INSTITUTO
UNIVERSITÁRIO
DE LISBOA

AI based Cybersecurity enhancement in 5G networks

Hamed Farkhari

PhD in Information Science and Technology with specialization in Telecommunications

Supervisors:
Pedro Joaquim Amaro Sebastião
Associate Professor with Aggregation
Instituto Universitário de Lisboa (ISCTE)

Luis Miguel Serra da Costa Campos
Head of research - PDMFC

December, 2024

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

AI based Cybersecurity enhancement in 5G networks

Hamed Farkhari

PhD in Information Science and Technology with specialization in Telecommunications

Supervisors:
Pedro Joaquim Amaro Sebastião
Associate Professor with Aggregation
Instituto Universitário de Lisboa (ISCTE)

Luis Miguel Serra da Costa Campos
Head of research - PDMFC

December, 2024

TECNOLOGIAS
E ARQUITETURA

Department of Information Science and Technology

AI based Cybersecurity enhancement in 5G networks

Hamed Farkhari

PhD in Information Science and Technology with specialization in Telecommunications

Jury:
Prof. Ana Garcia Armada, Full Professor, La Universidad Carlos III de Madrid (UC3M)
Prof. Fernando José da Silva Velez, Associate Professor, Universidade da Beira Interior
Prof. Américo Correia, Full Professor, Instituto Universitário de Lisboa (ISCTE)
Prof. Pedro Sebastião, Full Professor, Instituto Universitário de Lisboa (ISCTE)

President:
Doctor Vitor Manuel Basto-Fernandes, Associate Professor with Habilitation of Iscte - Instituto Universitário de Lisboa

December, 2024

To all people that helped me to write this thesis

Acknowledgment

This thesis not only marks the culmination of my academic endeavors, but also represents a significant milestone in my professional journey. Its successful completion would not have been possible without the support, guidance, and contributions of numerous individuals and organizations for whom I am profoundly grateful.

First and foremost, I extend my deepest gratitude to my supervisors, Professors Pedro Sebastião and Luís Campos, for their unwavering support, patience, motivation, and extensive knowledge. Their expert guidance has been invaluable throughout the course of my research and the preparation of this thesis.

This research was made possible through the generous funding of the European Union's Horizon 2020 Research and Innovation Programme under the Marie Skłodowska-Curie Project Number 813391. I am also sincerely grateful for the additional financial support provided by ISCTE and the Instituto de Telecomunicações.

I would like to express my heartfelt appreciation to Professors Rui Dinis, Victor P. Gil Jiménez, and Periklis Chatzimisios from the Instituto de Telecomunicações in Lisbon (Portugal), UC3M in Madrid (Spain), and International Hellenic University in Thessaloniki (Greece), respectively. Their invaluable assistance and technical expertise were instrumental to the success of this research.

Special thanks go to Sarang Kahvazadeh and Josep Mangues-Bafalluy for their exceptional support and guidance during my secondments at CTTC in Barcelona (Spain). Their insights and contributions significantly enriched this work.

I am also very grateful to the technical and administrative staff at PDMFC in Lisbon (Portugal) for their constant assistance and support. I give particular thanks to PDMFC for their help with experimental setups and for providing access to computational resources and also supporting during the three years of this work which were essential to my research.

I am grateful to TeamUp5G, a European Training Network (ETN) under the Marie Skłodowska-Curie Innovative Training Networks (MSCA ITN), for creating an exceptional platform that allowed researchers from diverse nations to connect, collaborate, and exchange knowledge. The opportunity to work on such an international project alongside PDMFC and ISCTE in Portugal; UC3M, CTTC, and Nokia in Spain; iS-Wireless in Poland; the International Hellenic University in Greece; and other partners from Norway, Denmark, and Switzerland, under the expert management of Professor Ana García Armada, has been an invaluable experience.

To my family, I am in immense debt for their wisdom, guidance, and unwavering support. My heartfelt thanks go to my parents for their unconditional love and encouragement throughout my academic journey. Their belief in me has been the foundation of my achievements.

Finally, I would like to acknowledge colleagues, especially Joseanne Viana for her collaboration in many publications and friends whose companionship and joyful moments provided much needed balance and respite from the demands of research. Their support has been a source of strength and happiness.

Hamed Farkhari
January 2020–December 2024

Resumo

Num cenário interconectado atual, a cibersegurança representa um pilar indispensável para o avanço tecnológico, enquanto as ameaças cibernéticas evoluem na mesma velocidade que os sistemas que visam comprometer. O surgimento de vetores de ataque cada vez mais sofisticados desafia os paradigmas tradicionais de segurança, demandando abordagens inovadoras para detecção e mitigação de ameaças. Dentre essas ameaças, destacam-se os ataques direcionados às comunicações sem fio, capazes de comprometer infraestruturas críticas e serviços essenciais em diversos setores.

Com o aumento da complexidade das redes de comunicação, a *Inteligência Artificial* (IA) e o aprendizado de máquina emergem como ferramentas essenciais para a análise de segurança e a identificação de ameaças. Essas tecnologias permitem a implementação de mecanismos de monitorização e resposta adaptativa em tempo real, fundamentais para a proteção de sistemas sem fio modernos. Contudo, a implementação de medidas de segurança eficazes, sem comprometer a eficiência operacional, apresenta desafios significativos, especialmente em ambientes com recursos limitados.

A integração da tecnologia quinta geração de comunicações móveis (5G) em *Veículos Aéreos Não Tripulados* (VANTs) exemplifica esses desafios, ampliando as capacidades dessas plataformas por meio de comunicações mais rápidas, baixa latência e elevada fiabilidade. Apesar dessas vantagens, a dependência de comunicações sem fio avançadas torna os VANTs vulneráveis a ataques de interferência (jamming), uma ameaça crítica que pode comprometer suas operações. A interferência ocorre quando sinais são emitidos com o objetivo de bloquear ou degradar os links de controlo e dados dos VANTs.

Nas aplicações dos VANTs, como vigilância, entrega de bens e gestão de desastres, a interferência pode acarretar problemas significativos, incluindo perda de controlo, falha na execução de missões e comprometimento da integridade dos dados. Essas ameaças são particularmente críticas em setores como defesa e segurança pública, nos quais os VANTs desempenham papéis estratégicos.

A identificação de interferências em VANTs é especialmente desafiadora devido à mobilidade dessas plataformas, à dinãmica dos ambientes e à complexidade do espectro de alta frequência associado ao 5G. Agentes mal-intencionados podem utilizar técnicas avançadas, como jamming inteligente e spoofing, para explorar frequências ou canais de comunicação específicos, dificultando ainda mais a detecção.

Métodos eficazes de identificação de interferências dependem, em grande medida, de abordagens baseadas em IA, que incluem análise espectral em tempo real e detecção avançada de anomalias. A IA possibilita a análise de grandes volumes de dados de rede

para identificar padrões que indiquem interferências, mesmo quando estas são subtis ou adaptativas. Utilizando modelos de aprendizagem de máquina, sistemas de VANTs podem classificar e prever, em tempo real, ameaças de potenciais interferências.

A interferência em VANTs tem o potencial de comprometer missões e gerar riscos de segurança significativos. Assim, estratégias proativas de detecção e mitigação são indispensáveis para proteger as operações de VANTs e manter a confiança nas aplicações suportadas pela tecnologia 5G. A possibilidade de garantir a resiliência contra interferência, não apenas salvaguarda os VANTs, mas também fomenta a adoção mais ampla do 5G em sistemas críticos, promovendo um avanço seguro e sustentável na era das comunicações sem fio.

Abstract

In today's interconnected landscape, cybersecurity is an indispensable pillar of technological advancement, while cyber threats evolve at a pace equivalent to the systems they are trying to compromise. The emergence of increasingly sophisticated attack vectors challenges traditional security paradigms, requiring innovative approaches to detect and mitigate threats. Among these threats, attacks targeting wireless communications are particularly concerning, as they have the potential to compromise critical infrastructures and essential services across various sectors.

As communication networks become more complex, Artificial Intelligence (AI) and Machine Learning (ML) have emerged as essential tools for security analysis and threat identification. These technologies enable the implementation of real-time monitoring and adaptive response mechanisms, which are crucial to protecting modern wireless systems. However, the implementation of effective security measures without compromising operational efficiency presents significant challenges, particularly in resource-constrained environments.

The integration of fifth Generation (5G) wireless technology into Unmanned Aerial Vehicles (UAVs) exemplifies these challenges, improving the capabilities of these platforms through faster communication, low latency, and high reliability. Despite these advantages, reliance on advanced wireless communications makes UAVs vulnerable to jamming attacks, a critical threat that can compromise their operations. Jamming occurs when signals are emitted to block or degrade the control and data links of UAVs.

In UAV applications such as surveillance, goods delivery, and disaster management, jamming can cause significant issues, including loss of control, failure to complete missions, and compromise of data integrity. These threats are especially critical in sectors such as defense and public security, where UAVs play a strategic role.

Detecting jamming in UAVs is particularly challenging due to their mobility, dynamic environments, and the complexity of the high-frequency spectrum associated with 5G. Malicious actors may employ advanced techniques, such as intelligent spoofing and jamming, to exploit specific communication frequencies or channels, further complicating detection efforts.

Effective jamming identification methods are heavily based on artificial intelligence-based approaches, including real-time spectral analysis and advanced anomaly detection. AI facilitates the analysis of large volumes of network data to identify patterns indicative of jamming, even when the interference is subtle or adaptive. By leveraging ML models, UAV systems can classify and predict potential jamming threats in real time.

Jamming into UAVs has the potential to compromise missions and pose significant security risks. Therefore, proactive detection and mitigation strategies are essential to protect UAV operations and maintain confidence in 5G-enabled applications. Ensuring resilience against jamming not only protects UAVs but also promotes the broader adoption of 5G in critical systems, fostering safe and sustainable progress in the era of wireless communications.

Contents

Acknowledgment iii

Resumo v

Abstract vii

List of Acronyms 1

Chapter 1. Introduction 3
1.1. Motivation 5
1.1.1. Jamming Threats in UAV Communications 5
1.1.2. Modern Detection Approaches 5
1.2. The Role of 5G Networks 7
1.3. Goals 8
1.4. Contributions 9
1.5. List of Publications 10
1.6. Thesis Organization 11

Chapter 2. Papers 15
2.1. Article #1: PCA-Featured Transformer for Jamming Detection in 5G UAV Networks 15
Article Details 15
2.2. Article #2: A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms 34
Article Details 34
2.3. Article #3: Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation 46
Article Details 46
2.4. Article #4: A Convolutional Attention-Based Deep Learning Solution for 5G UAV Network Attack Recognition 63
Key Contributions 63
Performance Evaluation 63
Experimental Validation 63
Article Details 63
2.5. Article #5: Two Methods for Jamming Identification in Unmanned Aerial Vehicles (UAV) Networks Using a New Synthetic Dataset 70
Key Contributions 70

Statistical Approach 70
Deep Learning Approach 70
Experimental Validation 70
Significance 70
Article Details 71
2.6. Article #6: Latent Space Transformers for Generalizing Deep Networks 78
Key Contributions 78
Proposed Approach 78
Applications and Benefits 78
Article Details 78
2.7. Article #7: New PCA-based Category Encoder for Efficient Data Processing in IoT Devices 85
Key Contributions 85
Experimental Validation 85
Article Details 85
2.8. Article #8: MOOC on "Ultra-dense Networks for 5G and its Evolution": Challenges and Lessons Learned 92
Key Aspects 92
MOOC Development Process 92
Course Content 92
Article Details 92

Chapter 3. Conclusions 99

References 101

List of Acronyms

5G	fifth Generation
AI	Artificial Intelligence
CNN	Convolutional Neural Networks
DAtR	Deep Attention Recognition
DL	Deep Learning
DNN	Deep Neural Networks
IA	*Inteligência Artificial*
IoT	Internet of Things
LoS	Line-of-Sight
LSTM	Long Short-Term Memory
ML	Machine Learning
MVA	Majority Voting Algorithm
MA	Mean Accuracy
MC	Mean Confidence
MOOC	Massive Open Online Course
NLoS	Non-Line-of-Sight
PCA	Principal Component Analysis
QoS	Quality of Service
RSSI	Received Signal Strength Indicator
RS	Reliability Score
STL	Seasonal Trend Decomposition
SINR	Signal-to-Interference-plus-Noise Ratio
SVM	Support Vector Machines
TSA	Time Series Augmentation
UAV	Unmanned Aerial Vehicles
VANTs	*Veículos Aéreos Não Tripulados*
XGB	eXtreme Gradient Boosting

CHAPTER 1

Introduction

The rapid evolution of wireless communications and the increasing prevalence of interconnected devices have fundamentally reshaped how systems interact, making cybersecurity a critical concern across all technological domains [**1**], [**2**], [**3**]. As networks grow in complexity and interconnectedness, the attack surface available to malicious actors continues to expand, introducing new vulnerabilities and security challenges. Traditional security paradigms, such as perimeter-based defenses, are no longer sufficient in the modern dynamic threat landscape, where sophisticated attacks often bypass conventional protective measures [**4**],[**2**],[**5**].

ML [**6**], [**7**] has emerged as a powerful tool for addressing contemporary security challenges, offering the capability to detect and respond to threats in real-time [**8**]. These approaches enable the identification of patterns and anomalies that are difficult or impossible to detect using traditional rule-based systems, providing a more adaptive and robust security framework [**8**]. However, the application of ML to security contexts introduces its own set of challenges, particularly in resource-constrained environments where computational efficiency is paramount.

The integration of Artificial Intelligence (AI) with security has led to an ongoing arms race between attackers and defenders, with both sides leveraging increasingly sophisticated techniques [**9**]. This competition has driven the development of innovative [**10**] methodologies for threat detection, classification, and response, which can adapt to evolving attack strategies [**11**], [**12**]. In particular, attacks targeting the availability and reliability of communications, such as jamming and interference [**13**], pose significant threats to system performance and safety, [**14**], [**15**].

To address these pressing challenges, this research proposes several complementary approaches. One such approach is a categorical encoder based on Principal Component Analysis (PCA) [**16**] that enables efficient processing of high-dimensional data while preserving critical information for security analysis. This method is particularly valuable in resource-constrained Internet of Things (IoT) and UAV systems, where real-time threat detection is crucial.

Another contribution of this research is the development of latent space transformers, which provide a novel framework for information sharing between deep networks. This approach improves interoperability, optimizes computational resources, and maintains robust security capabilities across distributed systems. Within the broader security landscape, the integration of UAV into 5G networks introduces unique challenges [**17**]. UAVs, increasingly used for applications such as emergency response, package delivery,

and surveillance, require secure communication to ensure reliable operation [**18**], [**19**],[**20**], [**21**], [**22**]. Among the security threats facing UAV communications, jamming attacks are particularly concerning [**11**]. These attacks range from basic power jamming, which aims to overwhelm legitimate signals, to advanced smart jamming techniques that employ frequency targeting and adaptive power allocation. Furthermore, mobility-based attacks, where attackers optimize their position to maximize interference, add another layer of complexity [**23**] .

The evolution of jamming detection techniques has paralleled the sophistication of attack methodologies. Modern detection methods leverage multiple technologies, including ML and statistical analysis [**24**], [**25**], [**26**], [**27**]. Techniques such as Deep Neural Networks (DNN)s, Convolutional Neural Networks (CNN)s, and Support Vector Machines (SVM)s have shown promise in identifying and classifying jamming attacks. These ML models are often complemented by statistical methods [**28**], [**29**], such as time series decomposition and pattern recognition, to create robust hybrid detection systems[**30**], [**31**].

For specific detection tasks, this research introduces a deep attention recognition architecture that analyzes network parameters to identify jamming attempts in UAV communications. Using metrics such as Signal-to-Interference-plus-Noise Ratio (SINR) and Received Signal Strength Indicator (RSSI), this approach achieves reliable detection across diverse channel conditions common in UAV operations.

The research culminates in a hybrid approach that combines deep neural networks with traditional ML techniques. This unified framework addresses uncertainty while maintaining high detection accuracy, offering a practical solution for real-world deployment in UAV security applications. Together, these methodologies form a comprehensive framework for securing UAV communications against jamming attacks on 5G networks. The proposed solutions strike a balance between security performance and practical constraints, making them suitable for implementation in resource-limited environments while ensuring the rapid response times required for UAV operations.

This research contributes significantly to the field of UAV security by demonstrating how advanced ML techniques can be effectively adapted for resource-constrained environments. Highlights the potential of these methods to maintain high detection accuracy and reliability, ultimately enhancing the security and resilience of UAV communications in 5G networks.

1.1. Motivation

The growing prevalence of jamming threats in UAV communications necessitates robust and adaptive detection methodologies to ensure the security and reliability of these systems. This section explores two critical aspects: the evolving landscape of jamming threats, including power, smart, and mobility-based attacks, and the modern detection approaches designed to counter them. By addressing both the challenges posed by sophisticated jamming techniques and the advancements in detection technologies such as ML and hybrid systems, this discussion highlights the importance of comprehensive frameworks to safeguard UAV operations against these escalating threats.

1.1.1. Jamming Threats in UAV Communications

Among the various security threats facing UAV communications, jamming attacks represent a particularly critical challenge. These attacks manifest themselves in multiple sophisticated forms, each presenting unique challenges to detection and mitigation [**27**], [**32**].

Power jamming represents the most straightforward approach, where attackers attempt to overwhelm legitimate signals through brute force. This technique focuses on degrading communication quality and disrupting control links between UAVs and their ground stations, potentially leading to complete loss of control over the aerial vehicle.

Smart jamming has emerged as a more sophisticated threat, employing intelligent techniques to maximize impact while minimizing probability detection. These attacks utilize selective frequency targeting and adaptive power allocation to efficiently interfere with communications. Pattern-based interference techniques allow attackers to synchronize their jamming activities with legitimate transmissions, making them particularly difficult to identify and counter using traditional detection methods [**33**], [**31**], [**29**] .

Mobility-based attacks represent the latest evolution in jamming threats, taking advantage of the dynamic nature of UAV operations. These attacks involve sophisticated positioning strategies where attackers actively track and follow their targets, optimizing their jamming effectiveness through strategic placement. The emergence of coordinated multi-attacker scenarios has further complicated the defense landscape, as multiple jammers can work in concert to create more effective and harder-to-detect interference patterns [**34**], [**35**].

1.1.2. Modern Detection Approaches

The evolution of jamming detection techniques has necessarily paralleled the advancement of attack methodologies. Contemporary approaches integrate multiple technologies, creating layered defense systems capable of identifying and responding to various attack types. ML solutions have emerged as a powerful tool in this domain, with DNNs, CNNs, and SVMs demonstrating remarkable success in identifying subtle attack patterns that

might elude traditional detection methods. These ML approaches are particularly effective in scenarios where attack signatures evolve rapidly, as they can be trained on new data to recognize emerging threat patterns[**24**], [**25**], [**26**], [**27**].

Statistical analysis continues to play a crucial role in jamming detection, offering robust and interpretable results through time series decomposition and pattern recognition techniques. These methods excel at identifying anomalies in communication patterns that may indicate ongoing attacks. Statistical approaches provide the advantage of clear confidence intervals and significance levels, making them particularly valuable in scenarios where decision-making must be thoroughly justified. Time series analysis, in particular, has proven effective in detecting periodic jamming attempts and understanding the temporal characteristics of such attacks[**28**], [**29**].

The combination of statistical approaches with ML has led to the development of powerful hybrid methodologies that leverage the strengths of both approaches. These hybrid systems have demonstrated superior performance compared to single-methodology approaches, as they can simultaneously utilize the pattern recognition capabilities of ML and the statistical rigor of traditional analysis. The fusion of these methodologies allows for more nuanced detection capabilities, where ML models identify complex patterns, while statistical analysis provides verification and validation of the findings.

Hybrid systems typically employ multi-layer detection architectures, combining traditional statistical analysis with advanced ML techniques. This layered approach begins with basic statistical measures to identify potential anomalies, followed by more sophisticated ML analysis to classify and characterize detected threats. The multi-layer architecture allows for progressive refinement of detection accuracy, with each layer contributing additional insights into the nature and severity of potential jamming attempts. This approach has proven particularly effective in reducing false positives while maintaining high detection rates [**30**], [**31**].

Adaptive threshold techniques enable these systems to maintain effectiveness across varying operating conditions by automatically adjusting detection parameters based on environmental factors and observed communication patterns. This adaptability is crucial in real-world deployments, where network conditions, interference levels, and legitimate usage patterns may fluctuate significantly. Threshold adjustment mechanisms typically incorporate both short-term and long-term historical data to establish appropriate baseline conditions, ensuring that detection sensitivity remains optimized for current operating conditions.

The integration of these various components creates a comprehensive detection framework capable of responding to both known and novel jamming attacks. The system's ability to combine multiple detection methodologies, adapt to changing conditions, and maintain high accuracy across different scenarios represents a significant advancement in jamming detection capabilities. This integrated approach enables reliable detection of

sophisticated attacks while retaining the ability to identify and respond to simpler, traditional jamming attempts. The framework's flexibility and adaptability ensure its continued effectiveness as new attack methodologies emerge and network conditions evolve.

1.2. The Role of 5G Networks

The integration of UAVs into 5G networks introduces both novel opportunities and complex security challenges. This dynamic interaction between UAV operations and 5G infrastructure necessitates careful consideration of vulnerabilities and defensive capabilities during the design and implementation of these systems [**36**], [**37**].

Modern 5G network architectures, characterized by dense small-cell deployments and advanced beam forming technologies, create a multifaceted operational environment. While the dense deployment of small cells enhances granular coverage and control, it simultaneously increases the number of potential attack vectors. Similarly, advanced beam forming techniques, which focus energy in specific directions to mitigate interference and counter jamming attempts, demand robust security protocols to safeguard the integrity of these mechanisms [**38**].

Performance requirements in 5G networks impose stringent constraints on security solutions, necessitating ultra-reliable communications with minimal latency. UAV applications, particularly those involving critical operations, often require real-time response capabilities with latency as low as 1 millisecond. Consequently, security protocols must ensure robust protection while adhering to these temporal constraints. This has driven the development of lightweight yet effective security measures optimized for low-latency environments [**39**].

Balancing high throughput with minimal security overhead is another critical consideration. Modern UAV applications, such as those involving high-definition video streaming or real-time sensor data transmission, generate substantial data volumes. Security mechanisms must efficiently process these high data rates without imposing significant additional overhead, which could compromise overall system performance. To address this, novel encryption methods have been developed to operate at line speed while maintaining strong security guarantees [**38**].

The integration of 5G networks also expands the scope of security considerations beyond traditional concerns. Enhanced authentication mechanisms and comprehensive interference management systems are now essential components of 5G-enabled UAV operations. These systems must dynamically adapt to evolving threat landscapes and operational demands [**40**] [**41**].

Quality of Service (QoS) guarantees remain critical, even in the presence of active jamming attempts. Achieving this requires seamless coordination between security systems and network management functions. Real-time monitoring and response mechanisms are essential to detect and mitigate interference while preserving essential communication links. The incorporation of AI and ML algorithms has significantly improved these capabilities, enabling accurate threat detection and automated responses to emerging threats.

In summary, the interplay between UAVs and 5G networks necessitates innovative security solutions [**40**] capable of meeting the stringent performance requirements and dynamic threat landscapes inherent in modern network environments. Continued advancements in AI-driven analytics, low-latency protocols, and high-throughput encryption technologies are pivotal to ensuring the secure and reliable integration of UAVs into 5G ecosystems.

1.3. Goals

The primary goal of this research is to advance data processing and resource optimization in UAV security systems. This work aims to develop highly efficient categorical data encoding techniques specifically tailored for resource-constrained IoT devices and UAV platforms. By employing innovative PCA-based methods, the research seeks to achieve significant dimensionality reduction while maintaining exceptional classification accuracy. A critical objective is to optimize these processes for real-time operations, enabling UAVs to perform complex security classifications with minimal computational overhead. This foundational goal addresses the core challenge of balancing processing efficiency with security effectiveness in aerial systems.

