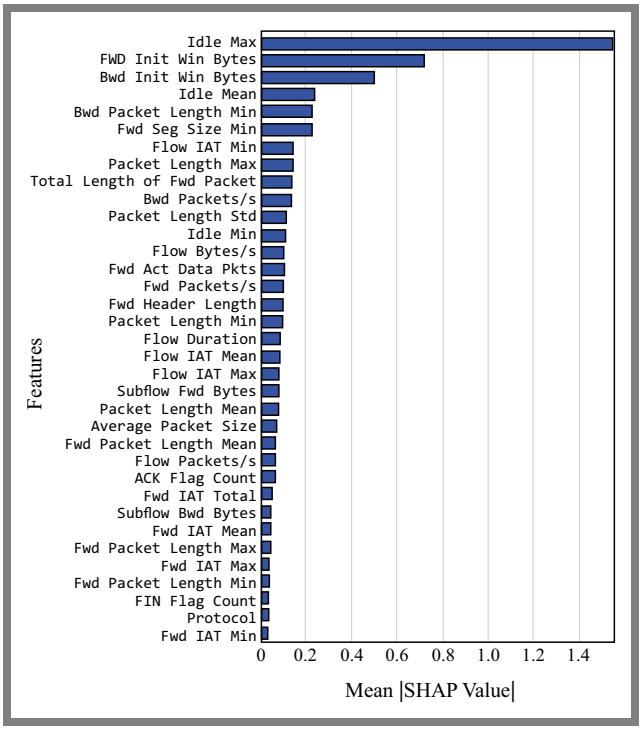


Fig. 5. 35 selected features representing 95% cumulative SHAP importance.

Figure 6 presents the category-SHAP analysis for the four traffic classes. Temporal features contribute to the classification process the most, particularly for non-VPN and VPN, indicating that timing-related behavior is the primary factor used by the model. TCP window features also play a significant role, especially for TOR and non-TOR. On the contrary, rate features show the lowest contribution across all classes.

These results suggest that the model relies mainly on traffic behavior and TCP communication dynamics to distinguish encrypted traffic categories.

4.4. Class-level Behavioral Insights

The results of the category-level SHAP analysis are shown in Tab. 3 and reveal different behavioral patterns across the various traffic classes. Temporal characteristics provide the dominant contribution for non-TOR (38.99%), non-VPN

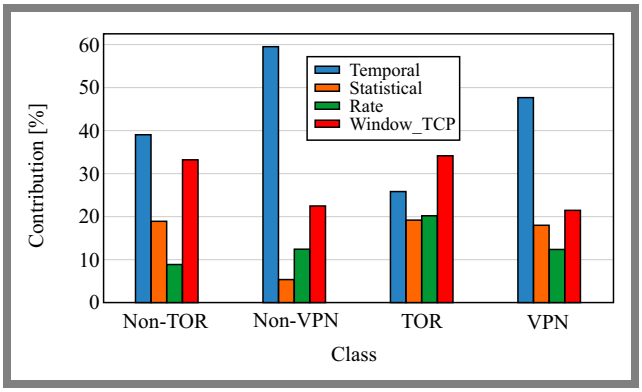


Fig. 6. Category-SHAP contribution analysis across traffic classes.

Tab. 3. Category-SHAP contribution analysis across traffic classes.

Class	Temporal	Statistical	Rate	Window_TCP
Non-TOR	38.99%	18.86%	8.90%	33.25%
Non-VPN	59.62%	5.35%	12.52%	22.52%
TOR	25.86%	19.32%	20.42%	34.40%
VPN	47.85%	18.12%	12.51%	21.52%

(59.62%), and VPN (47.85%), indicating that temporal flow characteristics are important for their classification.

In contrast, Window_TCP is the most influential feature category for TOR (34.40%), followed by rate (20.42%) and statistical features (19.32%). Also, non-TOR shows a substantial contribution from Window TCP (33.25%), meanwhile rate features are characterized by a comparatively lower contribution (8.90%).

Broadly speaking, the results indicate that the model exploits observable temporal and transport-layer behavioral characteristics, rather than payload information distinguishing encrypted and anonymized traffic.

4.5. Key Findings and Implications

Combining XGBoost with SHAP-based interpretability delivers robust classification performance and meaningful interpretability. The cumulative SHAP approach reduces the mean of the feature dimensions, while maintaining interpretability. Additionally, it makes the model more efficient and applicable. Furthermore, category-SHAP analysis provides a deeper understanding of traffic behavior. In particular, valuable cybersecurity applications are as interpretable and confident as model predictions.

To evaluate the effectiveness of the main framework, it was also compared to the LSTM approach, given its ability to model dependencies and temporal relationships in sequencing network traffic. The performance differences among XGBoost, random forest, and LSTM can be attributed primarily to the representation of the input data. Although network traffic contains some temporal characteristics, the data used in this study are represented as aggregated flow-level features rather than fine-grained sequences.