Another crucial objective is network architecture standardization. This involves establishing standardized frameworks to enable seamless interoperability among deep networks within 5G communication systems. The research focuses on creating modular network components with standardized latent spaces, thereby facilitating efficient information sharing and significantly reducing latency and bandwidth consumption. The scope of this objective extends beyond technical standardization to include practical implementation considerations, ensuring that these frameworks are deployable across diverse UAV communication systems.

In the domain of jamming detection and classification, this research pursues a sophisticated multi-layered approach. The goal is to develop advanced systems capable of accurately distinguishing between intentional jamming attacks and environmental interference. This entails creating robust detection methods that combine traditional statistical analysis with cutting-edge Deep Learning (DL) techniques. The objective is to achieve unprecedented accuracy in identifying various jamming scenarios while maintaining adaptability to evolving threat landscapes. Special focus is placed on designing systems capable of countering sophisticated jamming techniques aimed at evading detection.

Energy efficiency is another critical goal within the research framework. Given the inherent battery limitations of UAV platforms, the study emphasizes developing detection systems that deliver high accuracy with minimal power consumption. This involves designing lightweight algorithms that optimize computational efficiency without compromising detection capabilities. The research aims to strike an optimal balance between performance requirements and operational constraints, ensuring that security measures do not significantly impact flight duration or operational range.

Integration and standardization form additional cornerstones of the research objectives. The study seeks to establish comprehensive evaluation frameworks and performance metrics to enable systematic comparisons of different detection approaches. This includes the development of standardized testing methodologies and benchmarks for assessing real-world effectiveness. These standards aim to accelerate the development and adoption of improved solutions across the industry while ensuring reliable performance measurements under diverse operational conditions.

The development of autonomous operational capabilities is a forward-looking objective of this research. The aim is to create self-adaptive detection systems capable of automatically adjusting their parameters based on operational conditions and threat levels. This involves designing systems that seamlessly integrate with UAV navigation and control mechanisms, enabling autonomous responses to detected threats while maintaining mission objectives within safety constraints. Furthermore, the research focuses on developing intelligent systems that learn and adapt from experience, continuously enhancing their detection capabilities.

Lastly, a comprehensive framework development goal drives the integration of multiple detection approaches into cohesive systems. This involves implementing sophisticated pre-processing and post-processing techniques to enhance reliability across various operational scenarios. The research seeks to develop real-time processing techniques capable of handling complex threat scenarios while maintaining system stability. The framework is designed to be flexible enough to incorporate new detection methods while maintaining backward compatibility with existing systems.

1.4. Contributions

This work presents several significant contributions to the field of UAV communication security and jamming detection, advancing both theoretical understanding and practical implementation capabilities:

- **Novel Deep Learning Architecture**

  We introduce Deep Attention Recognition (DAtR) framework, representing a fundamental advancement in jamming detection for UAV communications. This architecture uniquely combines Convolutional neural networks with self-attention mechanisms, enabling more effective processing of temporal patterns in communication signals. Our approach reduces the total number of trainable parameters to under 100,000 while maintaining high detection accuracy, making it suitable for deployment on resource-constrained UAV platforms. The architecture demonstrates superior performance, particularly in complex urban environments, achieving 90.80% accuracy in Line-of-Sight (LoS) conditions and 83.07% accuracy in Non-Line-of-Sight (NLoS) scenarios.
- **Advanced Preprocessing and Post-processing Techniques**

  We develop two complementary techniques that significantly enhance detection capabilities:

  - **Time Series Augmentation (TSA):** This novel preprocessing approach enhances signal data by generating augmented versions of the original signal while preserving critical jamming signatures.
  - **Majority Voting Algorithm (MVA):** A robust post-processing framework that reduces false positives while maintaining high detection sensitivity.

  When combined, these methods improve overall detection accuracy by 15% in challenging NLoS conditions compared to baseline approaches.

- **Comprehensive Testing Framework**

  Our work establishes a rigorous evaluation methodology for jamming detection systems, incorporating multiple channel conditions, attacker configurations, and environmental scenarios. We develop a synthetic dataset comprising over 2,400 distinct test cases, covering various jamming attack patterns and UAV operational scenarios. This testing framework enables systematic comparison of different detection approaches and provides benchmarks for future research in this field.

- **Performance Analysis and Metrics**

  We propose new metrics for evaluating jamming detection performance, including the *Reliability Score (RS)*, which quantifies the relationship between detection confidence and accuracy. Our analysis provides detailed insights into system behavior across different operational scenarios, establishing new benchmarks for performance evaluation in UAV security systems. The work includes comprehensive statistical analysis of detection performance under varying channel conditions and attack patterns.

1.5. List of Publications

1. H. Farkhari et al., "PCA-Featured Transformer for Jamming Detection in UAVs," in *Proc. IEEE Int. Conf. Machine Learning and Computer Networks (ICMLCN)*, 2025, Submitted.

2. H. Farkhari et al., "A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms," in *IEEE Access*, doi: 10.1109/ACCESS.2024.3504729.

3. H. Farkhari et al., "Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation," in *IEEE Transactions on Vehicular Technology*, vol. 73, no. 1, pp. 131–146, Jan. 2024, doi: 10.1109/TVT.2023.3302814.

4. H. Farkhari et al., "A Convolutional Attention Based Deep Learning Solution for 5G UAV Network Attack Recognition over Fading Channels and Interference," in

2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall), London, United Kingdom, 2022, pp. 1–5, doi: 10.1109/VTC2022-Fall57202.2022.10012726.

5. H. Farkhari et al., "Two methods for Jamming Identification in UAV Networks using New Synthetic Dataset," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, Helsinki, Finland, 2022, pp. 1–6, doi: 10.1109/VTC2022-Spring54318.2022.9860816.

6. H. Farkhari et al., "Accurate and Reliable Methods for 5G UAV Jamming Identification With Calibrated Uncertainty," in *2023 RCIS*.

7. H. Farkhari et al., "A Synthetic Dataset for 5G UAV Attacks Based on Observable Network Parameters," *arXiv e-prints*, 2022, doi:10.48550/arXiv.2211.09706.

8. H. Farkhari et al., "New PCA-based Category Encoder for Efficient Data Processing in IoT Devices," in *2022 IEEE Globecom Workshops (GC Wkshps)*, Rio de Janeiro, Brazil, 2022, pp. 789–795, doi: 10.1109/GCWkshps56602.2022.10008757.

9. H. Farkhari et al., "Latent Space Transformers for Generalizing Deep Networks," in *2021 IEEE Conference on Standards for Communications and Networking (CSCN)*, Thessaloniki, Greece, 2021, pp. 130–135, doi: 10.1109/CSCN53733.2021.9686099.

10. M. J. Lopez-Morales et al., "MOOC on 'Ultra-dense Networks for 5G and its Evolution': Challenges and Lessons Learned," in *2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE)*, Coimbra, Portugal, 2022, pp. 1–6, doi: 10.1109/EAEEIE54893.2022.9819989.

1.6. Thesis Organization

This thesis begins with a comprehensive examination of the challenges associated with categorical data processing in IoT device security systems. A key innovation is introduced through the development of a PCA-based Category Encoder, which significantly reduces computational overhead while maintaining high classification accuracy. This novel encoding approach demonstrates exceptional efficiency in handling high-cardinality categorical variables, achieving up to an 98.81% reduction in dimensionality compared to traditional one hot encoding method. Extensive experimental validation using the NSLKDD dataset highlights how this method enables resource-constrained IoT devices to perform complex security classifications with minimal computational resources while maintaining accuracy rates exceeding 89%.

The research advances to address fundamental challenges in deep network interoperability through the introduction of an innovative framework for latent space DNNs. This groundbreaking framework establishes a standardized approach for information sharing between deep networks in 5G communication systems, enabling seamless integration of

diverse network architectures. The framework demonstrates how complex deep networks can be modularized into components with standardized latent spaces, facilitating efficient information exchange while reducing computational demands.

In the context of specific security applications, the thesis presents two complementary methods for jamming identification in UAVs. The first approach uses advanced time series analysis via Seasonal Trend Decomposition (STL), achieving an 84.38% detection accuracy for high-power jamming attacks. The second method employs a meticulously designed (CNN-Long Short-Term Memory (LSTM)) architecture, which achieves 99.99% accuracy in identifying various jamming scenarios. These methods are validated using a comprehensive synthetic dataset that simulates realistic UAV communication scenarios, encompassing a range of jamming powers, distances, and channel conditions.

The research further extends its technical contributions through the development of a Convolutional attention-based deep learning architecture. This solution specifically addresses the challenges posed by dynamic channel conditions and sophisticated jamming attacks in 5G environments. By incorporating innovative attention mechanisms, the architecture enables precise feature extraction from complex signal patterns, demonstrating significant improvements in detection accuracy under varying channel conditions. The system achieves consistent performance even in challenging scenarios involving multiple interferers and dynamic channel characteristics.

Building upon these foundations, the thesis introduces DAtR framework optimized for attack identification in UAV communications. This framework integrates advanced attention mechanisms with efficient neural network architectures, enabling real-time attack detection while maintaining low computational overhead. The system demonstrates robust performance across diverse operational scenarios, maintaining high accuracy even under adverse conditions, such as low signal-to-noise ratios and multiple simultaneous attacks.

A hybrid approach to jamming identification is also presented, combining traditional ML with advanced deep learning methods. This integrated solution leverages the strengths of multiple detection approaches, including statistical analysis and signal processing techniques. The hybrid system exhibits superior reliability across diverse operational scenarios, achieving consistent detection rates above 95% while maintaining false alarm rates below 1%, even under challenging environmental conditions.

Throughout the thesis, a strong emphasis is placed on practical implementation considerations, particularly with respect to the unique constraints of UAV platforms and 5G network environments. Each technological advancement is meticulously evaluated in terms of computational requirements, power efficiency, and real-world applicability. Detailed analyses of implementation trade-offs, including processing latency, resource utilization, and detection accuracy, are provided for various operational conditions.

The research concludes with the integration of these components into a comprehensive security framework for UAV communications. This synthesis demonstrates how the

combination of efficient data processing, standardized network architectures, and sophisticated detection methods creates a robust and practical solution. The integrated system achieves superior performance in real-world scenarios, with validation results indicating sustained high accuracy across different attack types, channel conditions, and operational scenarios. This framework lays a solid foundation for future advancements in secure UAV communications while ensuring practical applicability in current network environments.

CHAPTER 2

Papers

2.1. Article #1: PCA-Featured Transformer for Jamming Detection in 5G UAV Networks

This article presents a novel transformer-based approach to jamming detection in 5G UAV networks, incorporating PCA features with a U-shaped architecture. The research focused on developing a reliable detection framework capable of effectively identifying jamming attacks in both LoS and NLoS scenarios. The study introduced innovative feature engineering and training optimization techniques in time series data, and setting new benchmarks for jamming detection accuracy in UAV communications.

The primary contributions of this work to the present thesis are as follows:

- Development of the PCA-featured transformer architecture, integrating PCA features with transformer models in a novel manner;
- Introduction of fundamental deep neural network training techniques such as chunking for time series data and batch size scheduling, which significantly enhanced jamming detection performance;
- Achievement of 90.33% accuracy in LoS conditions and 84.35% in NLoS scenarios, surpassing conventional methods by approximately 4%.

The study demonstrated the effectiveness of these methods by applying them to realistic UAV network scenarios with varying numbers of attackers, power levels, and user densities. The proposed architecture was rigorously evaluated against baseline DNN and the XGBoost classifier, providing robust validation of its efficacy.

Article Details

- **Title:** PCA-Featured Transformer for Jamming Detection in 5G UAV Networks
- **Date:** 12-2024 (submission date)
- **Authors:** Joseanne Viana, Hamed Farkhari, Pedro Sebastião, Victor P. Gil Jimenez
- **Status:** Submitted
- **Journal:** IEEE Open Journal of The Communications Society
- **DOI:** -

The significance of this paper lies in its introduction of a comprehensive transformer-based approach addressing key challenges in 5G UAV jamming detection while maintaining practical implementation feasibility. The research established foundational concepts

in combining PCA features with transformer architectures, which are anticipated to influence future advancements in wireless security and signal processing for attack detection systems.

IEEE Open Journal of the Communications Society

Received XX Month, XXXX; revised XX Month, XXXX; accepted XX Month, XXXX; Date of publication XX Month, XXXX; date of current version 11 January, 2024.

Digital Object Identifier 10.1109/OJCOMS.2024.011100

PCA-Featured Transformer for Jamming Detection in 5G UAV Networks

Joseanne Viana[1,2] *IEEE Member*, **Hamed Farkhari**[3] *IEEE Member*, **Pedro Sebastião**[3,4] *IEEE Member*, **Victor P Gil Jimenez**[1] *IEEE Senior Member*

[1]UC3M - Universidad Carlos III de Madrid, Madrid, Spain;
[2]Tyndall National Institute, Ireland;
[3]ISCTE – Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026 Lisbon, Portugal;
[4]IT – Instituto de Telecomunicações, Av. Rovisco Pais, 1, Torre Norte, Piso 10, 1049-001 Lisboa, Portugal;

Corresponding author: Joseanne Viana (joseanne.viana@tyndall.ie)

This work was supported by the Project SOFIA-AIR (PID2023-147305OB-C31) (MICIU /10.13039/501100011033 / AEI / EFDR, UE)

ABSTRACT Unmanned Aerial Vehicles (UAVs) face significant security risks from jamming attacks, which can compromise network functionality. Traditional detection methods often fall short when confronting AI-powered jamming that dynamically modifies its behavior, while contemporary Machine Learning (ML) approaches frequently demand substantial feature engineering and struggle with temporal patterns in attack signatures. The vulnerability extends to 5G networks employing Time Division Duplex (TDD) or Frequency Division Duplex (FDD), where service quality may deteriorate due to deliberate interference. We introduce a novel U-shaped transformer architecture that leverages Principal Component Analysis (PCA) to refine feature representations for improved wireless security. The training process is regularized by incorporating the output entropy uncertainty into the loss function, a mechanism inspired by the Soft Actor-Critic (SAC) algorithm in Reinforcement Learning (RL) to enable robust jamming detection techniques. The architecture features a modified transformer encoder specially designed to process critical wireless signal features, including Received Signal Strength Indicator (RSSI) and Signal-to-Interference-plus-Noise Ratio (SINR) measurements. We complement this with a custom positional encoding mechanism that specifically accounts for the inherent periodicity of wireless signals, enabling a more accurate representation of temporal signal patterns. In addition, we propose a batch size scheduler and implement chunking techniques to optimize training convergence for time series data. These advancements contribute to achieving up to a ten times improvement in training speed within the advanced U-shaped encoder-decoder transformer model introduced in this study. Experimental evaluations demonstrate the effectiveness of our entropy-based approach, achieving detection rates of 89.46% under Line-of-Sight (LoS) conditions and 85.06% in non-Line-of-Sight (NLoS) scenarios. The method significantly outperforms existing solutions, surpassing XGBoost (XGB) classifiers by approximately 4.5% and other Deep Learning (DL) approach by more than 2%.

INDEX TERMS UAVs, Security, Transformers, Deep Learning, Jamming Detection, Jamming Identification, Unmanned Aerial Vehicles, 5G, 6G.

I. Introduction

The intersection of Unmanned Aerial Vehicles (UAVs) and wireless communication systems represents a rapidly evolving research domain with significant technological and security implications. As UAVs transition from specialized military applications to widespread commercial deployment, their integration into communication networks introduces novel challenges and opportunities [1], [2], [3], [4], [5], [6]. Base stations mounted on UAVs demonstrate potential for applications including emergency response, surveillance of borders, and providing temporary network coverage [7], [8],

[§]Collaborative authors with equal contribution

[9], [4]. However, when UAVs function as end-devices in services such as package delivery, they introduce distinct security vulnerabilities within 5G networks [10], [11]. A critical concern is the vulnerability of UAV communication systems to sophisticated jamming attacks, which can manipulate Time Division Duplex (TDD) and Frequency Division Duplex (FDD) systems, resulting in severe service disruptions with impacts reaching up to 99% in TDD uplink and 82% in FDD downlink scenarios [12], [13], [14]. Jamming can affect private networks, compromising the integrity and availability of mission critical communications in industrial, corporate, and specialized operational environments where UAVs are increasingly deployed.

Machine learning, particularly deep learning approaches, offers promising avenues for developing more effective and proactive jamming detection systems [15], [16], [17]. Convolutional Neural Networks (CNNs) have shown effectiveness in extracting spatial features from signal data [18], [19], [20], while Long Short-Term Memory (LSTM) networks excel at modeling temporal patterns in wireless communications [21], [22], [23].

Several studies have advanced this field. The authors in [24] developed a branched deep neural network architecture for simultaneous jamming detection and link scheduling in dense wireless networks. Their system employs two specialized subnetworks: one leverages geographical information and signal power measurements to detect and locate jammers, while the other optimizes link scheduling based on jamming detection results to maximize network throughput under adverse conditions. The authors of [25] proposed neural networks for compound jamming signal recognition, while [26] introduced a singular value decomposition approach for jamming identification in Global Navigation Satellite System (GNSS)-based systems.

Addressing cognitive radio vulnerabilities, [27] created a one-dimensional convolutional neural network operating directly on raw signal data to detect primary user emulation and jamming attacks. This approach eliminates the manual feature engineering required by traditional methods, as noted by [28]. Their architecture incorporates three convolutional layers with Rectified Linear Unit (ReLU) activation functions, followed by dense and softmax output layers.

More recently, [29] developed a deep learning system using ensemble techniques for detecting jamming attacks in 5G networks, combining RF domain and physical layer features with a Temporal Epistemic Decision Aggregator to enhance detection reliability despite signal impairments and carrier frequency offset. Similarly, [30] proposed a feature- and spectrogram-tailored machine learning approach for jamming detection in Orthogonal Frequency-Division Multiplexing (OFDM)-based UAVs.

Despite advances in CNN-based and LSTM-based jamming detection methods, these models exhibit fundamental architectural limitations that hinder their effectiveness in modern UAV scenarios. CNNs are inherently constrained by fixed-size receptive fields, limiting their ability to adaptively capture jamming patterns that occur over variable time scales—from millisecond-level pulse jamming to sustained interference lasting minutes. While LSTMs are designed for sequential modeling, they suffer from vanishing gradients in long sequences and require inherently sequential processing, which creates computational bottlenecks that are unsuitable for real-time UAV applications. These limitations reduce the effectiveness of both approaches in heterogeneous and dynamic RF environments, where sophisticated jamming may span irregular time intervals, exhibit non-stationary behavior, or fluctuate unpredictably in amplitude and frequency. In contrast, transformer architectures leverage self-attention mechanisms to dynamically model dependencies across entire input sequences, regardless of their temporal or spectral distance. Their parallel processing capability and adaptive attention allocation make transformers particularly well-suited for detecting evolving, intermittent, and multi-modal jamming attacks particularly in heterogeneous environments that incorporate both 5G New Radio (NR) and Narrowband Internet of Things (NB-IoT) interfaces [31], [32], [13], [33].

Transformer architectures, with their self-attention mechanisms [34], have improved sequential data analysis and offer significant potential for jamming detection [35]. These models are particularly well-suited for identifying long-range dependencies in wireless signal data, enabling the detection of sophisticated jamming patterns across various time scales and frequency bands [36]. The multi-head attention mechanism further enhances this capability by simultaneously analyzing diverse signal attributes, from immediate interference to subtle, persistent disruptions, making transformers especially effective against energy-efficient selective jamming techniques.

With the accelerating deployment of 5G networks and increasingly complex security threats [37], [38], there is a need for advanced jamming detection methods that can preemptively identify and counteract these attacks [39], [40]. Integrating transformers with other machine learning techniques offers promising new approaches for addressing these challenges, establishing foundations for more resilient network security frameworks [41], [42].

This paper introduces an innovative transformer-based architecture specifically designed for UAV-integrated 5G networks, focusing on early detection of jamming attacks, [13]. By incorporating Principal Component Analysis (PCA)-derived features [43], [44], our approach enables efficient analysis of critical signal metrics, including Received Signal Strength Indicator (RSSI) and Signal to Interference plus Noise Ratio (SINR) [45]. Multi-head attention mechanisms allow the model to identify and classify complex jamming patterns, while computational optimizations make the framework suitable for edge device deployment [46], [47]. This work not only advances the current state of jamming detection but also proposes techniques to improve training

time in machine learning integrated wireless communications [48], [49].

Furthermore, our proposed solution addresses gaps in existing research by prioritizing real-time detection capabilities [50], [51] through a transformers models. The architecture's adaptability allows it to function effectively across various deployment scenarios, ensuring broad applicability for diverse UAV applications in both civilian and defense contexts [52], [53], [54]. As UAV technology continues to expand [55], [56], ensuring communication system security and reliability remains essential. Our work makes a meaningful contribution toward this goal, providing a robust and scalable solution to one of the most pressing challenges in modern wireless communication.

Contributions and Motivation

Recent advancements in cellular networks, particularly in UAV and 5G technologies, have revealed significant vulnerabilities to jamming attacks. While research exists on various detection methods, there remains a critical gap in leveraging transformer architectures for jamming detection. This paper presents the listed key contributions:

- Developed an innovative transformer-based architecture for detecting jamming attacks in UAV-integrated 5G networks. The system features a custom deep neural network that combines state-of-the-art CNNs with specialized activation functions in a U-Net architecture, optimized for analyzing jamming signatures across 5G NR interfaces. Previous research consider deep network approaches using CNNs, LSTMs and attention layers in specialized designed architectures.
- Introduced time-series PCA-features and efficient tokenization method for detecting jamming patterns in UAV-integrated 5G networks.
- Proposed incorporating the output entropy uncertainty into the loss function.
- Optimized deep network training algorithm by introducing batch_size scheduler and chunking (grouping) in the training dataset.
- Comprehensive experimental validation demonstrating superior detection capabilities compared to existing methods deep learning and machine learning methods for jamming attacks.

II. System Model

The system model consists of an authenticated UAV operating within a small cell network environment designed to detect malicious jamming activities through power variation analysis. In order to train our model, we generated a dataset that has two distinct communication scenarios: Line-of-Sight (LoS) and Non-Line-of-Sight (NLoS). Each scenario category contains four unique experimental configurations that vary key parameters including UAV mobility patterns, operational speeds, attack intensities, and network user density. The dataset architecture specifically accounts for urban environment dynamics, where building structures and other obstacles can significantly impact signal propagation. To ensure robust detection capabilities, we simulate various attack patterns using different numbers of hostile UAVs (ranging from 1 to 4 attackers) with varying transmission powers. The dataset inherently presents an unbalanced distribution between attack and non-attack scenarios, necessitating careful preprocessing to maintain classification accuracy. Each simulation captures temporal sequences of received power measurements of RSSI and SINR.

A. Channel Attenuation

The wireless channel between aerial platforms and ground stations experiences signal degradation through multiple mechanisms. We characterize the total channel attenuation as the superposition of deterministic distance-dependent losses and stochastic variations. The aggregate channel loss $H(r, f_c)$ in decibels can be expressed as:

$$H(r, f_c) = \Psi_d(r, f_c) + \Omega \tag{1}$$

where $\Psi_d(r, f_c)$ represents the distance-dependent attenuation component for a link spanning distance r (measured in kilometers) at carrier frequency f_c (in MHz), while Ω captures the random fluctuations arising from environmental factors such as shadowing effects caused by buildings and vegetation.

B. Propagation State Classification

Urban A2G channels exhibit distinct propagation characteristics depending on the presence or absence of obstructions between transmitter and receiver. We categorize the propagation environment into two fundamental states: Line-of-Sight (LoS) conditions where an unobstructed propagation path exists, and Non-Line-of-Sight (NLoS) scenarios where intermediate objects block the direct signal path. This binary classification forms the foundation for modeling distance-dependent attenuation and statistical channel variations.

The distance-dependent attenuation component adapts based on the prevailing propagation conditions. Under LoS conditions, the path loss reflects the maximum between free-space propagation and near-ground effects:

$$\Psi_d^{\text{LoS}}(r, f_c) = \max\{\Psi_{\text{free}}(r, f_c), \Psi_{\text{near}}(r, f_c)\} \tag{2}$$

The free-space propagation component follows the fundamental Friis transmission equation:

$$\Psi_{\text{free}}(r, f_c) = 32.45 + 20\log_{10}(r) + 20\log_{10}(f_c) \tag{3}$$

Near-ground propagation introduces altitude-dependent effects that modify the path loss exponent:

$$\Psi_{\text{near}}(r, f_c) = A_0 + A_1\log_{10}(r) + 20\log_{10}(f_c) \tag{4}$$

where the coefficients $A_0 = 30.9$ and $A_1 = 22.25 - 0.5\log_{10}(z)$ depend on the UAV altitude z measured in meters. For NLoS propagation scenarios, additional attenuation

occurs due to diffraction and scattering phenomena. The effective path loss under NLoS conditions incorporates both the LoS component and obstruction-induced losses:

$$\Psi_d^{\text{NLoS}}(r, f_c) = \max\{\Psi_d^{\text{LoS}}(r, f_c), \Psi_{\text{blocked}}(r, f_c)\} \quad (5)$$

The blockage-induced attenuation exhibits stronger distance dependence and altitude sensitivity:

$$\Psi_{\text{blocked}}(r, f_c) = B_0 + B_1 \log_{10}(r) + 20\log_{10}(f_c) \quad (6)$$

with parameters $B_0 = 32.4$ and $B_1 = 43.2 - 7.6\log_{10}(z)$ calibrated for urban environments.

C. Statistical Channel Fluctuations

Beyond deterministic path loss, the wireless channel experiences random variations due to shadowing effects from buildings, vegetation, and other urban structures. These fluctuations follow log-normal distributions with standard deviations that depend on both the propagation state and UAV altitude. For LoS conditions, the shadowing standard deviation decreases exponentially with altitude, reflecting reduced interaction with ground-level obstructions:

$$\sigma_\Omega^{\text{LoS}} = \max\{5\exp(-0.01z), 2\} \quad [\text{dB}] \quad (7)$$

In contrast, NLoS scenarios exhibit more severe and altitude-independent shadowing with $\sigma_\Omega^{\text{NLoS}} = 8$ dB. These parameters remain valid for UAV operations within the altitude range of 22.5 to 300 meters, encompassing typical low-altitude urban flight scenarios.

D. Propagation State Probability

The occurrence of LoS or NLoS conditions follows a stochastic model based on the geometric relationship between UAV position and urban topology. The probability of a LoS link decreases with horizontal distance and increases with UAV altitude, capturing the intuitive notion that higher-flying UAVs experience fewer obstructions. We model the LoS probability as:

$$P_{\text{LoS}} = \frac{\xi_1}{r_{xy}} + \exp\left(\frac{-r_{xy}}{\xi_2}\right)\left(1 - \frac{\xi_1}{r_{xy}}\right) \quad (8)$$

where r_{xy} denotes the horizontal projection of the three-dimensional link distance. The altitude-dependent parameters $\xi_1 = \max\{294.05\log_{10}(z) - 432.94, 18\}$ and $\xi_2 = -233.98\log_{10}(z) - 0.95$ ensure appropriate limiting behavior at both low and high altitudes. The complementary NLoS probability follows directly as $P_{\text{NLoS}} = 1 - P_{\text{LoS}}$.

E. Multipath Channel Structure

Small-scale fading arises from the constructive and destructive interference of multiple signal replicas arriving via different propagation paths. We adopt a clustered delay line approach where multipath components group into clusters, each characterized by specific delay, power, and angular properties. The multipath contribution $\Phi(N_c, N_r)$ depends on the number of clusters N_c and rays per cluster N_r, with parameters including azimuth and elevation spreads for both arrival and departure angles. This geometric channel model enables accurate characterization of spatial correlation and antenna pattern effects in multi-antenna systems.

F. Signal Power and Quality Metrics

The received signal power at the aerial platform incorporates transmit power, antenna gains, and all channel attenuation mechanisms:

$$P_r = P_t + G_t + G_r - H(r, f_c) - \Phi(N_c, N_r) \quad [\text{dBm}] \quad (9)$$

where P_t denotes the transmit power in dBm, while G_t and G_r represent the transmit and receive antenna gains in dBi, respectively.

In the presence of interference from co-channel users and intentional jamming sources, the signal quality degrades according to the signal-to-interference-plus-noise ratio (SINR):

$$\gamma = \frac{P_r}{N_0 + \sum_{k=1}^{K} I_k} \quad (10)$$

where N_0 represents the thermal noise power and I_k denotes the interference power from the k-th source among K total interferers.

Network infrastructure commonly reports the Received Signal Strength Indicator (RSSI) as a measure of total received power across the allocated spectrum. Unlike metrics focused solely on the desired signal, RSSI captures contributions from all sources:

$$\text{RSSI}_{\text{dBm}} = 10\log_{10}\left(P_r^{\text{linear}} + \sum_{k=1}^{K} I_k^{\text{linear}} + N_0^{\text{linear}}\right) \quad (11)$$

The proposed channel model is valid for UAV operations at altitudes between 22.5 and 300 meters, encompassing typical urban flight profiles below controlled airspace. It is applicable to standard cellular frequency bands and assumes single-antenna configurations at both the UAV and ground station. Extensions to multi-antenna systems are straightforward, leveraging the geometric structure of the multipath model [13], [57].

G. Dataset on Jamming Detection

A sample from the dataset on the jamming effects is presented in figure 1. These simulation results capture the signal propagation characteristics under both LoS and NLoS conditions. The RSSI measurements (a) show four distinct traces over 300 samples: LoS without attack (green) maintaining stable values around -83 dBm, LoS under attack (red) fluctuating around -81 dBm with occasional spikes, NLoS without attack (blue) exhibiting significant variability with deep fades to -95 dBm, and NLoS under attack (orange) showing consistently degraded performance near -95 dBm.

The LoS scenarios demonstrate relatively stable RSSI patterns, while both NLoS conditions suffer from severe signal attenuation, with the attack scenario showing more consistent degradation.

The SINR measurements (b) display corresponding signal quality metrics for the same four scenarios. LoS without attack (green) shows highly variable SINR values ranging from -5 to +12 dB, while LoS under attack (red) exhibits similar variability between -8 and +14 dB. NLoS without attack (blue) maintains relatively constant SINR around -11 to -12 dB, and NLoS under attack (orange) shows the most severely degraded SINR values at approximately -16 dB with minimal variation. The measurements demonstrate a clear ordering of signal quality: LoS no attack > LoS attack > NLoS no attack > NLoS attack. To enable reliable data collection even under degraded conditions, the connection was maintained throughout, even during periods of low SINR. More comprehensive information regarding the dataset, the quantity of jamming UAVs, and signal parameters can be found in [13] and [57].

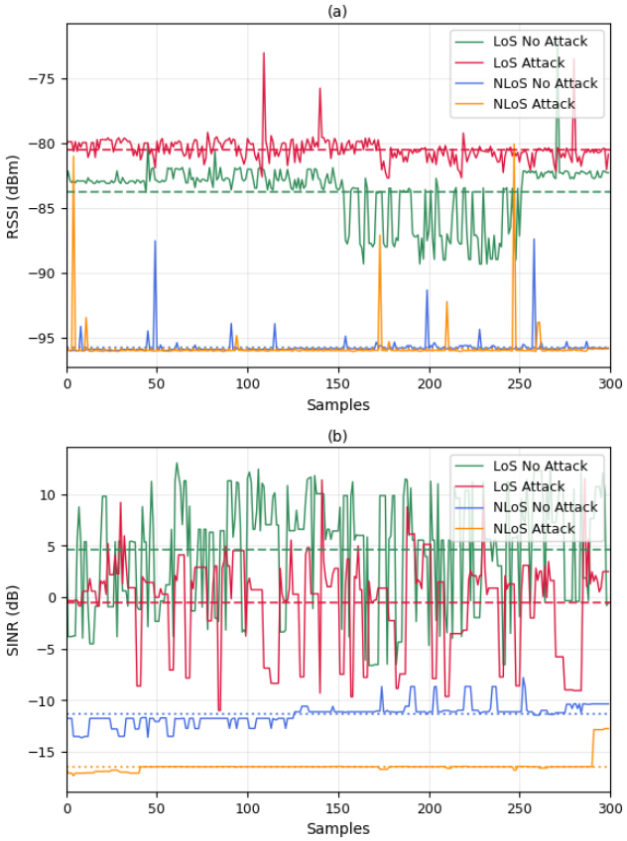


FIGURE 1: RSSI and SINR measurements for wireless communication: (a) RSSI in dBm and (b) SINR in dB for LoS and NLoS scenarios with and without jamming attacks.

A distribution of the dataset on the jamming is presented in figure 2, which illustrates the distributions of RSSI and SINR.

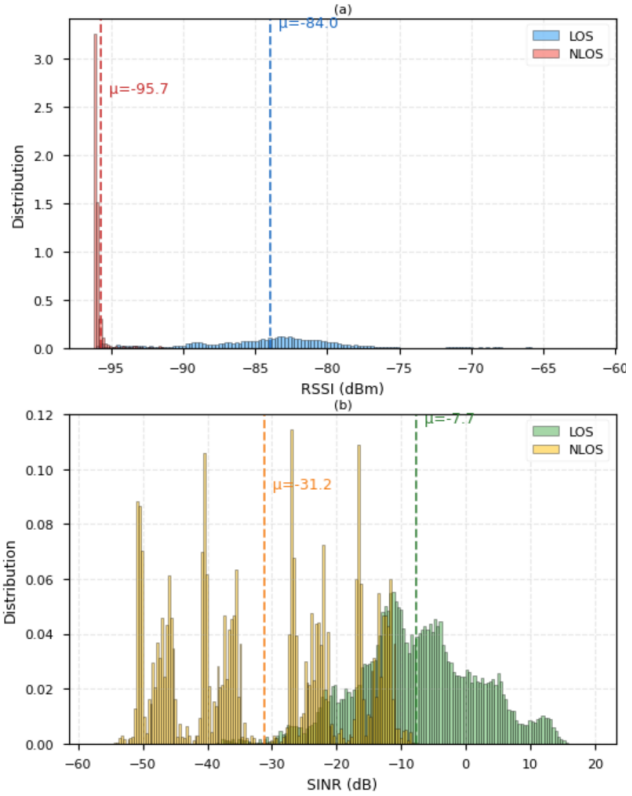


FIGURE 2: RSSI and SINR with jamming experienced by the UAV receiver.

The RSSI measurements exhibit a distinct bimodal distribution pattern between LoS and NLoS conditions. The NLoS signals demonstrate a concentrated distribution with $\mu_{rssi} = -95.7$ dBm, indicating substantial signal attenuation. Conversely, the LoS signals present a more dispersed distribution centered at $\mu_{rssi} = -84.0$ dBm, reflecting superior signal strength characteristics typical of direct path propagation. The NLoS distribution's pronounced, narrow peak suggests consistent attenuation patterns, while the broader LoS distribution indicates more diverse signal propagation paths despite maintaining direct visibility.

The SINR distributions effectively illustrate the jamming environment's impact on signal quality. NLoS signals exhibit multiple distinct peaks in the range of -50 dB to -20 dB, with a mean value of $\mu_{sinr} = -31.2$ dB, demonstrating severe interference effects. The LoS signals display a more favorable distribution extending into positive values, characterized by $\mu_{sinr} = -7.7$ dB. The broader, right-skewed distribution observed in LoS conditions indicates that despite direct visibility, jamming significantly degrades signal quality, albeit to a lesser extent compared to NLoS scenarios.

H. Jamming Detection algorithm for UAVs

The proposed approach combines PCA features with transformer architectures to create an efficient and robust detection system.

1) **Feature Engineering with PCA for Time Series Data**

Direct application of PCA to time series data often fails to capture temporal dependencies effectively, resulting in suboptimal performance. In our initial experiments, training the model solely with PCA components did not yield satisfactory results. To overcome this limitation, we utilized PCA to generate additional features, which were then integrated with the raw data. This approach improved classification accuracy by up to 5% for both LoS and NLoS datasets.

*a: **Sample Creation***

For each time series signal S (either RSSI or SINR), we transformed the 1D signal into a 2D sample matrix. Using a rolling window of size 300, a signal S of length N was converted into a matrix X of shape $(N-299, 300)$, where each row represents a sample of 300 consecutive time steps.

*b: **Transformations for Feature Enhancement***

To capture diverse temporal patterns in the 2D sample matrix X, we applied a series of transformations. These include:

- The original samples X,
- Moving averages with window sizes of 2, 3, and 5, adjusted via slicing to align output dimensions,
- Sub-sampled versions of X by selecting every 2nd or 3rd feature, with varying starting indices to account for phase shifts.

In total, nine distinct transformations per signal were generated. For the precise definitions and implementation details of each transformation, refer to **Algorithm 1** (Feature Creation Algorithm). These transformations enhance the temporal representation of the data, preparing it for the subsequent PCA-based feature extraction step.

*c: **PCA Application and Feature Extraction***

For each of the nine transformed matrices:

1) A PCA model was fitted to the data, retaining the principal components (PCs) that collectively explain 99% of the total variance.
2) For further process, the first five principal components were selected to reducing the feature dimensionality while retaining a significant portion of the data's variability.

This process generated up to five PCA features for each transformation applied to the RSSI and SINR signals. Given that a total of nine transformations were applied to each signal, this resulted in a maximum of 45 features for each original signal (RSSI or SINR). Consequently, for both RSSI and SINR combined, a total of 90 features were obtained for LoS scenarios, whereas for NLoS scenarios, a total of 54 features were derived. This discrepancy arises because, in some transformed signals, only a single PC captured the entire variance of the transformation.

*d: **Feature Scaling and Integration***

Each of the 45 PCA feature columns per signal was normalized to the range of the original sample matrix X using the `MinMaxScaler`, with the feature range set to $(\min(X), \max(X))$. The scaled PCA features were then concatenated with the original samples X along the feature axis, creating an enhanced feature set. This integration preserved the original data structure and avoided modifications to the tokenization process.

Algorithm 1 PCA-Based Feature Creation for Time Series Signals

Input: Time series signals S_{RSSI} and S_{SINR}
Output: Enhanced feature sets for RSSI and SINR
for each signal S in $\{S_{\text{RSSI}}, S_{\text{SINR}}\}$ **do**
  $X \leftarrow \text{rolling\_window}(S, \text{size} = 300)$ ▷ Create samples, shape: $(|S| - 299, 300)$
  Define transformation set $\mathscr{T}$:
    $T_1(X) = X$ ▷ Original samples
    $T_2(X) = \text{moving\_average}(X, n = 2)[:, 2:]$ ▷ Window 2, columns 2 to end
    $T_3(X) = \text{moving\_average}(X, n = 3)[:, 3:]$ ▷ Window 3, columns 3 to end
    $T_4(X) = \text{moving\_average}(X, n = 5)[:, 5:]$ ▷ Window 5, columns 5 to end
    $T_5(X) = X[:, :: 2]$ ▷ Every 2nd point, start at 0
    $T_6(X) = X[:, 1 :: 2]$ ▷ Every 2nd point, start at 1
    $T_7(X) = X[:, :: 3]$ ▷ Every 3rd point, start at 0
    $T_8(X) = X[:, 1 :: 3]$ ▷ Every 3rd point, start at 1
    $T_9(X) = X[:, 2 :: 3]$ ▷ Every 3rd point, start at 2
  for each transformation T_i in $\mathscr{T}$ **do**
    Fit PCA_i on $T_i(X)$ with $n_{\text{components}}$ retaining 99% variance
  end for
  $X_{\text{pca}} \leftarrow \text{concatenate}([\text{PCA}_i.\text{transform}(T_i(X))[:, :5] \text{ for } i = 1 \text{ to } 9], \text{axis} = 1)$
  $\min_X, \max_X \leftarrow \min(X), \max(X)$ ▷ Global min and max of X
  $X_{\text{pca}} \leftarrow \text{scale}(X_{\text{pca}}, \text{range} = [\min_X, \max_X])$ ▷ Scale each column
  $X_{\text{enhanced}} \leftarrow \text{concatenate}([X, X_{\text{pca}}], \text{axis} = 1)$
end for
return X_{enhanced} for each signal

*e: **Tokenization***

The data processing pipeline of the proposed transformer model begins with the collection of enhanced variants of RSSI and SINR derived via Algorithm 1. Each enhanced signal is discretized into 50 equal-sized bins using percentile-based discretization, ensuring a uniform distribution across the data range. This process assigns a unique integer value to each bin, yielding 50 tokens per signal type. Consequently, two sets of 50 bins are produced—one for RSSI and one for SINR—resulting in a total of 100 distinct tokens representing the wireless signal characteristics. The selection of 50 equal-sized bins was determined through systematic hyperparameter optimization, where bin counts ranging from

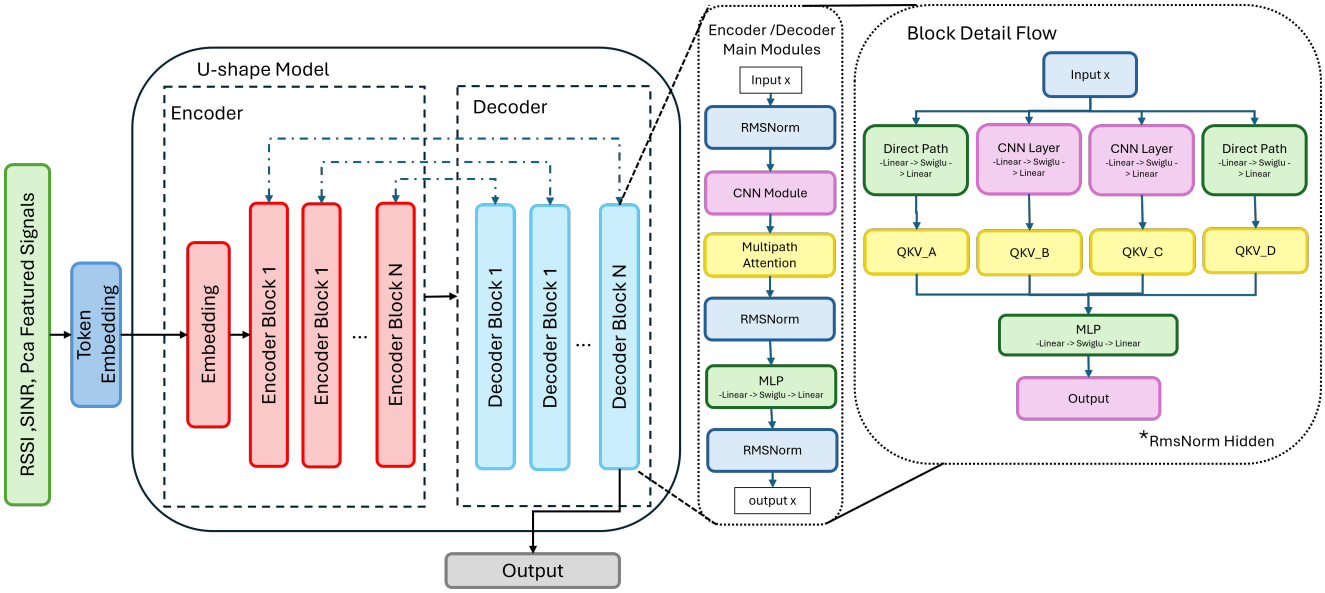


FIGURE 3: U_Shaped Architecture

25 to 100 were evaluated based on the model's classification performance metrics.

To construct the token vocabulary, additional utility tokens are incorporated: `CLS` (assigned index 1), `MASK` (index 4), and binary classification tokens `NO_ATTACK` (index 5) and `ATTACKED/JAMMED` (index 6). Other indices until 9 are reserved for potential future use, while the RSSI and SINR tokens span indices 9 to 109. This configuration results in a total vocabulary size of 110 tokens, with the maximum index capped at 110 for consistency across both LoS and NLoS scenarios. For training data, Time-Series Augmentation (TSA) techniques, inspired by [13], are applied to the discretized RSSI and SINR signals to enhance robustness. The input sequence is then formed by concatenating the `CLS` token, the discretized enhanced RSSI, the discretized enhanced SINR, and the label token. For LoS, this yields an input length of $1 + 345 + 345 + 1 = 692$ tokens, while for NLoS, the length is $1 + 345 + 309 + 1 = 656$ tokens, reflecting differences in signal characteristics between the two scenarios.

2) Deep Network Design

Building upon prior work on multi-headed deep networks with attention mechanisms [13], we propose a U-shaped deep network architecture, illustrated in figure 3, designed to deliver state-of-the-art performance in NLoS scenarios without reliance on post-processing techniques. Inspired by the U-Net architecture [58], this model incorporates modifications to integrate multi-headed attention and variable embedding dimensions across layers, enhancing its capability for robust signal analysis.

The architecture consists of an encoder pathway with three stages, a corresponding decoder pathway with three stages, and residual skip connections that link the two. These skip connections enable direct gradient propagation from deeper to shallower layers, mitigating gradient degradation during backpropagation. The model processes input features and generates output tokens that encapsulate predictions, achieving competitive performance without additional refinement steps.

Input processing begins with the integration of RSSI measurements, SINR values, and PCA-derived features. These inputs are transformed into high-dimensional representations via a token embedding layer, preparing them for subsequent transformer-based processing.

The encoder pathway, depicted in red in figure 3, comprises sequential blocks that progressively refine input representations. Each block integrates Convolutional Neural Network (CNN) modules for local feature extraction, multi-headed attention mechanisms for selective signal focus, and Root Mean Square Normalization (RMSNorm) layers [59] for training stability. These components are complemented by Multi-Layer Perceptron (MLP) units featuring a Linear-SwiGLU-Linear activation sequence, forming a robust feature extraction pipeline [60].

The decoder pathway, shown in blue, mirrors the encoder's structure while incorporating skip connections from the corresponding encoder layers to preserve critical information. Symmetry between the encoder and decoder is maintained, with the decoder leveraging similar components to reconstruct and interpret encoded features for classification. This U-shaped configuration enhances multi-scale feature analysis within the transformer framework.

Within each block, parallel processing paths enhance feature extraction: direct Linear-SiLU-Linear transformations[61] operate alongside CNN layers, while multiple Query-Key-Value (QKV) attention mechanisms [34] process distinct signal aspects concurrently. Inspired by the Diff Transformer from [62], differential computations are applied to the attention outputs of specific heads (e.g., heads 6 and 7) to enrich feature representations. A final MLP layer integrates these diverse features for comprehensive analysis.

RMSNorm is employed at multiple stages to ensure consistent normalization and training stability, enabling the model to focus on pertinent signal characteristics while suppressing noise. Skip connections further preserve gradient flow, enhancing training efficiency. The architecture culminates in a classification layer that leverages these rich feature representations to accurately predict jamming threats, capturing both overt and subtle attack patterns.

This design excels in processing multiple input feature types simultaneously while preserving their interrelationships. The U-shaped structure facilitates hierarchical feature extraction, and skip connections ensure information retention across processing stages, making the model well-suited for complex signal analysis tasks.

a: Detailed Architecture Specification

The encoder and decoder blocks operate at varying embedding dimensions, defined as follows:

- **Encoder Dimensions:** 256, 128, 64
- **Decoder Dimensions:** 64, 128, 256

Each block comprises:

- An RMSNorm layer for normalization.
- A CNN layer with kernel size 3 and padding 1 for local feature extraction.
- A Linear-SiLU-Linear transformation sequence.
- An 8-headed attention mechanism with differential computations.
- Residual connections paired with SwiGLU-activated MLP blocks.

The hierarchical reduction of embedding dimensions in the encoder is symmetrically reversed in the decoder, maintaining structural balance.

b: Attention Mechanism and Differentiation

The multi-headed attention mechanism processes embeddings through distinct groups:

- Heads 0 and 1 directly process normalized inputs.
- Heads 2 and 3 process CNN-transformed embeddings via an MLP.
- Heads 4 and 5 rely solely on CNN transformations.
- Heads 6 and 7 compute differences between normalized inputs and CNN-transformed outputs.