This representation favors tree-based models which can effectively capture non-linear relationships and interactions among structured features, while also limiting the ability of LSTM to exploit temporal dependencies.

Nevertheless, the superior performance of the proposed model indicates that effective traffic classification relies not only on temporal information or transport-layer characteristics, but also on complementary statistical, behavioral, and protocol-related patterns. By capturing these multidimensional characteristics, the proposed approach provides a more comprehensive representation of the behavior of network traffic, leading to better performance.

5. Conclusions

This study presented an explainable machine learning framework for encrypted network traffic classification using flow-based statistical and behavioral features extracted from the CIC-Darknet2020 dataset. Several algorithms, including ID3, k-nearest neighbors (KNN), random forest, CatBoost, XGBoost, and LSTM were employed to evaluate the classification of network traffic into four categories: non-TOR, non-VPN, TOR, and VPN. The experimental results demonstrated that the selected model offers good classification performance, confirming the effectiveness of flow-level features for encrypted traffic analysis without relying on packet payload inspection.

To improve model transparency, SHAP was used to explain the contribution of individual characteristics to classification decisions. The global SHAP analysis revealed that a limited subset of characteristics accounted for most of the model's predictive behavior, with temporal and TCP-related attributes emerging as the most influential factors. Furthermore, a cumulative SHAP-based feature selection strategy showed that a reduced set of features was sufficient to explain most of the model's predictions.

The main contribution of this work is dependent on the introduction of a category-SHAP analysis framework, which extends traditional feature-level explanations by grouping features into higher-level behavioral categories. This approach provides a more intuitive understanding of encrypted traffic behavior and enables cybersecurity analysts to interpret classification decisions from a behavioral perspective rather than through isolated features alone.

In general, the proposed framework demonstrates that explainability and high predictive performance can be achieved simultaneously in encrypted traffic classification. The obtained results contribute to the development of more transparent, trustworthy, and operationally useful AI-based network security systems.

References

[1] D. Sarkar, P. Vinod and S.Y. Yerima, "Detection of Tor Traffic using Deep Learning", *2020 IEEE/ACS 17th International Conference on Computer Systems and Applications (AICCSA)*, Antalya, Turkey, 2020 (https://doi.org/10.1109/AICCSA50499.2020.9316533).

[2] R. Venkateswaran, "Virtual Private Networks", *IEEE Potentials*, vol. 20, pp. 11–15, 2001 (https://doi.org/10.1109/45.913204).

[3] Cloudflare, "Worldwide Overview: Cloudflare Radar". [Online] Available: https://radar.cloudflare.com [Accessed: 10.07.2026].

[4] V. Gautam and S. Ganguli, *Securing Digital India Against AI and Cyber Threats*, Zscaler, 2026 (ISBN: 9798903339068).

[5] Y. Chandiramani, "Attention aux Fausses Versions de Super Mario Run sur Android!", *Global Security Mag Online*, 2017 https://www.globalsecuritymag.fr/Yogi-Chandiramani-Zscaler,20170116,68299.html (in French).

[6] S. Mali, M. Gujral, and A.K. Cherukuri, "Encrypted Network Traffic Classification Using Intelligent Techniques", *Cureus Journal of Computer Science*, 2025 (https://doi.org/10.7759/s44389-024-02701-2).

[7] G. Aceto *et al.*, "Characterization and Prediction of Mobile-app Traffic Using Markov Modeling", *IEEE Transactions on Network and Service Management*, vol. 18, pp. 907–925, 2021 (https://doi.org/10.1109/TNSM.2021.3051381).

[8] M. Lotfollahi, R.S.H. Zade, M.J. Siavoshani, and M. Saberian, "Deep Packet: A Novel Approach for Encrypted Traffic Classification Using Deep Learning", *ArXiv*, 2018 (https://doi.org/10.48550/arXiv.1709.02656).

[9] M. Ghaleb *et al.*, "Explainable AI for Lightweight Network Traffic Classification Using Depthwise Separable Convolutions", *IEEE Open Journal of the Computer Society*, vol. 6, pp. 908–920, 2025, (https://doi.org/10.1109/OJCS.2025.3576495).

[10] H. Hagras, "Towards Human Understandable Explainable AI", *Computer*, vol. 51, pp. 28–36, 2018 (https://doi.org/10.1109/MC.2018.3620965).

[11] S.M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions", ArXiv, 2017 (https://doi.org/10.48550/arXiv.1705.07874).

[12] M.T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier", *Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016 (https://doi.org/10.1145/2939672.2939778).

[13] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks", *ArXiv*, 2017 (https://doi.org/10.48550/arXiv.1703.01365).