This configuration, combined with differential attention inspired by [62], enhances the model's ability to capture nuanced signal variations, contributing to its effectiveness in identifying jamming patterns.

3) Proposed Training Algorithm

In this work, we propose a robust training framework for time series data that accelerates convergence and improves generalization. The framework integrates four key components: (i) a chunking strategy for data sampling, (ii) dynamic batch size scheduling combined with learning rate adjustment, (iii) an Exponential Moving Average (EMA) for model weights with an integrated restoration mechanism, and (iv) mixed precision training with gradient clipping. These components are detailed in the following sections.

a: Chunking Strategy

To mitigate temporal correlations and reduce training time, we adopt a chunking strategy that partitions the training dataset into a fixed number of chunks (e.g., 10). In each epoch, a subset is selected from the dataset by choosing every nth sample with an offset determined by a randomized permutation. This ensures that samples used in a mini-batch maintain a minimum temporal gap, thereby promoting diversity and reducing overfitting. The pseudocode for the chunking algorithm is provided in Algorithm 2.

Algorithm 2 Chunking Strategy for Data Selection

Require: Number of chunks n, current epoch e, training dataset D_{train}

Set $n \leftarrow 10$ ▷ User-defined number of chunks
Compute $s \leftarrow e \mod n$
if $s = 0$ **then**
    Generate a new random permutation P of $\{0, 1, \ldots, n-1\}$
    set $s \leftarrow 0$
else
    Set $s \leftarrow P[s]$
end if
Select subset: $D_{\text{subset}} \leftarrow D_{\text{train}}[s :: n]$
return D_{subset}

In each epoch, the subset D_{subset} is used for training, ensuring that samples are minimally correlated (with a gap of at least 10 time steps) and that the overall data diversity is maintained.

b: Dynamic Batch Size Scheduling and Learning Rate Adjustment

To optimize convergence, our framework employs a dynamic batch size scheduler that adjusts the effective batch size via gradient accumulation. Initially, the scheduler is deactivated. If validation loss and accuracy fail to improve over successive epochs, the scheduler increases the effective batch size by modifying the gradient accumulation steps. This approach enables the network to benefit from larger batch sizes without incurring additional memory overhead. Concurrently, a learning rate scheduler with a warmup phase (e.g., 8 epochs) adjusts the learning rate dynamically. The

combination of these schedulers accelerates convergence and enhances validation performance. [63]

c: Weight Moving Average and Restoration Mechanism

To stabilize training and improve generalization, an EMA of the model parameters is maintained. When validation performance improves, the current model state is saved as the best checkpoint. The EMA is then updated according to:

$$W_{\text{new}} = \alpha \cdot W_{\text{prev}} + (1-\alpha) \cdot W_{\text{current}}, \tag{12}$$

where α is a small factor (e.g., $\alpha = 0.001$), W_{prev} denotes the previously maintained weight vector, and W_{current} represents the current weights

- Parameter Selection and Initialization: The factor $\alpha = 0.001$ was selected based on established EMA practices in deep learning [64] and provides robust performance without extensive tuning. Initialization is straightforward: the EMA vector is initialized using the model's parameters after the first epoch, ensuring the moving average begins with meaningful weights rather than random values. This value balances training stability (preventing noise amplification) with adaptation responsiveness (allowing genuine improvements to influence the averaged weights).

In cases where validation metrics degrade for two consecutive epochs (tracked via a restoration counter), the model is restored to the best checkpoint and a modified weight update is applied with $\alpha = 0.005$ to accelerate recovery from performance degradation. This parameter was determined through empirical evaluation during our preliminary experiments and eliminates the need for manual parameter tuning in practical applications, as the system automatically adjusts between conservative averaging ($\alpha = 0.001$) and aggressive adaptation ($\alpha = 0.005$) based on training dynamics. This restoration mechanism ensures that the training process remains stable despite fluctuations in performance while providing built-in robustness without requiring hyperparameter optimization.

Our implementation includes several key features that informed this decision: 1. Dual validation criteria: The model checkpoint is saved when either validation loss decreases OR validation accuracy improves, providing flexibility in optimization trajectories. 2. Soft restoration mechanism: When patience is exceeded, we don't simply revert to the best weights. Instead, we apply exponential moving average (EMA) with a factor of 0.005, blending 99.5% of the best model weights with 0.5% of the original initialization. This approach helps maintain some exploration capacity while primarily focusing on the proven good configuration. 3. Adaptive batch size scheduling: Upon restoration, we enable batch size scheduling, which provides an additional mechanism for escaping local minima through gradient noise modulation.

d: Mixed Precision Training and Gradient Clipping

To improve computational efficiency, the training process employs mixed precision training using PyTorch's Automatic Mixed Precision (AMP) framework. Computations are performed in `bfloat16` precision on CUDA devices, and a gradient scaler is used to avoid numerical underflow. Furthermore, gradient clipping with a maximum norm of 1.0 is applied to prevent exploding gradients, thereby ensuring stable and robust model updates.

e: Loss Function with Entropy Regularization

Inspired by the maximum entropy reinforcement learning framework underpinning the SAC algorithm [65], we adapt the entropy regularization principle to supervised classification. In Soft Actor Critic (SAC), the policy is optimized not only to maximize expected reward but also to maximize the entropy of the action distribution, promoting stochasticity that leads to better exploration and robustness in high-dimensional environments.

While SAC operates in the reinforcement learning domain, the underlying principle of entropy regularization translates effectively to supervised learning for uncertainty quantification [66], [67]. In our classification context, the entropy term serves an analogous purpose: encouraging the model to maintain prediction uncertainty when the evidence is ambiguous, thereby preventing overconfident predictions on noisy or adversarial inputs. This approach has been shown to improve generalization in classification tasks, particularly in domains with high noise or distributional shift

The entropy regularization in our loss function thus adapts SAC's exploration strategy to the supervised learning paradigm, promoting robustness against jamming attacks that may not perfectly match training scenarios. The standard classification loss is augmented with an entropy regularizer. The modified loss function is defined as

$$L = \frac{L_{\text{base}} - \lambda_{\text{entropy}} H(X)}{N_{\text{accum}}}, \tag{13}$$

where L_{base} is the standard cross-entropy loss,

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{14}$$

denotes the entropy of the predicted probability distribution, λ_{entropy} (e.g., 0.4) is a hyperparameter controlling the regularization strength, and N_{accum} is the number of gradient accumulation steps. By incorporating an entropy regularizer into the classification loss, we encourage the model to maintain higher predictive entropy—avoiding overly confident or deterministic output distributions unless strongly supported by the data. This entropy regularization promotes better generalization by mitigating overfitting, particularly in low-data or noisy settings, and aligns with recent findings that uncertainty-aware models tend to perform more robustly. In practice, if the computed loss becomes non-finite (e.g., NaN or infinity), the corresponding batch is skipped to ensure stable and reliable training. This approach adapts SAC's entropy-based regularization from reinforcement learning to supervised classification in a principled way, preserving the core benefit of uncertainty-driven learning dynamics.

Scenario Parameters	**Values**
Terrestrial Users	0, 5, 10
Authenticated UAVs	1
Small Cells	10
Small cell height	10 m
Attackers	0, 1, 2, 3, and 4
Speeds	10 m/s
Small cell power	4 dBm
Authenticated UAV power	2 dBm
Attackers power	0, 2, 5, 10, and 20 dBm
Authenticated UAV position	URD*
Attackers position	URD*
Small cells position	URD*
Distance	100, 200, 500, and 1000 m
Central Frequency	3.5e9
Bandwidth	20e6
Noise Figure	5dB

*URD - Uniformly Random Distributed.

TABLE 1: Dataset Parameters. [13]

Parameter	**LoS** and **NLoS**
Block size (LoS, NLoS)	(692, 656)
Layer number	6
Learning Rate	1×10^{-4}
Heads number	8
Vocab size	110
Encoder Embedding	[256, 128, 64]
Decoder Embedding	[64, 128, 256]
Dropout	0.4
Batch Size	64
Noise	0.03
(Rand. Mask Prob., Target Mask Prob.) (training)	(0.25, 0.85)
(Rand. Mask Prob., Target Mask Prob.) (prediction)	(0.0, 1.0)
Model Parameters	2.2 M

TABLE 2: U-shaped Transformer Model configuration

Category	**NLoS**	**LoS**
Proposed + entropy	**85.06**	89.46
Proposed	79.10	87.37
DNN [13]	75.60	89.59
DNN+M1 [13]	83.07	89.98
DNN+M2 [13]	79.00	**90.80**
XGBoost [13]	80.58	86.33
CNN + entropy [68]	68.21	79.89
CNN [68]	64.88	81.86

TABLE 3: Classification accuracy (%) with window size 300. Best results are in bold.

f: Summary

The integration of a data chunking strategy, dynamic batch size and learning rate scheduling, an EMA with a restoration mechanism, and mixed precision training with gradient clipping results in a comprehensive framework for training models on time series data. Experimental results demonstrate that this approach not only accelerates convergence but also enhances generalization, yielding improved validation and test performance.

III. Results

a: Experimental Setup and Methodology

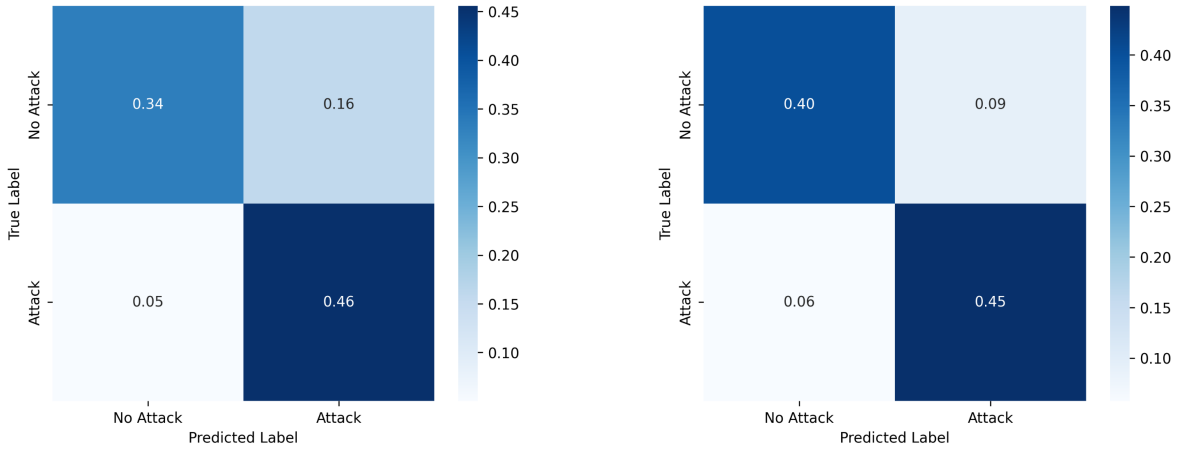
This section presents the results and performance analysis of our proposed U-shaped transformer architecture with PCA-enhanced features against established baseline methods. We evaluate our approach against seven comparison algorithms: DNN, DNN+M1, DNN+M2, and XGBoost classifiers from [13], CNN architecture from [68], and our proposed transformer variants with and without entropy regularization. All experiments employ consistent dataset partitioning with 70% training, 15% validation, and 15% testing using temporal stratification to prevent data leakage. The proposed transformer architecture follows the configuration parameters detailed in Tables 1 and 2, respectively.

A. Classification Comparison with Other Algorithms

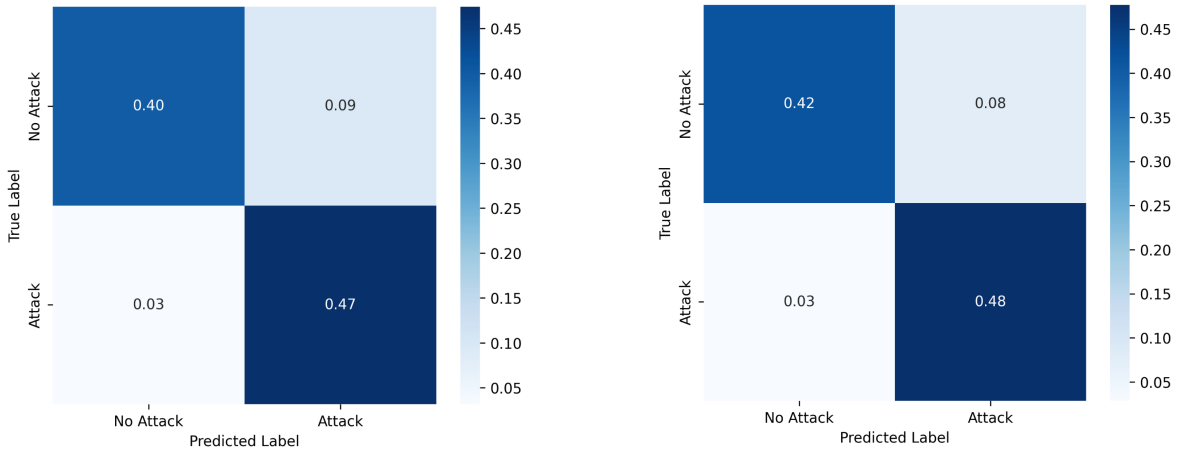
Table 3 presents the comparative performance of our proposed approaches against seven different baseline methods for classification accuracies in both NLoS and LoS scenarios.

The entropy-enhanced model ("Proposed + entropy") achieves the highest detection rate in NLoS conditions at 85.06%, significantly outperforming all competing methods. This performance underscores the effectiveness of our uncertainty-based regularization technique in challenging signal environments where direct paths are obstructed. In LoS conditions, our entropy-enhanced model achieves a competitive 89.46% detection rate, while the DNN+M2 approach from [13] shows marginally better performance at 90.80%.

The results demonstrate that entropy-based regularization consistently improves detection capabilities, as evidenced by the performance gap between our basic and entropy-enhanced models (79.10% vs. 85.06% in NLoS, and 87.37% vs. 89.46% in LoS). This pattern is also observed in CNN-based methods in [68], where entropy integration shows mixed effects depending on the signal environment. Notably, our proposed method without entropy regularization ("Proposed") reaches 87.37% in LoS scenarios, outperforming both XGB (86.33%) and CNN-based approaches (81.86% and 79.89%), further validating the effectiveness of our U-shaped transformer architecture even without the additional entropy component. When comparing our approach with DNN variants in [13], we observe that our entropy-enhanced model provides more balanced performance across both LoS and NLoS conditions. While DNN+M1 and DNN+M2 achieve strong results in LoS scenarios (89.98% and 90.80% respectively), they demonstrate more significant performance degradation in the challenging NLoS environment (83.07%



(a) Confusion matrices for jamming classification in Non-Line-of-Sight (NLoS) conditions: standard classification approach (left) vs. entropy-enhanced classification (right).



(b) Confusion matrices for jamming classification in Line-of-Sight (LoS) conditions: standard classification approach (left) vs. entropy-enhanced classification (right).

FIGURE 4: Performance comparison of jamming classification algorithms under different signal propagation conditions. The matrices illustrate classification accuracy with and without entropy-based feature enhancement, demonstrating the impact of both propagation environment and algorithm selection on jamming detection performance.

and 79.00%). This highlights the robustness of our proposed U-shaped transformer architecture with entropy regularization, which maintains high detection rates even under adverse signal conditions, making it a reliable solution for real-world wireless security applications such as the UAV non-line-sight applications.

B. Confusion matrix with entropy and no entropy

Figure 4 illustrates the performance matrices for our signal jamming classification framework evaluated under varied propagation environments. Our investigation contrasts traditional classification techniques against our novel entropy-enhanced methodology across both NLoS and LoS transmission scenarios.

In the NLoS condition (Figure 4a), the standard classification approach achieves a true positive rate of 0.46 for attack detection and a true negative rate of 0.34 for non-attack classification. However, the relatively high false positive rate of 0.16 indicates a tendency to misclassify legitimate transmissions as attacks. When the entropy-enhanced approach is applied, we observe a significant improvement in discrimination capability. While the true positive rate remains comparable at 0.45, the true negative rate increases substantially to 0.40, with a corresponding reduction in false positives to 0.09. This represents a 43.8% decrease in false alarms, which is crucial for practical deployment of jamming detection systems where false positives can lead to unnecessary countermeasures and system disruption.

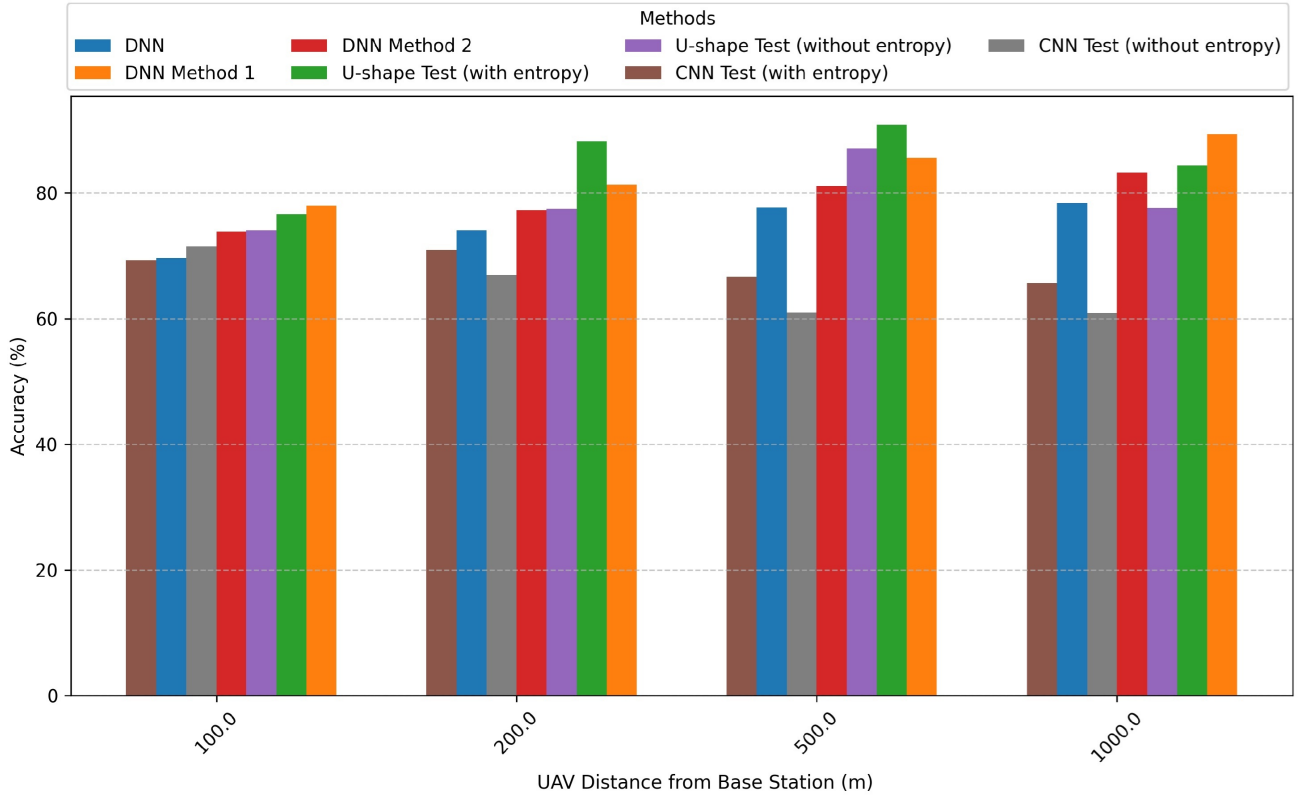


FIGURE 5: Comparison with Deep Learning Results Based UAV - BS Distance in NLoS Conditions. The proposed model is accentuated in green for emphasis

For LoS configurations (Figure 4b), both methodologies exhibited enhanced performance relative to NLoS conditions, due to the more consistent signal properties characteristic of direct-path transmission. The conventional approach recorded a true positive rate of 0.47, true negative rate of 0.40, and false positive rate of 0.09. Our entropy-enhanced algorithm further refined these metrics: increasing the true positive rate to 0.48, true negative rate to 0.42, while reducing false positives to 0.08. Although these improvements appear incremental compared to the NLoS scenario, they consistently validate the efficacy of entropy-based feature incorporation across diverse propagation conditions.

The performance variations between NLoS and LoS environments highlight how propagation characteristics fundamentally influence jamming detection reliability. NLoS configurations introduce significant challenges through multipath propagation effects, signal degradation, and high variability—factors that can mask the characteristic of jamming activities. Our entropy-enhanced approach demonstrates particular effectiveness in addressing these challenges through the implementation of our entropy-augmented loss function.

C. Distance Based Accuracy Comparison

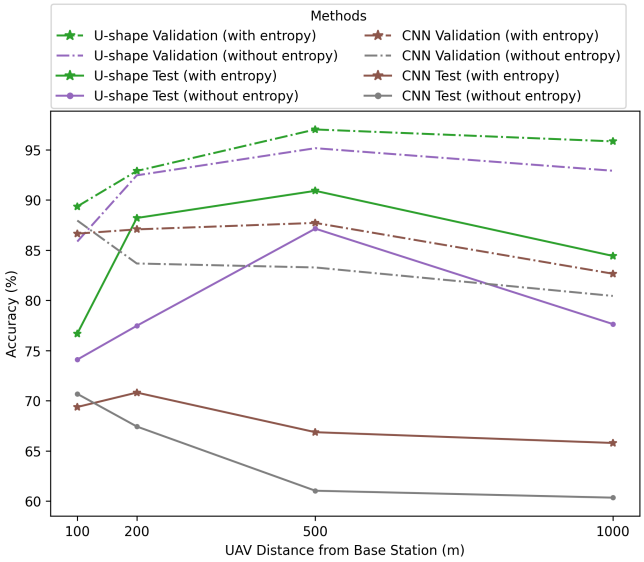
The bar graph in figure 5 compares the accuracy performance of seven neural network methods at four UAV distances from a base station: 100m, 200m, 500m, and 1000m. The methods include three DNN variants (standard DNN, DNN Method 1, DNN Method 2) and four testing approaches (U-shape and CNN tests, each with and without an entropy regularizer).

At 100m, all methods achieve between 65-80% accuracy, with DNN Method 1 and U-shape Test with entropy showing slightly better performance. As distance increases to 200m, a performance gap emerges with U-shape Test with entropy reaching 88.2%, while CNN Test without entropy drops to about 66.9%. The 500m distance marks the peak performance point for most methods, with U-shape Test with entropy exceeding 90.9% accuracy, closely followed by U-shape Test without entropy at around 87.1% and DNN Method 1 at approximately 85%. Both DNN Method 2 and standard DNN also perform well at this distance, while CNN tests remain significantly lower. At the maximum tested distance of 1000m, DNN Method 1 achieves the highest accuracy at approximately 89.3%, followed by DNN Method 2 and U-shape Test with entropy at about 84%. This suggests DNN Method 1 has superior performance at extreme distances, while CNN-based approaches consistently show the poorest results across all tested ranges. The consistent superiority of entropy-incorporated methods across all distances indicates that entropy provides valuable

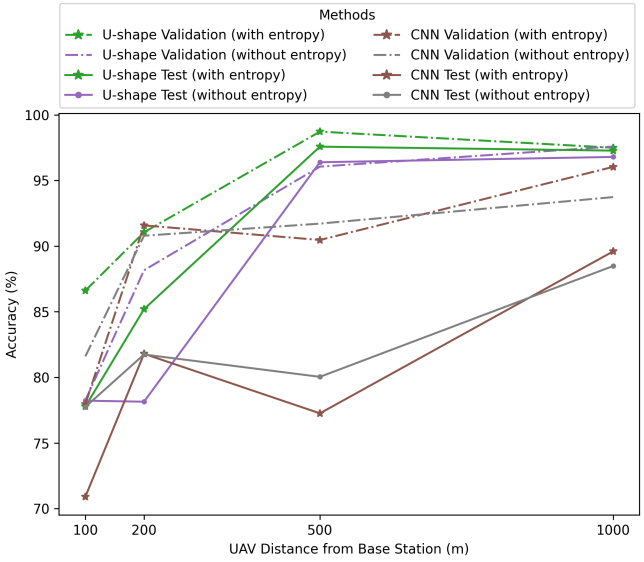
information for the neural network classification process in this UAV application context.

D. Comparison with entropy and no entropy

Figures 6a and 6b present a comparative analysis of U-shape and CNN network architectures for UAV communications at varying distances from the base station. For clarity, other algorithms were omitted from the visualizations.



(a) LoS Condition



(b) NLoS Condition

FIGURE 6: U_shaped and CNN [68] model's Entropy and Non-Entropy analysis using validation data under LoS and NLoS conditions.

The analysis is presented in two distinct scenarios: LoS conditions (upper panel) and NLoS conditions (lower panel). Under LoS conditions, 6a, the U-shape architecture demonstrates high resilience, with accuracy initially climbing from 78% at 100m to peak at 97% at 500m before stabilizing at 96% at 1000m. In contrast, the CNN architecture shows lower overall performance and greater sensitivity to distance, dropping to its minimum of 75% at 500m before partially recovering to 88% at 1000m. In Non-LoS environments 6b, performance degradation is evident for both architectures. The U-shape model maintains superiority with accuracy ranging from 75-90%, while CNN performance declines dramatically with increasing distance, falling to approximately 60% at distances beyond 500m. The integration of entropy-based mechanisms (marked with stars in both panels) proves to be a critical enhancement factor across all tested configurations. For both U-shape and CNN architectures, entropy incorporation consistently yields higher accuracy values compared to their standard counterparts. This improvement is particularly pronounced in challenging scenarios, suggesting that entropy-based approaches effectively capture uncertainty in signal processing, leading to more robust classification performance in variable UAV communication environments. Additionally, the validation metrics (dash-dot lines) typically exceed test performance (solid lines), suggesting some degree of model generalization challenges. These results emphasize the U-shape architecture's superior robustness for UAV communication applications, particularly in challenging long-distance and NLoS environments where reliability is critical, with entropy-based approaches further enhancing performance across all experimental conditions.

E. Alternative Methods and Validation

We also experimented with alternative methods for creating derived signals, such as first and second differentiation. However, PCA applied to these signals resulted in a large number of PCs, with the leading components capturing only a small proportion of the signal variance. Consequently, these methods were not included in our final approach.

The methods we adopted ensured that the new PCs captured over 70% of the signal variance, indicating that the derived features were both informative and valuable. These extra features approach improved the overall model accuracy and performance.

IV. Conclusion

The study presents a novel transformer-based framework for jamming attack identification within UAV-integrated 5G networks, augmented through dimensional reduction via PCA and entropy regularization techniques. The incorporation of a modified cross-entropy objective with entropy-based regularization effectively mitigates prediction overconfidence, yielding substantial enhancements in detection capability. Our architecture attains peak detection efficacy of 85.06% in NLoS environments, with our entropy-enhanced

variants surpassing traditional machine learning approaches like XGBoost by approximately 4.5% and contemporary deep learning methodologies by roughly 2%. The architecture's effectiveness stems from its sophisticated processing of multidimensional wireless indicators, including RSSI and SINR measurements, combined with attention mechanisms that effectively capture temporal correlations within signal patterns.

The comparative analysis demonstrates our approach's resilience across varied propagation conditions. Within challenging NLoS environments, our entropy-enhanced architecture significantly outperforms alternative methodologies, while maintaining competitive performance in LoS scenarios—validating the efficacy of our entropy regularization strategy in complex transmission environments. These findings illuminate the considerable potential of transformer architectures in strengthening wireless security frameworks, particularly for unmanned aerial systems where jamming vulnerabilities continue to proliferate.

Several promising research directions emerge from this work. Future investigations should address computational optimization for deployment in resource-constrained environments, enhance adaptability to emerging jamming strategies, and evaluate applicability within next-generation wireless frameworks. As UAV integration within communication networks accelerates, the demand for sophisticated interference detection systems will intensify correspondingly. While our proposed architecture represents a meaningful advancement in wireless security, continued research remains essential to address evolving threats, strengthen resilience against novel attack vectors, and ensure efficient implementation across heterogeneous operational environments.

Acknowledgment

This work was partly funded by Project SOFIA-AIR (PID2023-147305OB-C31) (MICIU /10.13039/501100011033 / AEI / EFDR, UE).

References

[1] Oussama Bekkouche et al. "A Service-Based Architecture for Enabling UAV Enhanced Network Services". In: *IEEE Network* 34.4 (2020), pp. 328–335. DOI: 10.1109/MNET.001.1900556.

[2] Syed Ahsan Raza Naqvi et al. "Drone-Aided Communication as a Key Enabler for 5G and Resilient Public Safety Networks". In: *IEEE Communications Magazine* 56.1 (2018), pp. 36–42. DOI: 10.1109/MCOM.2017.1700451.

[3] M. Mahdi Azari, Fernando Rosas, and Sofie Pollin. "Cellular Connectivity for UAVs: Network Modeling, Performance Analysis, and Design Guidelines". In: *IEEE Transactions on Wireless Communications* 18.7 (2019), pp. 3366–3381. DOI: 10.1109/TWC.2019.2910112.

[4] Wenbo Jin et al. "Research on Application and Deployment of UAV in Emergency Response". In: *ICEIEC 2020 - Proceedings of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication* (2020), pp. 277–280. DOI: 10.1109/ICEIEC49280.2020.9152338.

[5] Giovanni Geraci et al. "What Will the Future of UAV Cellular Communications Be? A Flight From 5G to 6G". In: *IEEE Communications Surveys and Tutorials* 24.3 (2022), pp. 1304–1335. DOI: 10.1109/COMST.2022.3171135.

[6] Fei Qi et al. "UAV Network and IoT in the Sky for Future Smart Cities". In: *IEEE Network* 33.2 (2019), pp. 96–101. DOI: 10.1109/MNET.2019.1800250.

[7] Lester Ho and Sobia Jangsher. "UAV Trajectory Optimization based on Predicted User Locations". In: *2024 IEEE Wireless Communications and Networking Conference (WCNC)*. 2024.

[8] Boris Galkin et al. "Experimental Evaluation of Air-to-Ground VHF Band Communication for UAV Relays". In: *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2023, pp. 1428–1432. DOI: 10.1109/ICCWorkshops57953.2023.10283656.

[9] Zeeshan Kaleem et al. "UAV-empowered disaster-resilient edge architecture for delay-sensitive communication". In: *arXiv* December (2018), pp. 124–132. DOI: 10.1109/MNET.2019.1800431.

[10] Liang Xiao et al. "User-Centric View of Unmanned Aerial Vehicle Transmission Against Smart Attacks". In: *IEEE Transactions on Vehicular Technology* 67.4 (2018), pp. 3420–3430. DOI: 10.1109/TVT.2017.2785414.

[11] Hoon Lee et al. "UAV-Aided Secure Communications With Cooperative Jamming". In: *IEEE Transactions on Vehicular Technology* 67.10 (2018), pp. 9385–9392. DOI: 10.1109/TVT.2018.2853723.

[12] Paweł Skokowski et al. "Practical Trial for Low-Energy Effective Jamming on Private Networks With 5G-NR and NB-IoT Radio Interfaces". In: *IEEE Access* 12 (2024), pp. 51523–51535. DOI: 10.1109/ACCESS.2024.3385630.

[13] Joseanne Viana et al. "Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation". In: *IEEE Transactions on Vehicular Technology* 73.1 (2024), pp. 131–146. DOI: 10.1109/TVT.2023.3302814.

[14] Donatella Darsena et al. "Detection and Blind Channel Estimation for UAV-Aided Wireless Sensor Networks in Smart Cities Under Mobile Jamming Attack". In: *IEEE Internet of Things Journal* 9.14 (2022), pp. 11932–11950. DOI: 10.1109/JIOT.2021.3132381.

[15] Yang Xin et al. "Machine Learning and Deep Learning Methods for Cybersecurity". In: *IEEE Access* 6

(2018), pp. 35365–35381. DOI: 10.1109/ACCESS.2018.2836950.

[16] Daniel S. Berman et al. "A Survey of Deep Learning Methods for Cyber Security". In: *Information* 10.4 (2019). ISSN: 2078-2489. DOI: 10.3390/info10040122. URL: https://www.mdpi.com/2078-2489/10/4/122.

[17] Qian Mao, Fei Hu, and Qi Hao. "Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey". In: *IEEE Communications Surveys and Tutorials* 20.4 (2018), pp. 2595–2621. DOI: 10.1109/COMST.2018.2846401.

[18] Sheraz Naseer and Yasir Saleem. "Enhanced network intrusion detection using deep convolutional neural networks". In: *KSII Transactions on Internet and Information Systems* 12.10 (2018), pp. 5159–5178. ISSN: 22881468. DOI: 10.3837/tiis.2018.10.028.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[20] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357.

[21] Bendong Zhao et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.

[22] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review". en. In: *Data Min. Knowl. Discov.* 33.4 (July 2019), pp. 917–963.

[23] Hamed Farkhari et al. "A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms". In: *IEEE Access* 12 (2024), pp. 178898–178908. DOI: 10.1109/ACCESS.2024.3504729.

[24] Yang Ju et al. "A Joint Jamming Detection and Link Scheduling Method Based on Deep Neural Networks in Dense Wireless Networks". In: *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*. 2019, pp. 1–5. DOI: 10.1109/VTCFall.2019.8891535.

[25] Fu Ruo-Ran. "Compound Jamming Signal Recognition Based on Neural Networks". In: *2016 Sixth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*. 2016, pp. 737–740. DOI: 10.1109/IMCCC.2016.163.

[26] Jian-Cong Li et al. "Jamming Identification for GNSS-based Train Localization based on Singular Value Decomposition". In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, pp. 905–912. DOI: 10.1109/IV48863.2021.9575412.

[27] Mehmet Ali Aygül et al. "Deep Learning-Assisted Detection of PUE and Jamming Attacks in Cognitive Radio Systems". In: *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. 2020, pp. 1–5. DOI: 10.1109/VTC2020-Fall49728.2020.9348579.

[28] Youness Arjoune et al. "A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication". In: *2020 International Conference on Information Networking (ICOIN)*. 2020, pp. 459–464. DOI: 10.1109/ICOIN48656.2020.9016462.

[29] Mohammadreza Amini et al. "Deep Fusion Intelligence: Enhancing 5G Security Against Over-the-Air Attacks". In: *IEEE Transactions on Machine Learning in Communications and Networking* 3 (2025), pp. 263–279. DOI: 10.1109/TMLCN.2025.3533427.

[30] Yuchen Li et al. "Jamming Detection and Classification in OFDM-Based UAVs via Feature- and Spectrogram-Tailored Machine Learning". In: *IEEE Access* 10 (2022), pp. 16859–16870. DOI: 10.1109/ACCESS.2022.3150020.

[31] Marouane Hachimi et al. "Multi-stage Jamming Attacks Detection using Deep Learning Combined with Kernelized Support Vector Machine in 5G Cloud Radio Access Networks". In: *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. 2020, pp. 1–5. DOI: 10.1109/ISNCC49221.2020.9297290.

[32] Ali Krayani et al. "Automatic Jamming Signal Classification in Cognitive UAV Radios". In: *IEEE Transactions on Vehicular Technology* 71.12 (2022), pp. 12972–12988. DOI: 10.1109/TVT.2022.3199038.

[33] Amiya Kumar Sahu et al. "Internet of Things Attack Detection Using Hybrid Deep Learning Model". In: *Computer Communications* (2021).

[34] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[35] Chun-Jie Chiu Kai-Jui Chen An-Hung Hsiao and Kai-Ten Feng. "Self-Attention based Semi-Supervised Learning for Time-varying Wi-Fi CSI-based Adjoining Room Presence Detection". In: *IEEE 95th Vehicular Technology Conference:, VTC Spring 2022, Helsinki, Finland, 19-22 June, 2022* (2022).

[36] Ibrahim Elleuch, Ali Pourranjbar, and Georges Kaddoum. "Convolutional Models for Anti-Jamming in Heavily Attacked UAV Environments". In: *IEEE Open Journal of the Communications Society* 5 (2024), pp. 5337–5347. DOI: 10.1109/OJCOMS.2024.3451288.

[37] M Liyanage et al. "A Comprehensive Guide to 5G Security". In: Hoboken, NJ, USA: Wiley, 2018. DOI: 10.1002/9781119293071.ch6.

[38] Peter Schneider and Günther Horn. "Towards 5G Security". In: *2015 IEEE Trustcom/BigDataSE/ISPA*. Vol. 1. 2015, pp. 1165–1170. DOI: 10.1109/Trustcom.2015.499.

[39] Kang Xue et al. "Performance and Reliability of 5G Communications for USV-UAV Critical Applications". In: *2023 17th European Conference on Antennas and Propagation (EuCAP)*. 2023, pp. 1–5. DOI: 10.23919/EuCAP57121.2023.10132977.

[40] Xingqin Lin. "An Overview of 5G Advanced Evolution in 3GPP Release 18". In: *IEEE Communications Standards Magazine* 6.3 (2022), pp. 77–83.

[41] Nishat I. Mowla et al. "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET". In: *Journal of Communications and Networks* 22.3 (2020), pp. 244–258. DOI: 10.1109/JCN.2020.000015.

[42] Haoran Sun et al. "Learning to Optimize: Training Deep Neural Networks for Interference Management". In: *IEEE Transactions on Signal Processing* 66.20 (2018), pp. 5438–5453. DOI: 10.1109/TSP.2018.2866382.

[43] Michael Greenacre, Patrick J.F. Groenen, Trevor Hastie, et al. "Principal component analysis". In: *Nature Reviews Methods Primers* 2 (2022), p. 100. DOI: 10.1038/s43586-022-00184-w. URL: https://doi.org/10.1038/s43586-022-00184-w.

[44] Hamed Farkhari et al. "New PCA-based Category Encoder for Efficient Data Processing in IoT Devices". In: *2022 IEEE Globecom Workshops (GC Wkshps)*. 2022, pp. 789–795. DOI: 10.1109/GCWkshps56602.2022.10008757.

[45] Bin Li, Zesong Fei, and Yan Zhang. "UAV Communications for 5G and Beyond: Recent Advances and Future Trends". In: *IEEE Internet of Things Journal* 6.2 (2019), pp. 2241–2263. DOI: 10.1109/JIOT.2018.2887086.

[46] Ning Gao et al. "Anti-Intelligent UAV Jamming Strategy via Deep Q-Networks". In: *IEEE Transactions on Communications* 68.1 (2020), pp. 569–581. DOI: 10.1109/TCOMM.2019.2947918.

[47] Hamed Farkhari et al. *Accurate and Reliable Methods for 5G UAV Jamming Identification With Calibrated Uncertainty*. 2022. DOI: 10.48550/ARXIV.2211.02924.

[48] Na Liu et al. "A DNN Framework for Secure Transmissions in UAV-Relaying Networks with a Jamming Receiver". In: *2020 IEEE 20th International Conference on Communication Technology (ICCT)*. 2020, pp. 703–708. DOI: 10.1109/ICCT50939.2020.9295902.

[49] Detao Su and Meiguo Gao. "Research on Jamming Recognition Technology Based on Characteristic Parameters". In: *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*. 2020, pp. 303–307. DOI: 10.1109/ICSIP49896.2020.9339393.

[50] Thuy T.T. Nguyen and Grenville Armitage. "A survey of techniques for internet traffic classification using machine learning". In: *IEEE Communications Surveys and Tutorials* 10.4 (2008), pp. 56–76. DOI: 10.1109/SURV.2008.080406.

[51] Anna Sperotto et al. "An Overview of IP Flow-Based Intrusion Detection". In: *IEEE Communications Surveys and Tutorials* 12.3 (2010), pp. 343–356. DOI: 10.1109/SURV.2010.032210.00054.

[52] Paweł Skokowski et al. "Jamming and jamming mitigation for selected 5G military scenarios". In: *ICMCIS*. 2022. URL: https://api.semanticscholar.org/CorpusID:252471854.

[53] Luis Bastos et al. "Potential of 5G technologies for military application". In: *2021 International Conference on Military Communication and Information Systems (ICMCIS)*. 2021, pp. 1–8. DOI: 10.1109/ICMCIS52405.2021.9486402.

[54] James F. Harvey, Michael B. Steer, and Theodore S. Rappaport. "Exploiting High Millimeter Wave Bands for Military Communications, Applications, and Design". In: *IEEE Access* 7 (2019), pp. 52350–52359. DOI: 10.1109/ACCESS.2019.2911675.

[55] Hongyue Kang et al. "Improving Dual-UAV Aided Ground-UAV Bi-Directional Communication Security: Joint UAV Trajectory and Transmit Power Optimization". In: *IEEE Transactions on Vehicular Technology* 71.10 (2022), pp. 10570–10583. DOI: 10.1109/TVT.2022.3184804.

[56] Xiucheng Wang et al. "Joint Flying Relay Location and Routing Optimization for 6G UAV–IoT Networks: A Graph Neural Network-Based Approach". In: *Remote Sensing* 14.17 (2022). ISSN: 2072-4292. DOI: 10.3390/rs14174377. URL: https://www.mdpi.com/2072-4292/14/17/4377.

[57] Joseanne Viana et al. *A Synthetic Dataset for 5G UAV Attacks Based on Observable Network Parameters*. 2022. DOI: 10.48550/ARXIV.2211.09706. URL: https://arxiv.org/abs/2211.09706.

[58] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[59] Biao Zhang and Rico Sennrich. "Root mean square layer normalization". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 12375–12384. DOI: 10.48550/arXiv.1910.07467.

[60] Noam Shazeer. *GLU variants improve transformer*. 2020. DOI: 10.48550/arXiv.2002.05202. arXiv: 2002.05202 [cs.LG].

[61] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning". In: *Neural Networks* 107 (2017), pp. 3–11. DOI: 10.1016/j.neunet.2017.12.012.

[62] Tianzhu Ye et al. "Differential Transformer". In: *The Thirteenth International Conference on Learning Representations*. 2025. URL: https://openreview.net/forum?id=OvoCm1gGhN.

[63] Lukas Balles, Jaap Romijnders, and Philipp Hennig. "Coupling adaptive batch sizes with learning rates". In: *arXiv preprint arXiv:1612.05086* (2017).

[64] Boris T Polyak and Anatoli B Juditsky. "Acceleration of stochastic approximation by averaging". In: *SIAM Journal on Control and Optimization* 30.4 (1992), pp. 838–855.

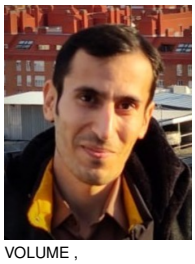
[65] Tuomas Haarnoja et al. "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor". In: *International Conference on Machine Learning (ICML)* (2018).

[66] Alex Kendall and Yarin Gal. "What uncertainties do we need in bayesian deep learning for computer vision?" In: *Advances in neural information processing systems* (2017).

[67] Gabriel Pereyra et al. "Regularizing neural networks by penalizing confident output distributions". In: *ICLR* (2017).

[68] Qinzhe Lv et al. "Deep Neural Network-Based Interrupted Sampling Deceptive Jamming Countermeasure Method". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 15 (2022), pp. 9073–9085. DOI: 10.1109/JSTARS.2022.3214969.

Joseanne Viana is a Ph.D. researcher at UC3M - Charles III University of Madrid. She received her bachelor's degree in telecommunication engineering from the University of Campinas, Brazil. She is an Early-Stage Researcher in the project TeamUp5G, a European Training Network under the MSCA ITN of the European Commission's Horizon 2020. Her research interests include wireless communications applied to interconnected systems such as UAVs, aerial vehicles, and non-terrestrial devices.



Hamed Farkhari is a Ph.D. researcher at ISCTE - Lisbon University Institute. He serves as an Early-Stage Researcher in the TeamUp5G group, a European Training Network under the Marie Skłodowska-Curie Actions (MSCA ITN) of the European Commission's Horizon 2020 program. His research interests and work focus on cybersecurity, machine learning, deep learning, data science, meta-heuristic techniques, and optimization algorithms.

Pedro Sebastião received the Ph.D. degree in electrical and computer engineering from IST.He is currently a Professor with ISCTE-IUL's Information Science and Technology Department, He is also the Board Director of AUDAX-ISCTE - Entrepreneurship and Innovation Center, ISCTE, responsible for the LABS LISBOA Incubator and Researcher at the Institute of Telecommunications. He has oriented several master's dissertations and doctoral theses. It has several scientific, engineering and pedagogical awards. Also, he has organized or co-organized more than 55 national and international scientific conferences. He planned and developed several postgraduate courses in technologies and management, entrepreneurship and innovation and transfer of technology and innovation. He has supported several projects involving technology transfer and creation of start-ups and spinoffs of value to society and market. He developed his professional activity in the National Defense Industries, initially in the Office of Studies and later as the Board Director of the Quality Department of the Production of New Products and Technologies. He was also responsible for systems of communications technology in the Nokia-Siemens business area. His main researching interests are in monitoring, control and communications of drones, unmanned vehicles, planning tools, stochastic process (modeling and efficient simulations), the Internet of Things, and efficient communication systems.

Víctor P. Gil Jiménez (Senior Member, IEEE) received a B.S. degree (Hons.) in Telecommunications from the University of Alcalá in 1998 and an M.S. degree (Hons.) in Telecommunications and a Ph.D. degree (Hons.) from Universidad Carlos III de Madrid in 2001 and 2005, respectively. In 1999, he was a Communications Staff member with the Spanish Antarctica Base. He visited the University of Leeds, U.K., in 2003, Chalmers Technical University, Sweden, in 2004, and the Instituto de Telecomunicações, Portugal, from 2008 to 2010. He is an Associate Professor with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid. He has led several private and national Spanish projects and participated in various European and international projects. He holds one patent and has published over 80 journal articles/conference papers and nine book chapters. His research interests include advanced multicarrier systems for wireless radio, satellite, and visible light communications. He was the IEEE Spanish Communications and Signal Processing Joint Chapter Chair from 2015 to 2023. He received the Master Thesis Award and the Ph.D. Thesis Award from the Professional Association of Telecommunication Engineers of Spain in 1998 and 2006, respectively.

2.2. Article #2: A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms

This article presents a novel hybrid approach to jamming detection in UAV communications, combining DNNs with ML algorithms. The primary objective was to develop a reliable detection framework capable of enhancing the accuracy and reliability of binary classification DNNs by effectively managing uncertainty levels. This research introduced innovative preprocessing and post-processing techniques, which paved the way for more robust jamming detection methodologies in UAV communications.

The key contribution of this work to the present thesis lies in the development of the hybrid detection framework itself, as well as the foundational principles underlying its design. Specifically, the research demonstrated that by strategically integrating DNNs with ML algorithms and employing novel preprocessing and post-processing techniques, significant improvements in jamming detection reliability could be achieved. A particularly noteworthy contribution was the introduction of calibration error metrics, confidence values, and the RS, which quantifies the disparity between Mean Accuracy (MA) and Mean Confidence (MC).

The effectiveness of these methods was demonstrated through application to simulated real-world scenarios, showcasing improvements in jamming detection reliability for UAV communications. The proposed algorithms were rigorously evaluated against baseline DNNs and DNNs enhanced with the eXtreme Gradient Boosting (XGB) classifier, providing robust validation of the hybrid approach.

Article Details

- **Title:** A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms
- **Date:** 2024
- **Authors:** Hamed Farkhari, , Joseanne Viana, Sarang KahVazadeh Pedro Sebastião, Victor P. Gil Jimenez, Rui Dinis
- **Status:** Accepted in a major international journal with rigorous peer review
- **Journal:** IEEE Access
- **DOI:** 10.1109/ACCESS.2024.3504729

The significance of this paper lies in its introduction of a comprehensive hybrid approach that addresses key challenges in UAV jamming detection while maintaining practical implementation feasibility. Furthermore, this work established fundamental concepts that would be expanded upon in subsequent research, particularly in the areas of attention mechanisms and uncertainty management in detection systems.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 October 2024, accepted 11 November 2024. Date of publication 00 xxxx 0000, date of current version 00 xxxx 0000.