[14] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning Important Features Through Propagating Activation Differences", *ArXiv*, 2019 (https://doi.org/10.48550/arXiv.1704.02685).

[15] M.D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks", *Computer Vision – ECCV 2014*, pp. 818–833, 2014 (https://doi.org/10.1007/978-3-319-10590-1_53).

[16] A. Nascita *et al.*, "XAI Meets Mobile Traffic Classification: Understanding and Improving Multimodal Deep Learning Architectures", *IEEE Transactions on Network and Service Management*, vol. 18, pp. 4225–4246, 2021 (https://doi.org/10.1109/TNSM.2021.3098157).

[17] K. Singh, A. Kashyap, and A.K. Cherukuri, "Interpretable Anomaly Detection in Encrypted Traffic Using SHAP with Machine Learning Models", *ArXiv*, 2025 (https://doi.org/10.48550/arXiv.2505.16261).

[18] S.N. Zeleke, A.F. Jember, and M. Bochicchio, "Integrating Explainable AI for Effective Malware Detection in Encrypted Network Traffic", *ArXiv*, 2025 (https://doi.org/10.48550/arXiv.2501.05387).

[19] A.N. Gummadi, J.C. Napier, and M. Abdallah, "XAI-IoT: An Explainable AI Framework for Enhancing Anomaly Detection in IoT Systems", *IEEE Access*, vol. 12, pp. 71024–71054, 2024 (https://doi.org/10.1109/ACCESS.2024.3402446).

[20] K. Alam *et al.*, "SXAD: Shapely eXplainable AI-Based Anomaly Detection Using Log Data", *IEEE Access*, vol. 12, pp. 95659–95672, 2024 (https://doi.org/10.1109/ACCESS.2024.3425472).

[21] Canadian Institute for Cybersecurity (CIC), "CIC-Darknet2020", *University of New Brunswick*. [Online] Available: https://www.unb.ca/cic/datasets/darknet2020.html [Accessed: 10.07.2026].

[22] Canadian Institute for Cybersecurity (CIC), "Tor-nonTor dataset (ISCXTor2016)", *University of New Brunswick*. [Online] Available: https://www.unb.ca/cic/datasets/tor.html [Accessed: 10.07.2026].

[23] Canadian Institute for Cybersecurity (CIC), "VPN-nonVPN dataset (ISCXVPN2016)", *University of New Brunswick*. [Online] Available: https://www.unb.ca/cic/datasets/vpn.html [Accessed: 10.07.2026].

[24] J.R. Almonteros and J.B. Matias, "Integration of Stratified KFold Cross Validation to Enhance Prediction Accuracy: A Comparison

Study", *2024 5th International Conference on Data Analytics for Business and Industry (ICDABI)*, Zallaq, Bahrain, 2024 (https://doi.org/10.1109/ICDABI63787.2024.10800425).

[25] O. Oyedele, "Determining the Optimal Number of Folds to Use in a K-fold Cross-validation: A Neural Network Classification Experiment", *Research in Mathematics*, vol. 10, art. no. 2201015, 2023 (https://doi.org/10.1080/27684830.2023.2201015).

Boubakar Seddik Deghem, Ph.D. Student
Artificial Intelligence and Systems Research Laboratory (LRIAS)
https://orcid.org/0009-0001-9828-4544
E-mail: boubakarseddik.deghem@univ-tiaret.dz
University of Tiaret, Tiaret, Algeria
https://www.univ-tiaret.dz

Lahcene Aid, Ph.D., Professor
Artificial Intelligence and Systems Research Laboratory (LRIAS)
https://orcid.org/0000-0001-9812-2871
E-mail: a_lahcen@esi.dz
University of Tiaret, Tiaret, Algeria
https://www.univ-tiaret.dz

Kadda Mostefaoui, Ph.D., Professor
Artificial Intelligence and Systems Research Laboratory (LRIAS)
https://orcid.org/0000-0002-6367-1245
E-mail: kadda.mostefaoui@univ-tiaret.dz
University of Tiaret, Tiaret, Algeria
https://www.univ-tiaret.dz

Information for Authors

Journal of Telecommunications and Information Technology (JTIT) is published quarterly since 2000. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review.** Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

We encourage submissions from a diverse range of authors from across all countries and backgrounds.

Manuscript

Latex files are preferred and Editorial Office provides a style to prepare the material along with the documentation. We also accept Microsoft Word and PDF files. A typical article is 10 pages long (approximately 6,000 words) and must include the following contents:

- Authors' names and affiliations in the following format:
  First name and surname (last name), academic title,
  Position held,
  ORCID number,
  E-mail address from the University's domain,
  Faculty and name of the University,
  Link to University website.
- Abstract (150-200 words). The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.
- Keywords related to the content of the article. About four keywords or phrases in alphabetical order should be used, separated by commas.
- The content of the article in a typical structure, i.e.: introduction, related work, conducted research, conclusions, references.