Digital Object Identifier 10.1109/ACCESS.2024.3504729

A Hybrid Approach to Reliable Jamming Identification in UAV Communications Using Combined DNNs and ML Algorithms

HAMED FARKHARI[1,*], (Member, IEEE), JOSEANNE VIANA[2,*], (Member, IEEE), SARANG KAHVAZADEH[3], (Member, IEEE), PEDRO SEBASTIÃO[1,4], (Member, IEEE), VICTOR P. GIL JIMENEZ[2], (Senior Member, IEEE), AND RUI DINIS[4,5], (Senior Member, IEEE)

[1]ISCTE–Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal
[2]Departamento de Teoría de la Señal y Comunicaciones, Universidad Carlos III de Madrid (UC3M), 28903 Madrid, Spain
[3]Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), CERCA, 08860 Barcelona, Spain
[4]Instituto de Telecomunicações (IT), 1049-001 Lisbon, Portugal
[5]FCT, Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal

Corresponding author: Hamed Farkhari (hamed_farkhari@iscte-iul.pt)

This work was supported in part by the European Union's Horizon 2020 Research and Innovation Programme under the Marie Sklodowska-Curie under Project 813391; in part by the Ministerio de Asuntos Económicos y Transformación Digital (MINECO), in part by the European Union (EU)-NextGenerationEU in the Frameworks of the "Plan de Recuperación, Transformación y Resiliencia" and of the "Mecanismo de Recuperación y Resiliencia" under Grant TSI-063000-2021-55 and Grant PID2021-126431OB-I00; in part by the Ministerio de Ciencia, Innovación y Universidades (MCIN)/Agencia Española de Investigación (AEI)/10.13039/501100011033; in part by the "European Regional Development Fund (ERDF) A way of making Europe" and Generalitat de Catalunya under Grant 2021 SGR 00770; in part by Project "SOFIA-AIR" PID2023-147305OB-C31, Ministerio de Ciencia, Innovación y Universidades (MICIU)/AEI/10.13039/501100011033/Ministerio de Ciencia, Innovación y Universidades (FEDER) EU, in part by FCT—Fundação para a Ciência e Tecnologia, I.P., and Instituto de Telecomunicações under Project UIDB/50008/2020, with DOI identifier https://doi.org/10.54499/UIDB/50008/2020; and in part by the Distributed Access Design for Cell-less Smart 6G Networks (CELL-LESS6G) project, 2022. 08786.PTDC with DOI identifier https://doi.org/10.54499/2022.08786.PTDC.

*Hamed Farkhari and Joseanne Viana contributed equally to this work.

ABSTRACT Deep Neural Networks (DNNs) have gained prominence due to their remarkable accomplishments across various domains, including telecommunications and security. Their integration into decision-making processes within 5G telecommunication systems and UAV security is noteworthy. However, the iterative nature of DNN data processing can introduce uncertainties in classification decisions, impacting their reliability. This paper presents novel combined preprocessing and post-processing techniques designed to enhance the accuracy and reliability of binary classification DNNs by managing uncertainty levels. The study evaluates these methods through calibration error metrics, confidence values, and the Reliability Score (RS), which quantifies the disparity between Mean Accuracy (MA) and Mean Confidence (MC). Additionally, the effectiveness of these methods is demonstrated by applying them to simulated real-world scenarios to improve jamming detection reliability in UAV communications. The proposed algorithms' impact is compared against baseline DNNs and DNNs augmented with the eXtreme Gradient Boosting (XGB) classifier, as well as the latest research to validate our approach. This paper comprehensively overviews the experimental setup, dataset, deep network architecture, preprocessing and post-processing techniques, evaluation metrics, and results. By addressing uncertainty in XGB and DNN outputs, this study improves the trustworthiness of ML-DNN-based decision-making processes in 5G UAV security scenarios.

INDEX TERMS Unmanned aerial vehicle, deep neural networks, machine learning, uncertainty, reliability, jamming identification, eXtreme gradient boosting (XGB) classifier, 5G, 6G.

The associate editor coordinating the review of this manuscript and approving it for publication was Xiao-Sheng Si.

I. INTRODUCTION

Deep Neural Networks (DNNs) have gained significant prominence due to their remarkable achievements across

various domains, including telecommunications and security [1], [2], [3]. These models are increasingly integrated into decision-making processes within 5G telecommunication systems and Unmanned Aerial Vehicle (UAV) [4] security. Notably, Machine Learning (ML) mechanisms, including DNNs, are anticipated to be incorporated into the standards of 6G telecommunication systems [1]. Extensive research has also focused on leveraging deep learning for decision-making in the physical layer [2]. In the 5G UAV security realm, DNNs offer capabilities such as universal function approximation, exceptional logic for complex time series modeling challenges, and potential for parallel data processing, contingent upon their design [5], [6]. However, the iterative nature of DNNs' data processing during classification tasks can lead to output probabilities accompanied by uncertainties, raising concerns regarding the reliability of classification decisions. Addressing the calibration of DNNs to ensure high accuracy and reliable output decisions is critical, as discussed in [7]. The authors present various calibration techniques that enhance these parameters by leveraging well-known datasets such as CIFAR-10 and ImageNet and pre-trained DNNs such as ResNet, WideNet, and LeNet. As augmentation techniques are integrated into the original data preprocessing stage, understanding concepts such as risk, uncertainty, and trust in a model's output becomes increasingly vital. In [8], the authors propose that preprocessing and post-processing techniques can enhance DNNs' performance. Furthermore, they introduce methods that improve classification accuracy while reducing uncertainty, accompanied by mathematical approaches to compute metrics like Expected Calibration Error (ECE) and Maximum Calibration Error (MCE). In this paper, we present novel combined preprocessing and post-processing techniques to enhance the accuracy and reliability of binary classification DNNs by managing uncertainty levels. Our main contributions are as follows:

- We introduce combined preprocessing and post-processing algorithms that improve the reliability and accuracy of ML-DNN outputs for jamming identification in 5G UAV scenarios.
- We provide a comprehensive overview of the experimental setup, dataset, deep network architecture, preprocessing and post-processing techniques, evaluation metrics, and results, highlighting improvements in trustworthiness for DNN-based decision-making processes in 5G UAV security scenarios.
- We propose a Time-Series Augmentation (TSA) technique as part of the preprocessing phase, generating diverse versions of each sample to provide diversity for post-processing techniques.
- We evaluate the effectiveness of our proposed methods through calibration error metrics, confidence values, and the Reliability Score (RS), quantifying the disparity between Mean Accuracy (MA) and Mean Confidence (MC).
- We demonstrate that the proposed methods can be directly applied to real-world scenarios to enhance the reliability of jamming detection in UAV communications. A comparative analysis is performed against baseline DNN and Enhanced ML-DNN using the eXtreme Gradient Boosting (XGB) classifier.

The structure of the paper is as follows: Section II provides a comprehensive description of all the components involved in the experiment. The dataset is explained in Subsection II-A. In Subsection II-B, the deep network used in the study is discussed. Subsection III delves into the combined preprocessing and post-processing techniques and elucidates how they enhance reliability and accuracy. Subsection III-B presents the metrics employed to evaluate the reliability of each method. Section IV presents the results for the proposed system. Finally, Section V summarizes the main conclusions drawn from the study and presents some topics for future work.

II. SYSTEM MODEL

We consider the DNN jamming classification system based on the Received Signal Strength Indicator (RSSI) and Signal to Interference-plus-Noise Ratio (SINR) signals as described in [9]. The scenario depicted in Figure 1 involves up to four attackers randomly positioned within a designated area, while the UAV maintains a connection to the base station via 5G.

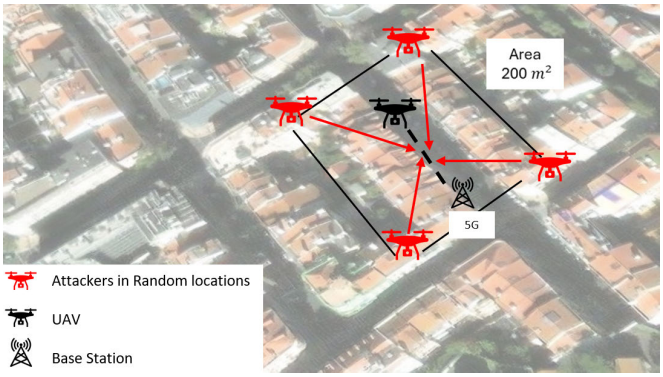


FIGURE 1. Jamming scenario.

The dataset utilized for this research is sourced from the same reference, and the classification is conducted using either a designed DNN or an XGB classifier algorithm. This system enhances accuracy and reliability by integrating preprocessing and post-processing algorithms, as illustrated in Figure 2. The proposed system comprises four main components, arranged from the upper left to right in the figure: the preprocessing algorithm, the primary classifier, the post-processing algorithms, and the auxiliary classifier. The preprocessing algorithm processes the input sample $sample_i$ and augmented samples generated via the TSA technique. The post-processing algorithm incorporates Methods 1, 2, and 3, each with their respective auxiliary algorithms. The system includes the "No Method" block for comparative purposes. At the end of the post-processing algorithm, we perform feature classification and independently evaluate the accuracy and reliability of each algorithm by analyzing the classification results from the primary and auxiliary

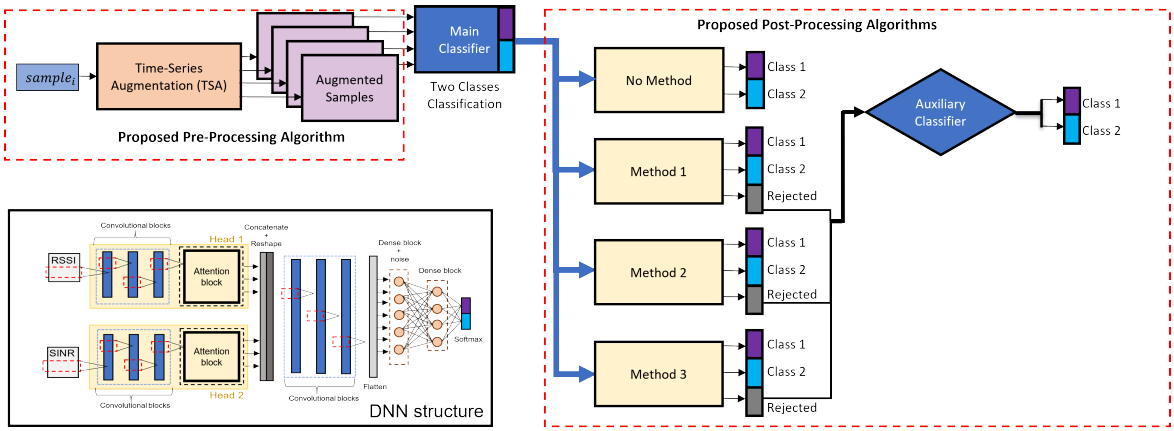


FIGURE 2. Schematic representation of the integrated preprocessing and post-processing algorithms, in which the DNN is the primary algorithm, and ML is the supporting algorithm. The DNN architecture is based on the work by [9].

classifiers. In our study, we employ the XGB classifier algorithm due to its superior performance in detecting the presence or absence of attacks across various scenarios and configurations, as detailed in the dataset discussed in Section II-A and described in [9] and [10].

A. DATASET

Our dataset consists of two signal parameters: RSSI and SINR measurements. These measurements are collected when an authenticated UAV connects to a small cell through the 5G communication system while being subjected to power attacks from other UAVs. Both values were collected from NS3 5G-Lena Simulator in [11]. We transform these two time series variables into supervised samples by employing a rolling window of 300 values. Additionally, other terrestrial users are connected to the network. Consequently, the measured parameters in the authenticated UAV exhibit variations as interference from other devices fluctuate. The dataset includes up to four attackers and 30 terrestrial users connected simultaneously. As part of our research, we are also investigating and analyzing other open-source datasets, including WSN-DS [12] and [13], to obtain a diverse dataset with mature data. Further details regarding the dataset construction and its potential applications can be found in [9] and [14].

B. DEEP NETWORK ARCHITECTURE

In this paper, our primary focus is to conduct a comprehensive analysis of the confidence values within the XGB and deep network, particularly assessing its reliability. Detailed discussions regarding the design and parameters of the DNN utilized in this study are available in previous works [9], [10].

III. DESIGNED SOLUTION

This section presents an overview of the proposed preprocessing and post-processing algorithms, precisely Method 1, Method 2, and Method 3, as well as the implementation of the overall solution.

1) PREPROCESSING TECHNIQUE

In the TSA technique, the time series sequence of each sample is inverted to generate a new augmented sample. For example, the RSSI and SINR will include the original sequence and its inverted counterpart within an appropriate rolling window. During the preprocessing phase, each variable's original and inverted sequences are combined to create four new samples. Generally, the technique can generate $2^{N_{variables}}$ augmented samples, where $N_{variables}$ is the number of variables the DNN uses as input. All original and augmented samples are used to train our ML and DNN, as depicted in Figure 2. After the primary classifier processes the augmented version of each sample, the results are utilized as inputs in the post-processing algorithms. Table 1 provides an example of generating the four new augmented samples using the preprocessing algorithm.

TABLE 1. Output of the TSA.

Sample	Sequence 1 (RSSI)	Sequence 2 (SINR)
Sample 1	Same	Same
Sample 2	Same	Flipped
Sample 3	Flipped	Same
Sample 4	Flipped	Flipped

2) POST-PROCESSING METHODS

In the preprocessing step of the proposed method, the TSA technique is applied to each instance, generating all augmented samples as outlined in Table 1. Subsequently, the primary classifier produces output for each sample, including its augmented versions. This paper introduces three post-processing techniques for the proposed system designated Methods 1, 2, and 3 (M1, M2, and M3). Each method leverages the main classifier outputs of all augmented samples generated in the preprocessing phase, as described in subsection III-1, to evaluate the classification output. These post-processing techniques aim to enhance the primary classifier's classification accuracy and reliability, providing greater flexibility in selecting the optimal approach for a given classification task. Method 1, Method 2, and Method 3 utilize the outputs from the primary classifier to classify each augmented sample into Class 1, Class 2, or Rejected. For samples rejected by Methods 1, 2, or 3, an auxiliary classifier is employed to refine these classifications further, thereby

improving overall accuracy and reliability. The auxiliary classifier processes only the rejected results to produce the final classification output. Each classification occurs inside an interval range defined in Section III-3.

3) FILTERS INTERVAL

The primary classifier produces probability outputs for N classes. A filter range encompassing all N probability values is defined for post-processing analysis. In binary classification, a sample is considered accepted if its confidence value exceeds the upper limit of the filter interval and the probability of other classes is below the lower limit of the filter interval. Conversely, a sample is rejected if at least one of the output vector probabilities falls within the filter range. Figure 3a shows the proportion of accepted and rejected samples per filter interval, while Figure 3b illustrates the accuracy of accepted samples after applying the filter. Adopting this filtering approach, the primary classifier can produce more refined outputs and ensure high-quality classification results. This method also facilitates the identification of samples that meet specific criteria, offering a more nuanced evaluation of model performance. In all figures, utilizing a filter range of 0.5-0.5 where the rejected samples are 0% signifies the exclusive application of methods on the main algorithm without conditions for rejecting samples. Consequently, no accepted samples are transmitted to the second auxiliary algorithm.

4) METHOD 1

In Method 1, the initial step identifies whether a sample is accepted or rejected and assigns the corresponding class label if accepted. This step involves checking the filter interval conditions for all four samples. If all four augmented versions are rejected, the sample is classified as rejected, and the process terminates. The output of Method 1 is the average probability per class for the accepted samples. The filter interval condition for this output is then re-evaluated, and the result is categorized as accepted or rejected. Finally, the accepted samples are classified into their respective classes, while the rejected samples are forwarded to an auxiliary classifier for final classification as in algorithm 1.

Algorithm 1 Method 1

Require: $0 \leq yp_{ij} \leq 1$ for $j \in \{1, 2, 3, 4\}, i \in 1, \ldots, N$
Ensure: $Accepted \mid\mid Rejected \leftarrow$ Assign yp_{ij}
  $(\beta_1, \beta_2) \leftarrow \beta$ filter range
  if $yp_{ij} \leq \beta_1 \| \beta_2 \leq yp_{ij}, \forall j \in \{1, 2, 3, 4\}$ **then**
    $yp_i \leftarrow Average_j(yp_{ij})$
  end if
  if $yp_i \leq \beta_1 \| \beta_2 \leq yp_i$ **then**
    $Accepted \leftarrow yp_i$
  else
    $Rejected \leftarrow yp_i$
  end if

5) METHOD 2

Method 2's filter interval conditions for each augmented output are not individually checked. Instead, the average probability per class is computed across all four results. This average is then evaluated against the filter interval conditions to determine whether the sample should be accepted or rejected. This approach contrasts with Method 1, in which the filter interval conditions for each augmented output are matched before computing the average probability per class for the accepted samples. The detail of Method 2 is presented in algorithm 2.

Algorithm 2 Method 2

Require: $0 \leq yp_{ij} \leq 1$ for $j \in \{1, 2, 3, 4\}, i \in 1, \ldots, N$
Ensure: $Accepted \mid\mid Rejected \leftarrow$ Assign yp_{ij}
  $(\beta_1, \beta_2) \leftarrow \beta$ filter range
  $yp_i \leftarrow Average_j(yp_{ij}), \forall j \in \{1, 2, 3, 4\}$
  if $yp_i \leq \beta_1 \| \beta_2 \leq yp_i$ **then**
    $Accepted \leftarrow yp_i$
  else
    $Rejected \leftarrow yp_i$
  end if

6) METHOD 3

Method 3, as detailed in Algorithm 3, explores three distinct techniques. The first technique, M3-max, involves selecting the output with the highest confidence. The second strategy, M3-min, applies a minimum trust threshold to reject samples with low confidence when integrating this method into the primary algorithm. In M3-N, augmented samples with the highest and lowest confidence values are initially eliminated due to concerns of overconfidence and underconfidence. Subsequently, the remaining augmented versions are averaged. Finally, averaging is performed over all enriched samples without removal when no augmented outputs remain after excluding high and low-confidence results. All techniques are summarized in Table 2. Section IV presents the results of Method 3.

7) CONFIDENCE VALUES

In DNNs, the softmax layer is typically employed as the output layer to generate a probability distribution over a set of classes for each input sample. The resulting output is a one-hot encoded vector, where each element represents the probability of the corresponding category. To determine the correct class for each sample, only the maximum value in the vector is considered as one, indicating the most probable class, while the rest are rounded to zero. Our study posits that the confidence level of a sample's classification in N-class problems can be determined by considering the maximum probability value of its corresponding one-hot vector. This approach results in confidence values ranging from 0.5 to 1 in binary classification. A minor difference between the confidence score and its rounded

Algorithm 3 Method 3

Require: $0 \leq yp_{ij} \leq 1$ for $j \in \{1, 2, 3, 4\}$, $i \in 1, \ldots, N$
Ensure: *Accepted* || *Rejected* || *Output* $\leftarrow$ Assign yp_{ij}
  $(\beta_1, \beta_2) \leftarrow \beta$ filter range
  $conf_b \leftarrow$ confidence base
  if $conf_b$ is Max **then** ▷ Method 3-max
    $yp_i \leftarrow \arg\max confidence_j(yp_{ij}), \forall j \in \{1, 2, 3, 4\}$
  else if $conf_b$ is Min **then** ▷ Method 3-min
    $yp_i \leftarrow \arg\min confidence_j(yp_{ij}), \forall j \in \{1, 2, 3, 4\}$
  else ▷ Method 3-N
    $j_1 \leftarrow \arg\max confidence_j(yp_{ij})$
    $j_2 \leftarrow \arg\min confidence_j(yp_{ij})$
    $yp_i \leftarrow Average_j(yp_{ij}), \forall j \in \{1, 2, 3, 4\} - \{j_1, j_2\}$
    if yp_i is {} **then**
      $yp_i \leftarrow Average_j(yp_{ij}), \forall j \in \{1, 2, 3, 4\}$
    end if
  end if
  if β filter defined **then**
    if $yp_i \leq \beta_1 \| \beta_2 \leq yp_i$ **then**
      $Accepted \leftarrow yp_i$
    else
      $Rejected \leftarrow yp_i$
    end if
  else
    $Output \leftarrow yp_i$
  end if

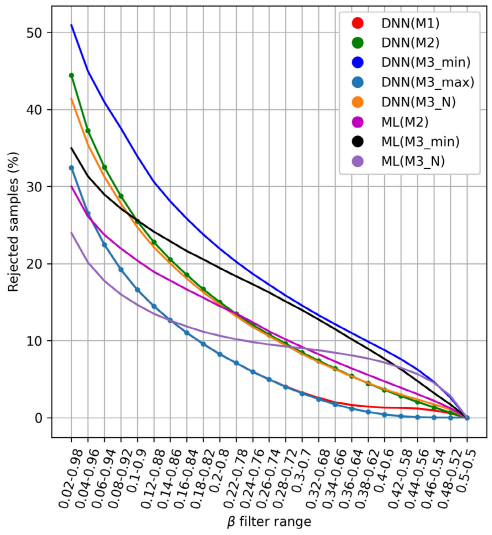
value indicates higher reliability for correctly classified samples. By leveraging this confidence value, one can assess the quality of the classification output and evaluate the classifier's performance.

8) OVERFITTING-UNDERFITTING OVER SAMPLES

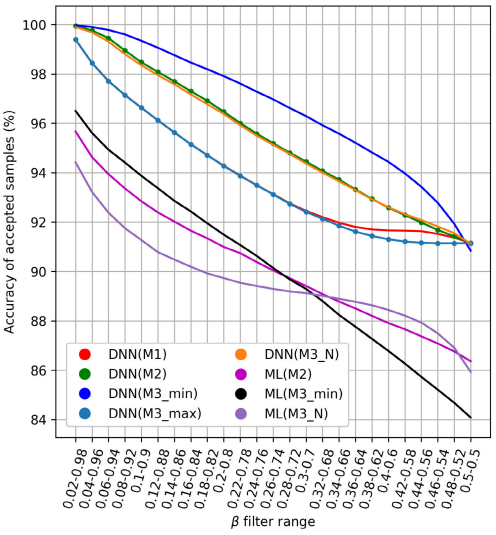
One of the motivations for introducing M3-N is the observed impact of M3-Max and M3-Min on our experimental results. While M3-Max involves selecting samples with the highest confidence, it does not improve accuracy. Instead, it leads to unstable uncertainty when combined with different algorithms, as evidenced in Table 2. Conversely, M3-Min results in a significant decrease in the quality of the final results, suggesting overfitting on samples with the highest confidence and underfitting on instances with the lowest confidence. M3-N is an initial step in studying this effect and attempts to enhance reliability by mitigating this phenomenon. Further research is required on different datasets and deep networks with equitable augmentation on sampling. We propose using proper augmentation, where all augmented versions contain similar information, such as the flipping technique, and avoiding using lossy augmentation like cropping to study this phenomenon.

A. ALGORITHM COUPLING AND FINAL SETUP

The final classification results are obtained by integrating the outputs of Deep Neural Networks (DNN) and Machine Learning (ML) using the aforementioned methods. Two



(a)



(b)

FIGURE 3. Flexible ranges with a resolution of 0.02 between the accuracy of accepted samples as classes 1 or 2 and the portion of rejected samples based on the β filter ranges: (a) Samples versus β (b) Accuracy versus β. The results of ML(M1) and ML(M3_max) were removed due to the worst achievement.

distinct scenarios are considered. In the first scenario, a fast ML algorithm is the primary classifier, while the DNN is an auxiliary classifier. Conversely, in the second scenario, the DNN is the primary classifier, with the ML algorithm acting as an additional classifier. During our experiments, the second scenario neither yielded significant improvements nor justified the additional complexity introduced to the final design. After thorough analysis, this scenario was ultimately discarded due to its suboptimal results. The rationale for the first scenario is that a faster ML algorithm can efficiently handle simple samples. If the ML algorithm lacks confidence, the more precise DNN can manage the complex samples, ensuring a balance between speed and accuracy. Various metrics are evaluated in the context of the following hybrid combinations:

- The output of only the main classifier as a baseline;
- The output of the primary classifier after applying either Method 1, Method 2, or Method 3, followed by using a secondary classifier for the rejected samples.

We have implemented the trained machine learning algorithms within the UAV to meet the fast-processing requirements essential for enabling the UAV to make timely and effective decisions while accomplishing its mission.

B. EVALUATION METRICS

We employ widely recognized metrics proposed by [7] to assess the model's uncertainty, accuracy, and quality, enabling a comparison of the improvements made by different methods. The following section provides a detailed explanation of these metrics:

1) ACCURACY PER CONFIDENCE

The visual representation of this metric is utilized to analyze the calibration and uncertainty characteristics of the DNN model. This chart, commonly called the "reliability diagram" by the authors in [15] and [16], evaluates the model's reliability. The metric is computed by partitioning the samples into groups based on their confidence values within specified interval ranges and subsequently estimating the accuracy of each group. Our deep network architecture employs a one-hot encoding output scheme with a softmax activation function and binary cross-entropy loss function. Given that the DNN under study produces results in one-hot encoding probabilities, the maximum probability value among the predicted output classes is assigned as the confidence score. The confidence values are then grouped within interval ranges from 0.5 to 1, with each interval defined by the user.

2) MEAN CONFIDENCE AND MEAN ACCURACY

These metrics, Mean Confidence (MC) and Mean Accuracy (MA), represent the total weighted average of confidence and accuracy for the number of samples within each confidence interval. In a fair scenario concerning reliability and accuracy, these two values should be equal. However, in DNN architectures, it is often observed that these values tend to exhibit biases towards one extreme or the other. Over-confidence arises when the Mean Confidence surpasses the Mean Accuracy, indicating excessive confidence in the model's predictions. Conversely, Under-confidence occurs when the Mean Accuracy exceeds the Mean Confidence, indicating a lack of confidence in the model's predictions. Calibrating uncertainty can bring the model's probabilistic outputs closer to the desired levels. This calibration process aims to minimize or eliminate any loss in accuracy values while achieving optimal confidence levels.

3) RELIABILITY SCORE

The distinction between the MC and MA values is defined in this paper by a metric referred to as the Reliability Score (RS). When the RS equals zero, the DNN achieves an optimal balance of accuracy and reliability. Over-confidence arises when the MC surpasses the MA, while Under-confidence occurs when the reverse condition holds. Previous research conducted by the authors in [7] demonstrates that DNNs with N classes and M inputs tend to exhibit overconfidence. Our study proposes that the implementation of simple preprocessing and post-processing algorithms has the potential to alter this behavior.

4) EXPECTED AND MAXIMUM CALIBRATION ERRORS

In the context of this paper, the error per confidence interval is determined by measuring the accuracy deviation from the center of the interval. The Expected Calibration Error quantifies the weighted error across all intervals [17], while the Maximum Calibration Error represents the maximum error observed among all intervals. In an ideal scenario, both errors would be zero.

5) NORMALIZED NEGATIVE LOG LIKELIHOOD LOSS (NLL)

The metric referred to as cross-entropy loss is employed as a loss function for DNNs [18], [19]. Furthermore, it serves as a metric for evaluating the efficacy of probabilistic models [20]. Initially, for each sample output from the DNN, the negative logarithm of the predicted probability of the ground truth class is computed. Then, these values are normalized per sample and summed together.

6) BRIER SCORE LOSS (BSL)

This metric is formulated as the mean squared error between the predicted probability, which ranges from zero to one, and the actual outcome, restricted to values of 0 or 1. The primary objective is to minimize this metric, aiming for a value that approaches zero as closely as possible [7]. While the metric inherently falls within the zero to one range, it is presented as a percentage to enhance the comparability of our research findings. To expand the applicability of the BSL and accommodate multiclass classification scenarios, we employ a computation method that compares the predicted output, represented as probabilities using one-hot encoding, with the corresponding ground truth output. The ground truth output is also encoded using a one-hot representation comprising zeros and ones. By calculating the squared difference between these two sets of values, we obtain the BSL for multiclass classification, averaged across all samples [21]. The computation of the BSL follows the formulation presented in Eq. (1), where M signifies the total number of samples. The variable N denotes the number of classes involved in the classification task. Furthermore, yp indicates the predicted probability for each class, while yg represents the ground truth encoded in one-hot form, with values of either zero or one for each class.

$$BSL = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N}(yp_{ij} - yg_{ij})^2. \tag{1}$$

TABLE 2. Performance evaluation of Methods 2 and 3 with varied output selection approaches in Validation and Test Pairs without considering Filter Range. Key performance parameters for various metrics are compared. M2 uses the average of augmented samples, M3-Max selects the output with maximum confidence, M3-Min selects the output with minimum confidence, and M3-N performs averaging while excluding the highest and lowest confidence values. The best test values in each column for DNN and ML are bold.

Val., Test	Accuracy (%)	AUC	ECE (%)	MCE (%)	BSL (%)	NLL	RS=MA-MC
DNN	94.58, 90.98	0.95, 0.91	**4.61, 3.48**	**7.8, 6.81**	7.95, 12.64	0.13, 0.2	**4.61, 1.21**
DNN (M2)	**94.85, 91.15**	**0.95, 0.92**	4.95, 3.77	10.3, 7.59	**7.58, 12.27**	**0.13, 0.19**	4.91, 1.4
DNN (M3-Min)	94.34, 90.84	0.94, 0.91	6.79, 4.04	18.25, 4.99	8.81, 12.4	0.15, 0.2	6.8, 3.76
DNN (M3-Max)	94.86, 91.14	**0.95, 0.92**	4.13, 3.75	8.93, 16.28	7.58, 13.37	0.13, 0.22	8.93, 16.28
DNN (M3-N)	94.84, 91.12	**0.95, 0.92**	4.81, 3.79	9.52, 7.93	7.57, 12.31	**0.13, 0.19**	4.79, 1.25
ML	87.2, 85.5	0.89, 0.87	5.17, 6.91	23.71, 27.37	21.06, 24.4	0.5, 0.6	-5.17, -6.91
ML (M2)	**88.48, 86.36**	0.89, 0.88	**1.61, 3.77**	**14.54, 18.89**	**17.43, 21.04**	0.35, 0.42	**-1.6, -3.77**
ML (M3-Min)	85.7, 84.08	0.87, 0.86	3.14, 4.26	17.07, 22.16	19.58, 22.35	**0.35, 0.41**	-2.61, -4.26
ML (M3-Max)	88.53, 86.28	**0.9, 0.88**	6.24, 8.52	32.25, 33.21	21.88, 26.38	0.69, 0.85	-6.24, -8.52
ML (M3-N)	88.03, 85.93	0.89, 0.88	2.99, 4.93	23.16, 27.87	18.68, 22.3	0.4, 0.5	-2.84, -4.93

Considering the utilization of the softmax function in our deep network for binary classification, our output predictions were binary but represented in the form of one-hot encoding. As a result, we have employed Eq. (1) as the method for computing the BSL.

IV. EXPERIMENTAL RESULTS

This section presents the outcomes of the two hybrid combinations and the scenario that uses fast ML as a primary classifier and DNN as the secondary classifier described in Section III-A. For each combination of algorithms and methods, results are produced for various filter choices in the test sets. Given that the filter variable is a hyperparameter, an appropriate filter is selected based on the validation results. These results are then utilized to assess the extent of divergence and disparity between the validation and test sets. Since relying on a single metric is insufficient for selecting the optimal filter choice, a comprehensive analysis is provided using all metrics outlined in Section III-B. Variations in the filters are determined by comparing these metrics and evaluating each result. The average deep neural network prediction from 10 runs is employed in this analysis.

A. FILTERING EFFECT

Despite combining different methods and algorithms, one of the most straightforward strategies involves altering the number of classes by introducing an additional category for rejected samples. This approach allows us to focus on classifying only highly reliable instances while categorizing the remaining samples into a separate neutral class. As a result, the percentage of rejected samples in both the validation and test sets remains nearly identical. Similarly, the achieved accuracy for accepted samples exhibits a similar behavior between the validation and test sets, albeit with slight variations in percentage. The percentage of rejected samples for the test set is illustrated in Figure 3a, and the accuracy for accepted samples for the test set is indicated in Figure 3b. Applying Methods 1 and 3-max on the DNN in this context yields similar results, while Methods 3-min and 2 produce more substantial changes. When comparing various methods on DNN using the narrowest filter range of 0.02 to 0.98, the utilization of Method 1 or 3-max results in a notable enhancement in accuracy, increasing from 90.98% to 99.39%. This improvement is achieved by discarding approximately 32.5% of the test set's data. Conversely, Method 2 attains an accuracy of 99.94%; however, it necessitates a higher proportion of data rejection in the test set, approximately 44.5%. Moreover, when the same filter range is applied to other methods, such as M3-N and M3-min, the rejection of 41% and 51% of the test set data leads to achieved accuracies of 99.9% and 99.97%. The suboptimal results obtained from the application of Method 1 on ML necessitate its removal from Figure 3. Consequently, the Method 1 approach is not recommended for the extra class strategy when ML is used as the primary classifier. Moreover, Method 2 significantly improves ML outputs, raising the accuracy from 86% to 95.67% when the same filter range of 0.02 to 0.98 is employed. In this case, the rejected sample rate is approximately 30%. We recommend the neutral class strategy tasks where AI's decision-making can be bypassed in hazardous situations, deferring to human experts or alternative algorithms. Consequently, the subsequent analysis will focus on applying another method or algorithm to the neutral class (rejected samples) to ensure the classification of all samples and examine its impact on reliability and accuracy, considering various calibration metrics.

B. ACCURACY ANALYSIS

Table 2 demonstrates that analyzing the combination of techniques and algorithms while considering their impact on uncertainty can lead to a slight increase in overall accuracy but negatively affects most calibration metrics (specifically in ML) or, to a lesser extent, in DNN. Given these observations, the necessity of conducting a comprehensive metrics analysis beyond accuracy becomes evident when comparing the effects of different methods. This approach allows for a more precise selection of combination methods and filter ranges to enhance most metrics or, at the very

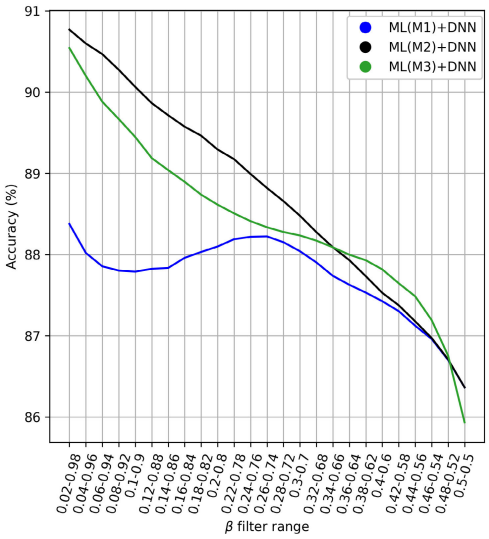
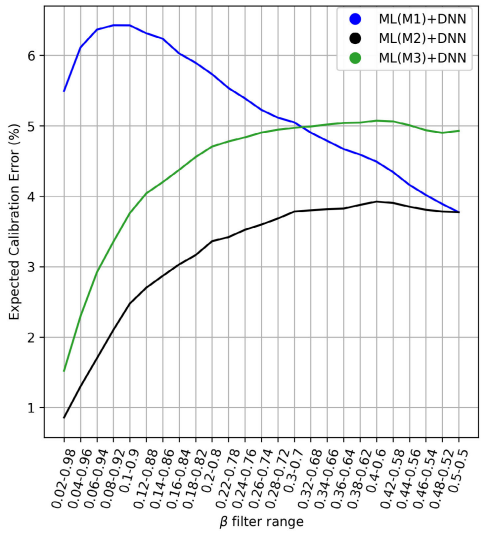
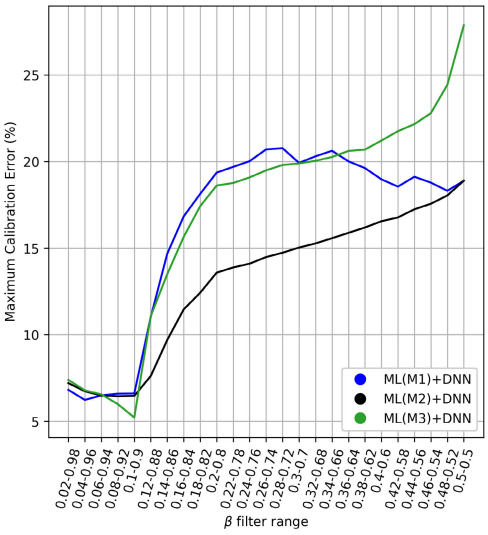


FIGURE 4. Accuracy per filters for the combination of algorithms and methods, using fast ML (XGB) as the primary classifier and DNN as the complementary algorithm.



(a) ECE per filters.



(b) MCE per filters.

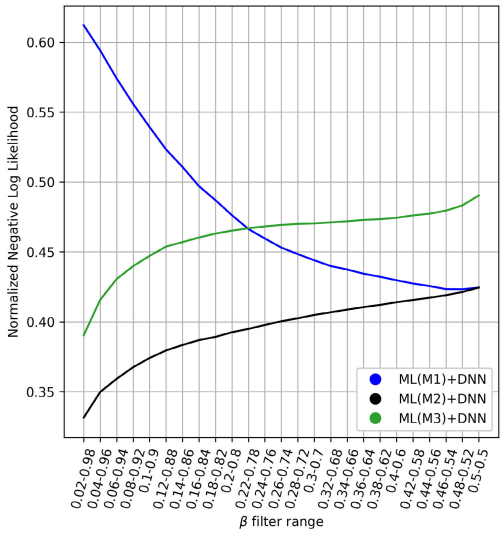
FIGURE 5. ECE and MCE per filters for the combination of algorithms and methods, using fast ML (XGB) as the main algorithm and DNN as the complementary algorithm.

least, improve accuracy while minimizing sacrifices in other metrics. In Figure 4, three combinations of methods and algorithms are presented for various filters, highlighting fast ML as the primary algorithm. Adjusting certain filters enables achieving test accuracy above 90% for two combinations. The impact of this combination on uncertainty analysis for a potentially favorable filter range will be further examined. However, according to Table 2, if Method 3-Max is applied, no improvement in uncertainty is expected, as Method 3-Max itself adversely affects uncertainty. Based on the findings from Table 2, the detrimental effect of M3-Max on uncertainty metrics prompts the exclusive use of the M3-N version for subsequent analysis to mitigate the decline in reliability. The selection of the M3-N version over M3-Max is driven by the need to minimize adverse effects on uncertainty metrics. This decision is informed by observed outcomes and the aim to uphold a higher level of reliability throughout the analysis. Using the M3-N version ensures that the analysis prioritizes reducing negative impacts on reliability while maintaining consistency in evaluating uncertainty metrics. When filters demand higher confidence, more samples are transferred to DNN, increasing accuracy. As anticipated from Figure 3, the combination of ML (Method 2) yields the most significant changes in accuracy. The combination of ML (Method 2) + DNN indicates a slightly superior effect on accuracy compared to employing ML (Method 3) + DNN.

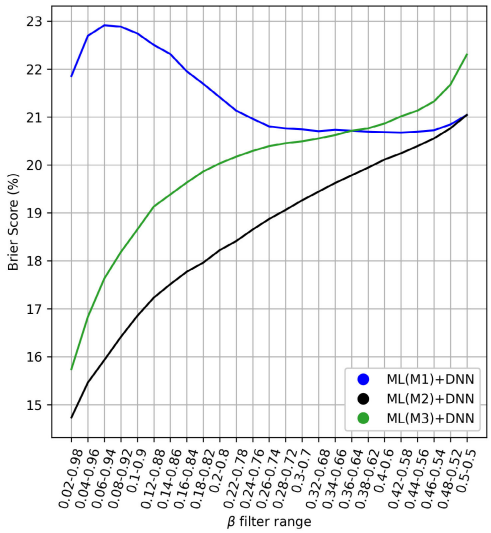
C. ECE AND MCE ANALYSIS

The analysis of ECE behavior, where ML assumes a primary role while employing DNN as a secondary algorithm, is depicted in Figure 5a. The most effective approaches identified within hybrid algorithms and methodologies are ML(M2)+DNN and ML(M3)+DNN. When comparing different hybrid algorithms that integrate ML(M2) or ML(M3) as the main component, it is evident that employing ML(M2) yields superior outcomes.

The MCE analysis offers valuable insights when considering ECE analysis, leading to more comprehensive conclusions. As depicted in Figure 5b, it is advisable to narrow the filter range from 0.2-0.8 to 0.02-0.98 to restrict the MCE values while simultaneously achieving improvements in this metric. The intersection of the new filter range is determined by considering ECE, MCE, and accuracy transitions. Based on Figure 5b, the analysis of MCE indicates that for all hybrid algorithms combining ML with DNN using various methods, it is feasible to obtain reliable MCE values by adjusting the range from 0.02-0.98 to 0.1-0.9. This adjustment is based on the validation results, which are also valid for the test set. This new tightened range handles



(a) NLL per filters.



(b) Brier Score per filters.

FIGURE 6. NLL and Brier Score per filters for the combination of algorithms and methods, using fast ML (XGB) as the main algorithm and DNN as the complementary algorithm.

around 70 to 80% of the data by ML. It transfers only around 20 to 30% of samples to DNN for classification as demonstrated in 5b, leading to a total accuracy of 89.5 to 91%.

D. NLL AND BSL ANALYSIS

The NLL and BSL are reliable indicators of model quality. Figures 6a and 6b illustrate their comparable behavior across different algorithms and methods applied per filter. Lower values of these metrics correspond to higher model quality. When XGB assumes the primary role and the filtered samples are fed into the DNN, an improvement in model quality is expected with an increase in the number of samples analyzed by the DNN. However, contrary to this expectation, the results indicate that the ML(M1) + DNN combination does not yield satisfactory performance based on the overall metrics. Therefore, Method 1 is not recommended for implementation when ML is the main algorithm. Different combinations utilizing Method 2 or 3 for XGB significantly demonstrate improved model quality. Specifically, when the filter range is closer to 0.02-0.98, the ML(M2)+DNN combination exhibits enhanced model quality.

E. FILTER RANGE SELECTION

Based on various metrics, when selecting an appropriate range for beta filters with ML playing the principal role, it is preferable to choose the left side of the filter range, spanning from 0.02-0.98 to 0.1-0.9. If a particular use case assigns greater importance to one of these metrics over the others, these recommendations may need to be adjusted accordingly.

V. CONCLUSION

In the context of 5G and 6G UAV communication systems, integrating fast Machine Learning (ML) and Deep Neural Networks (DNN) mechanisms is highly anticipated. Consequently, it is essential to comprehend the uncertainties associated with their utilization and assess their reliability when used individually or in combination. This paper explores the uncertainties of ML and DNN algorithms in these systems, considering different filtering ranges based on uncertainty metrics. By examining the reliability and total accuracy, we demonstrate that combining a DNN algorithm with ML can enhance overall accuracy. However, to ensure reliability is not compromised, limiting the contribution of the ML algorithm is crucial. As ML algorithms exhibit faster processing than DNN, employing a fast ML algorithm as the primary algorithm and DNN as an auxiliary can mitigate the negative impact of DNN prediction latency in 5G/6G networks while improving overall accuracy and reliability.

We study and analyze various probabilistic-based combinations to identify the most reliable and accurate combination, considering uncertainty metrics. We define a probability filtering range and introduce different methods for combining outputs within this reliable range. Furthermore, we demonstrate that augmentation techniques can enhance reliability in this combination approach. Our study proposes three main combined methods to concurrently increase accuracy and reliability in binary classification applied to UAV security scenarios. We aim to identify the optimal range and the most reliable combination based on analyzing reliability metrics. By implementing suitable preprocessing techniques, such as Time-Series Augmentation (TSA), for classification tasks, we demonstrate the generation of diverse versions of each sample, providing diversity for post-processing techniques.

Ultimately, while the proposed methods successfully improve accuracy, not all enhance reliability. Therefore, network engineers and developers must exercise caution when designing DNN architectures and thoroughly analyze them for accuracy and reliability. Our study focuses on the collaborative utilization of DNN and ML algorithms, such as when the ML algorithm serves as the main algorithm and the DNN as an auxiliary. The objective of this study does

not encompass simultaneous or parallel combinations of both DNN and ML algorithms or the usage of different DNN or ML algorithms for parallel predictions. These topics may be considered for future research, requiring the development of appropriate methods and algorithms to explore their effects.

REFERENCES

[1] X. Lin, "An overview of 5G advanced evolution in 3GPP release 18," *IEEE Commun. Standards Mag.*, vol. 6, no. 3, pp. 77–83, Sep. 2022, doi: 10.1109/MCOMSTD.0001.2200001.

[2] Z. Qin, H. Ye, G. Y. Li, and B. F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019, doi: 10.1109/MWC.2019.1800601.

[3] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018, doi: 10.1109/ACCESS.2018.2836950.

[4] T. Taleb, N. Sehad, Z. Nadir, and J. Song, "VR-based immersive service management in B5G mobile systems: A UAV command and control use case," *IEEE Internet Things J.*, vol. 10, no. 6, pp. 5349–5363, Mar. 2023, doi: 10.1109/JIOT.2022.3222282.

[5] Y. Li, J. Pawlak, J. Price, K. Al Shamaileh, Q. Niyaz, S. Paheding, and V. Devabhaktuni, "Jamming detection and classification in OFDM-based UAVs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16859–16870, 2022, doi: 10.1109/ACCESS.2022.3150020.

[6] A. Krayani, A. S. Alam, L. Marcenaro, A. Nallanathan, and C. Regazzoni, "Automatic jamming signal classification in cognitive UAV radios," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 12972–12988, Dec. 2022, doi: 10.1109/TVT.2022.3199038.

[7] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn.* (Proceedings of Machine Learning Research), vol. 70, D. Precup and Y. W. Teh, Eds. PMLR, Aug. 2017, pp. 1321–1330. [Online]. Available: http://proceedings.mlr.press/v70/guo17a/guo17a.pdf

[8] D. Hafner, D. Tran, T. Lillicrap, A. Irpan, and J. Davidson, "Noise contrastive priors for functional uncertainty," in *Proc. 35th Uncertainty Artif. Intell. Conf.*, vol. 115, R. P. Adams and V. Gogate, Eds., Jul. 2020, pp. 905–914. [Online]. Available: https://proceedings.mlr.press/v115/hafner20a.html

[9] J. Viana, H. Farkhari, P. Sebastião, L. M. Campos, K. Koutlia, B. Bojovic, S. Lagén, and R. Dinis, "Deep attention recognition for attack identification in 5G UAV scenarios: Novel architecture and end-to-end evaluation," *IEEE Trans. Veh. Technol.*, vol. 73, no. 1, pp. 131–146, Jan. 2024, doi: 10.1109/TVT.2023.3302814.

[10] J. Viana, H. Farkhari, L. M. Campos, P. Sebastião, K. Koutlia, S. Lagén, L. Bernardo, and R. Dinis, "A convolutional attention based deep learning solution for 5G UAV network attack recognition over fading channels and interference," in *Proc. IEEE 96th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2022, pp. 1–5, doi: 10.1109/VTC2022-Fall57202.2022.10012726.