Figures, Tables and Photos

Together with the article, please send files with graphics with the highest resolution available, 150 dpi or more in bitmap resolution (jpg, png) and vector (cdr, svg, ps, pdf) formats are welcomed.

References

We use four main citation styles for a journal article, for an Internet article, for a conference paper, and for a book. Below are examples of citations. In each item, the DOI number or link to the PDF of the cited article should be provided.

[1] R.K. Meyers and A.H. Desoky, "An implementation of the blowfish cryptosystem", *2008 IEEE International Symposium on Signal Processing and Information Technology*, 2008 ((https://doi.org/10.1109/ISSPIT.2008.4775664).

[2] K. Nowicki and T. Uhl, *Ethernet End-to-End*, 1st ed. Germany, Shaker-Publisher, 2008 (ISBN: 978383832271404).

[3] C. Shorten and T.M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019 (https://doi.org/10.1186/s40537-019-0197-0).

[4] S. Wong *et al.*, "Traffic forecasting using vehicle-to-vehicle communication", *3rd Annual Conference on Learning for Dynamics and Control*, pp. 917–929, 2021 (https://arxiv.org/pdf/2104.05528).

Submission

The paper with full PDF version and anonymous PDF version for the blind review process should be submitted on the JTIT website https://www.jtit.pl/jtit/about/submissions.

Reviewing Process

The article is initially approved by the Editor-In-Chief and if the decision is positive, is then sent to the reviewers. Depending on the subject of the article, it takes few weeks. In the next step, reviews are showed to authors who have 2 weeks to correct the article. Finally, the corrected text can be re-presented to the reviewer for reevaluation, which will take another 2 weeks.

As a result, after about 3 months, we are able to send the text for publication in the upcoming issue of JTIT.

When the reviews are inconsistent, additional corrections are necessary, or the reviewer expects additional verification because the corrections ordered by the author are insufficient or additional problems arise, the review of the article may be extended by another month or more.

Editorial Work

Positively reviewed and corrected article is next prepared by the editorial office for publication. At the end of this process the author receives an copyedited version for approval.

Licensing

Manuscript submitted to JTIT should not be published or simultaneously submitted for publication elsewhere. By submitting a manuscript the author grants license to the National Institute of Telecommunications, for the use of the paper in the fields of exploitation: reproducing and fixing the paper, distributing the paper by means of introduction to trade, letting for use or rental of the original or copies, and distributing the paper by means of public exhibition, screening, presentation and broadcast as well as rebroadcast, and making the paper publicly available in such a manner that anyone could access it at a place and time selected thereby, or by making it available in a way not allowing selection of time or place, including by means of Internet or other networks.

Ghostwriting Declaration

We require formal declaration that the process of writing the paper was not influenced by any third party. In the article, all the contributions of other people are clearly indicated. The theories presented, methods used, analysis and research, as well as the copyrights to the drawings, photographs and other figures belong to the authors or are clearly credited in the text. The author must also indicate whether his work has received financial support and if the realization of the whole project was possible thanks to the permission and cooperation with scientific institutions, associations and others.

Other Information

- The JTIT being an Open Access Journal (OAJ) has no article processing charges (APCs). The published articles can be downloaded freely without payment.
- JTIT supports open access and using continuous publishing "publish-as-you-go" scheme. This means that we no longer wait to accumulate several articles into a quarterly issue before publication. Rather, articles are continuously added to current issues after acceptation. Publish-as-you-go reduces publication lag for our authors, and make the newest research available quickly. After completing the review process, an article is published online in the current issue with DOI registration. When the issue period ends, a new issue is activated. So accepted articles are published without waiting for the quarterly issue end.

Harmonic Trust Spectrum Modeling for Cognitive Radio Intrusion Detection

S. Jeyaseelan, P.G.S. Velmurugan, G. Prabhakar, and J. Shanthi 86

Two-way Planar Array Radar with Diamond- and X-shaped Configurations for Sidelobe Suppression

J.R. Mohammed 95

An Adaptive Threshold Tracking Area Update and Selective Probabilistic Paging Scheme for Signaling Cost Reduction in 5G NR Networks

K. Popat, D. Meva, and H.K. Molia 101

Explainable Network Traffic Classification Using XGBoost and SHAP: Interpreting Feature Contributions Across Traffic Classes

B.S. Deghem, L. Aid, and K. Mostefaoui 114

National Institute of Telecommunications

Editorial Office

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland
https://www.gov.pl/web/instytut-lacznosci

phone +48 22 512 81 83
fax +48 22 512 84 00

e-mail: journal@jtit.pl
www.jtit.pl

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 3/2026 vol.105