[11] N. Patriciello, S. Lagen, B. Bojovic, and L. Giupponi, "An E2E simulator for 5G NR networks," *Simul. Model. Pract. Theory*, vol. 96, Nov. 2019, Art. no. 101933, doi: 10.1016/j.simpat.2019.101933. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X19300589

[12] I. Almomani, B. Al-Kasasbeh, and M. AL-Akhras, "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, pp. 1–16, Jan. 2016, doi: 10.1155/2016/4731953.

[13] O. Puñal, C. Pereira, A. Aguiar, and J. Gross. (May 2014). *CRAWDAD Dataset Uportorwthaachen/Vanetjamming2012 (v. 2014-05-12)*. Downloaded From. [Online]. Available: https://crawdad.org/uportorwthaachen/vanetjamming2012/20140512

[14] J. Viana, H. Farkhari, P. Sebastiao, S. Lagen, K. Koutlia, B. Bojovic, and R. Dinis, "A synthetic dataset for 5G UAV attacks based on observable network parameters," 2022, *arXiv:2211.09706*.

[15] M. H. DeGroot and S. E. Fienberg, "The comparison and evaluation of forecasters," *Statistician*, vol. 32, no. 1, p. 12, Mar. 1983. [Online]. Available: http://www.jstor.org/stable/2987588

[16] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*. New York, NY, USA: Association for Computing Machinery, 2005, pp. 625–632, doi: 10.1145/1102351.1102430.

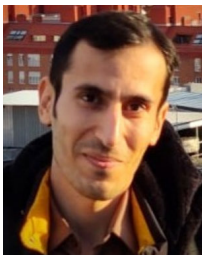
[17] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Red Hook, NY, USA: Curran Associates, 2019, pp. 1–11. [Online]. Available: https://proceedings.neurips.cc/paperfiles/paper/2019/file/8558cb408c1d76621371888657d2eb1d-Paper.pdf

[18] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Springer Series in Statistics). Cham, Switzerland: Springer, 2009, doi: 10.1007/978-0-387-84858-7.

[20] M. P. Naeini, F. G. Cooper, and M. Hauskrecht, "Obtaining well calibrated probabilities using Bayesian binning," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2901–2907, doi: 10.1609/aaai.v29i1.9602.

[21] G. W. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Rev.*, vol. 78, no. 1, pp. 1–3, Jan. 1950, doi: 10.1175/1520-0493(1950)078<0001:VOFEIT>2.0.CO;2.



HAMED FARKHARI (Member, IEEE) is currently pursuing the Ph.D. degree with the ISCTE–University Institute of Lisbon. He is an Early-Stage Researcher with the TeamUp5G Group, a European Training Network, under the Marie Skłodowska-Curie Actions (MSCA ITN) of the European Commission's Horizon 2020 Program. His research interests include cybersecurity, machine learning, deep learning, data science, meta-heuristic techniques, and optimization algorithms.

JOSEANNE VIANA (Member, IEEE) received the bachelor's degree in telecommunication engineering from the University of Campinas, Brazil. She is currently pursuing the Ph.D. degree with the Universidad Carlos III de Madrid (UC3M). She is an Early-Stage Researcher in the project TeamUp5G, a European Training Network under the MSCA ITN of the European Commission's Horizon 2020. Her research interests include wireless communications applied to interconnected systems, such as UAVs, aerial vehicles, and non-terrestrial devices.

SARANG KAHVAZADEH (Member, IEEE) received the B.S. degree in industrial engineering from Olom va Fonon Mazandaran, Iran, in 2010, the M.S. degree in software information technology from Debrecen University, Hungary, in 2014, and the Ph.D. degree from the Department of Computer Architecture, Universitat Politècnica de Catalunya Barcelona Tech–UPC, Barcelona, Spain, in 2019. He is currently a Researcher with the Services as NetworkS (SaS) Research Unit, Center Tecnologic de Telecommunications de Catalunya (CTTC), Barcelona. He has been involved in numerous European projects and co-authored many articles in international journals and conferences. His research interests include security, 5G NFVI, Kubernetes, OSM, cloud-native, microservices, testbed as a service, NFV, and distributed security mechanisms in the 5G environment.

PEDRO SEBASTIÃO (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from IST. He is currently a Professor with the Information Science and Technology Department, ISCTE-IUL. He is also the Board Director of the AUDAX-ISCTE's Entrepreneurship and Innovation Center, ISCTE. He is responsible for the LABS LISBOA Incubator. He is a Researcher with the Institute of Telecommunications. He has served as an Expert and an Evaluator for over 100 national and international civil and defense research and development projects. He has supervised several master's dissertations and Ph.D. theses. Furthermore, he has led several national and international research and development projects. He has received several awards in scientific, engineering, and pedagogical fields. He has organized or co-organized more than 55 national and international scientific conferences. In addition, he has planned and developed several postgraduate courses in technologies and management, entrepreneurship and innovation, and technology and innovation transfer. He has supported numerous projects involving technology transfer and the creation of startups and spinoffs that contribute value to society and the market. His professional experience includes work in the National Defense Industries, initially in the Office of Studies and later as the Board Director of the Quality Department for the Production of New Products and Technologies. He was also responsible for communication technology systems in the Nokia-Siemens business area. He is the author or the co-author of over 200 scientific articles. His research interests include monitoring, control, and communications of drones and unmanned vehicles, planning tools, stochastic processes (modeling and efficient simulations), the Internet of Things, and efficient communication systems.

VICTOR P. GIL JIMENEZ (Senior Member, IEEE) received the B.S. degree (Hons.) in telecommunications from the University of Alcala, in 1998, and the M.S. and Ph.D. degrees (Hons.) in telecommunications from the Universidad Carlos III de Madrid, in 2001 and 2005, respectively. In 1999, he was a Communications Staff Member with the Spanish Antarctica Base. He visited the University of Leeds, U.K., in 2003; the Chalmers Technical University, Sweden, in 2004; and the Instituto de Telecomunicações, Portugal, from 2008 to 2010. He is currently an Associate Professor with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid. He has led several private and national Spanish projects and participated in various European and international projects. He holds one patent and has published over 80 journal articles/conference papers and nine book chapters. His research interests include advanced multicarrier systems for wireless radio, satellite, and visible light communications. He received the Master's Thesis Award and the Ph.D. Thesis Award from the Professional Association of Telecommunication Engineers of Spain, in 1998 and 2006, respectively. He was the IEEE Spanish Communications and Signal Processing Joint Chapter Chair, from 2015 to 2023.

RUI DINIS (Senior Member, IEEE) received the Ph.D. degree from the Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal, in 2001, and the Habilitation degree in telecommunications from the Faculdade de Ciências e Tecnologia (FCT), Universidade Nova de Lisboa (UNL), in 2010. He was a Researcher with the Centro de Análise e Processamento de Sinal (CAPS), IST, from 1992 to 2005. From 2001 to 2008, he was a Professor with IST. In 2003, he was an Invited Professor with Carleton University, Ottawa, Canada. He conducted research with the Instituto de Sistemas e Robótica (ISR), from 2005 to 2008. Since 2009, he has been a Researcher with the Instituto de Telecomunicações (IT). He is currently an Associate Professor with FCT, Universidade Nova de Lisboa (UNL). He actively participates in several national and international research projects in the broadband wireless communications area. His research interests include transmission, estimation, and detection techniques. He serves as an Editor for IEEE Transactions on Wireless Communications, IEEE Transactions on Communications, IEEE Transactions on Vehicular Technology, IEEE Open Journal of Communications, and *Physical Communication* (Elsevier). He also served as the Guest Editor for *Physical Communication* (Elsevier) on the Special Issue on Broadband Single-Carrier Transmission Techniques. He is recognized as a VTS Distinguished Lecturer.

2.3. Article #3: Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation

This article presents an advanced framework for the identification of attacks in UAV communications by introducing DAtR. The primary objective was to develop a robust detection system capable of identifying attacks in realistic 5G UAV scenarios under LoS, NLoS, and probabilistic combinations of these conditions. The research focused on implementing a compact deep network embedded in authenticated UAVs, capable of processing observable parameters such as RSSI and SINR to recognize attacks under various challenging conditions.

The main contributions to this thesis are summarized as follows:

- Development of the novel DAtR architecture that integrates convolutional neural networks with self-attention mechanisms to improve detection accuracy;
- Introduction of two new pre-processing and post-processing techniques designed to improve accuracy while maintaining computational efficiency;
- Comprehensive experimental validation across multiple scenarios, demonstrating the framework's effectiveness in real-world conditions.

Extensive validation of the proposed framework was performed using calibration error metrics, confidence values, and detailed performance analyzes in various operational scenarios. A detailed comparison with six widely used classifiers highlighted the superior performance of the proposed approach, with notable improvements in detection accuracy under challenging NLoS conditions. Furthermore, the research included novel implementations to optimize resource usage and energy efficiency, critical considerations for UAV platforms.

Article Details

- **Title:** Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation
- **Date:** January 2024
- **Authors:** Joseanne Viana, Hamed Farkhari, Pedro Sebastião, Luis Miguel Campos, Katerina Koutlia, Biljana Bojovic, Sandra Lagén, Rui Dinis
- **Status:** Accepted in a major international journal with rigorous peer review
- **Journal:** IEEE Transactions on Vehicular Technology
- **DOI:** 10.1109/TVT.2023.3302814

The significance of this paper goes beyond technical innovations in attack detection to address practical implementation challenges in UAV systems. It establishes a comprehensive framework for evaluating and implementing security solutions in real-world UAV deployments, balancing detection accuracy with resource utilization. This research provides a foundation for future advancements in attention-based security systems for aerial platforms, emphasizing practical deployment considerations in 5G networks.

Deep Attention Recognition for Attack Identification in 5G UAV Scenarios: Novel Architecture and End-to-End Evaluation

Joseanne Viana, Hamed Farkhari, Pedro Sebastião, Luis Miguel Campos, Katerina Koutlia, Biljana Bojovic, Sandra Lagén, *Senior Member, IEEE*, and Rui Dinis, *Senior Member, IEEE*

***Abstract*—Despite the robust security features inherent in the 5G framework, attackers will still discover ways to disrupt 5G unmanned aerial vehicle (UAV) operations and decrease UAV control communication performance in Air-to-Ground (A2G) links. Operating under the assumption that the 5G UAV communications infrastructure will never be entirely secure, we propose Deep Attention Recognition (DAtR) as a solution to identify attacks based on a small deep network embedded in authenticated UAVs. Our proposed solution uses two observable parameters: the Signal to Interference plus Noise Ratio (SINR) and the Received Signal Strength Indicator (RSSI) to recognize attacks under Line-of-Sight (LoS), Non-Line-of-Sight (NLoS), and a probabilistic combination of the two conditions. Several attackers are located in random positions in the tested scenarios, while their power varies between simulations. Moreover, terrestrial users are included in the network to impose additional complexity on attack detection. Additionally to the application and deep network architecture, our work innovates by mixing both observable parameters inside DAtR and adding two new pre-processing and post-processing techniques embedded in the deep network results to improve accuracy. We compare several performance parameters in our proposed Deep Network. For example, the impact of Long Short-Term-Memory (LSTM) and Attention layers in terms of their overall accuracy, the window size effect, and test the accuracy when only partial data is available in the training process. Finally, we benchmark our deep network with six widely used classifiers regarding classification accuracy. The eXtreme Gradient Boosting (XGB) outperforms all other algorithms in the deep network, for instance, the three top scoring algorithms: Random Forest (RF), CatBoost (CAT), and XGB obtain mean accuracy of 83.24 %, 85.60 %, and 86.33% in LoS conditions, respectively. When compared to XGB, our algorithm improves accuracy by more than 4% in the LoS condition (90.80% with Method 2) and by around 3% in the short-distance NLoS condition (83.07% with Method 1).**

***Index Terms*—4G, 5G, convolutional neural networks, deep learning, jamming detection, jamming identification, security, UAV, unmanned aerial vehicles.**

I. Introduction

UNMANNED aerial vehicles (UAVs) have the potential to bring revolutionary changes that will fulfill consumer demands in several industry verticals [1]. UAVs will play a crucial role in emergency response [2], [3], package delivery in the logistics industry, temporal events [3] and remote areas [4], [5]. UAVs are becoming more common and reliable [6] due to technological advancements [7], [8], as well as the improvements in energy-efficient UAV trajectory optimization algorithms [9], [10], [11] that are able to be executed in practice to take into account the dynamics of the UAV as a parameterized method. Thus integrating UAVs into 5G and 6G networks will increase telecommunication coverage and reduce costs for businesses willing to invest in this technology. However, UAVs can easily be hacked by malicious users [12] throughout their wireless communication channels, which might divert delivery packets from their destinations. This can have disastrous consequences in unfortunate climate events where UAVs are transporting people to hospitals or in cases of criminal investigations. A jamming attack can lead to loss of UAV communication control, UAV robbery, UAV destruction, and property damage in urban areas, which would generate problems for business leaders. The authors in [13], [14], [15], [16] emphasize the need for research on new robust methods for attack detection and its associated challenges in 5G UAV communications. The ability to recognize different patterns in communication connectivity plays a vital role in the UAV security paradigm. Therefore, a Self-Identifying Solution against Attacks (SISA) becomes essential for UAV

Manuscript received 14 October 2022; revised 10 March 2023 and 10 June 2023; accepted 25 July 2023. Date of publication 7 August 2023; date of current version 17 January 2024. This work was supported in part by European Union's Horizon 2020 Research and Innovation Program through Marie Sklodowska-Curie Project under Grant 813391 and in part by the Fundação para a Ciência e a Tecnologia and Instituto de Telecomunicações under Grant UIDB/50008/2020. The work of Katerina Koutlia, Biljana Bojovic, and Sandra Lagén was supported in part by MCIN/AEI/10.13039/501100011033 and "ERDF A way of making Europe," TSI-063000-2021-56/57 6G-BLUR Project by the Spanish Government under Grant PID2021-126431OB-I00 and in part by Generalitat de Catalunya under Grant 2021 SGR 00770. The review of this article was coordinated by Dr. Wei Quan. *(Joseanne Viana and Hamed Farkhari contributed equally to this work.) (Corresponding authors: Joseanne Viana; Hamed Farkhari.)*

Joseanne Viana and Pedro Sebastião are with the ISCTE, Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal, and also with the Instituto de Telecomunicações, 1049-001 Lisboa, Portugal (e-mail: joseanne_cristina_viana@iscte-iul.pt; pedro.sebastiao@iscte-iul.pt).

Hamed Farkhari is with the ISCTE, Instituto Universitário de Lisboa, 1649-026 Lisbon, Portugal, and also with the PDMFC, 1300-609 Lisboa, Portugal (e-mail: hamed_farkhari@iscte-iul.pt).

Luis Miguel Campos is with the PDMFC, 1300-609 Lisboa, Portugal (e-mail: luis.campos@pdmfc.com).

Katerina Koutlia, Biljana Bojovic, and Sandra Lagén are with the Centre Tecnològic de Telecomunicacions de Catalunya, 08860 Barcelona, Spain (e-mail: katerina.koutlia@cttc.es; bbojovic@cttc.es; slagen@cttc.es).

Rui Dinis is with the Instituto de Telecomunicações, 1049-001 Lisboa, Portugal, and also with the FCT, Universidade Nova de Lisboa, 1099-085 Lisbon, Portugal (e-mail: rdinis@fct.unl.pt).

Digital Object Identifier 10.1109/TVT.2023.3302814

communication control. Furthermore, According to [17], identifying interference must be the basis for selecting anti-jamming solutions. Statistical models have recently been recognized as a viable way to monitor network activity in wireless communications and detect suspicious attacks through wireless parameters. Using Bayesian estimators, Cheng et al. [18] employ a sequential change point detection algorithm to detect the state changes in the time series. The authors of [19] present a jamming detection approach based on a Naive Bayes classifier trained on a small sample of data and addresses just noise effects. Lu et al. [20] propose the message invalidation ratio as a new metric for evaluating performance under jamming attacks in time-critical applications. In [21], the authors offer a jamming detection strategy for Global Navigation Satellite System (GNSS) based trained localization that makes use of Singular Value Decomposition (SVD). However, most research needs to account for the effects of the wireless propagation channel in their solutions.

Concerning machine learning, Krayani et al. use a Bayesian network to identify jammers [22]. Youness et al. [23] create a dataset based on signal property observations and use Random Forest (RF), Support Vector Machines (SVM), and a neural network algorithm to classify the features extracted by the jamming signal. [24] also uses an SVM and a Self-Taught Learning method to identify attacks in UAV Networks. In [25], the authors utilize a Machine Learning Intrusion Detection System (ML-IDS) based on SVM to identify jamming in the Cloud Radio Access Network (C-RAN). Deep Learning (DL) has been used to create models with high-level data abstraction by utilizing numerous layers with activation function processing.

In DL, Deep Neural Networks (DNNs), such as Convolutional Neural Networks (CNNs), can define trends and seasonality in time series data [26], [27], [28]. These characteristics make deep network-based algorithms helpful in discovering patterns in wireless networks by analyzing time series and spatial information [29]. The authors in [30] identify jamming samples using signal-extracted features and 2D samples and pre-trained networks, such as AlexNet, VGG-16, and ResNet-50. In [31], the authors also use pre-trained deep networks to develop a three-step framework to identify jamming in radar scenarios. In [32], the signal features in the time domain, frequency domain, fractal dimensions, and deep networks are used to recognize jamming attacks. Nevertheless, DL presents its own challenges when applied in the wireless context:

1) It is challenging to collect network parameters for DL input layers. All deep learning algorithms need training and testing. In each phase, the DNN's input layer comprises the parameters of the data samples. The greater the sample coverage in terms of data qualities, the better the DL can identify network features. However, some wireless data may be missing due to the stochastic nature of the communication paths. Consequently, DL models should be built to tolerate missing parameters, data errors, and out-of-range values in their input layers;
2) UAVs have constraints in memory, CPU capabilities, and available batteries. In addition, complex algorithms cannot be programmed into their current protocols because DL is iterative in nature. This may prolong system response time. The DL algorithms should use techniques to save memory space without increasing the number of layers, nodes, or trainable parameters. Also, the algorithms should be optimized to minimize execution time;
3) DL needs entire or nearly complete training samples to effectively detect network patterns. However, because of the difficulty of collecting so many data points for each potential network condition, the training samples may be relatively restricted. This dictates that DL should be capable of adding additional samples after failing to recognize a new pattern. The fresh samples may help to increase the accuracy of the DL models;
4) Furthermore, network engineers/programmers are required to carefully design the DL data formats since various network parameters have extremely distinct data properties and formatting requirements. The correct numerical representations and data normalization algorithms must be explicitly stated to combine numerous network parameters into the same DL input layer;

A. *Objectives and Contributions*

In this article, we study the attack identification problem in authenticated UAVs in 5G communications. To enable UAVs to cope with jamming recognition, we propose a deep network called DAtR (Deep Attention Recognition) that uses only two observable parameters: Signal to Interference plus Noise Ratio (SINR) and Received Signal Strength Indicator (RSSI). We demonstrate that utilizing these two parameters as inputs to our deep neural network (DNN) enables precise and reliable identification of jamming attacks because channel variations impact both values, and their values include information regarding the wireless channel state. The SINR represents the ratio of the desired signal power to the combined interference and noise power. In the presence of channel variations, such as fading, multipath propagation, and interference, the SINR can fluctuate, leading to changes in the quality and reliability of the received signal. The RSSI quantifies the power level of the total received signal, considering the useful signal plus interference and noise components. Channel variations can cause fluctuations in the RSSI value, as the received signal power may vary due to factors like distance, obstacles, fading, and interference.

5G communication networks provide these measurements in the receivers in Line-of-Sight (LoS), Non-Line-of-Sight (NLoS), and probabilistic LoS and NLoS conditions in the deep network and compare the accuracy for each channel condition case. We use a neural network that includes Attention layers with optimized parameters to decrease the chances of low accuracy when adding users and attackers to the network. We demonstrate that the DAtR can recognize jamming attacks from other malicious aerial agents in complex urban environments where terrestrial users are connected to the network. The final goal is to demonstrate that it is possible to identify attacks in the UAV's receiver that deal with the temporal dynamic behavior of the 5G network using learning techniques, such as deep network architectures, which have significantly fewer layers than well-known pre-trained networks. Also, the deep network does not rely on

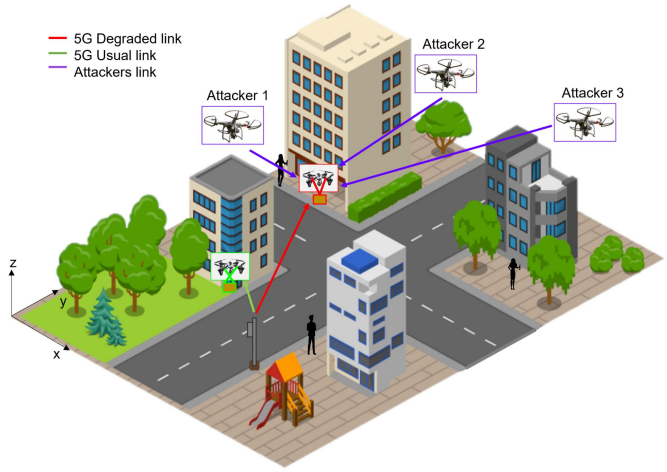


Fig. 1. Simulation scenario.

transfer learning techniques, and it could provide better accuracy than other well-known classifiers.

Taking these into account, the main contributions of this work are highlighted in the following:

1) A novel, robust, and effective Convolutional Attention deep network for UAVs, named DAtR, detects jamming in complex environments under LoS and NLoS conditions and tolerates incomplete raw data inputs. To the best of the authors' knowledge, this is the first time an Attention model has been proposed to detect jamming in LoS, NLoS, and hybrid conditions;
2) Two new complementary methods are named Time Series Augmentation (TSA) and Majority Voting Algorithm (MVA) to improve classification accuracy and detect false alarms for deep networks.;
3) A study of deep network architectures for UAVs considering Long Short-Term-Memory (LSTM) and Attention layers for 5G UAV communication data;
4) An accuracy comparison with six other state-of-the-art machine learning classifiers;
5) An analysis of the trade-offs between accuracy and added latency in the model while identifying attacks;

The remaining parts of this article are organized as follows. Section II presents the preliminaries and the attack identification problem in authenticated UAVs. Additionally, it describes the transmission and channel models, as well as the observable parameters of SINR and RSSI and the attacks dataset we developed. Section III illustrates the proposed deep network architecture for jamming identification where we discuss the layer's selection and implementation in detail. Section IV describes the novel proposed pre-processing and post-processing techniques that we embed in the deep network to improve accuracy results. Section V presents the accuracy analysis of the network simulation results, comparisons of parameter configurations, comparisons between the proposed deep network with six different classifiers, and the average processing time for each classifier. Finally, Section VI includes our conclusions. Table I summarizes the abbreviations used in this article.

II. Preliminaries and Problem Formulation

A. Scenarios

Fig. 1 illustrates the UAV simulation environment. In addition, it identifies the adopted X-Y-Z Cartesian coordinates. We consider a scenario where authenticated UAVs fly in a 1 km $\times$ 1 km square area while they are connected to a serving small cell through Air-to-Ground (A2G) 5G wireless data links. In this environment, we include authenticated terrestrial users placed on the ground. UAV attackers are placed in predetermined, randomly assigned spots. They fly towards the authenticated UAVs inside the coverage area of the small cell. To create our model, we assume that the authenticated UAV transmission power is fixed during each simulation, and we use Clustered Delay Line (CDL) channels, including slow and fast fading components, to model their propagation conditions. UAV attackers use the same propagation models as authenticated UAVs [33], [34]. For the terrestrial users, we follow the 5G wireless terrestrial propagation models defined in [34] instead. Fig. 1 shows a configuration example with two authenticated UAVs, three terrestrial users, three UAV attackers, and one small cell.

For the sake of simplicity, the authors considered the UAV to be a "flying antenna"; assuming that the UAV's mechanical components are not considered for this experiment and the antenna location in the UAV is ideal.

When UAV attackers move, their speed is kept constant, and they head toward the authenticated UAVs getting closer to them as simulation time evolves. The attackers' and authenticated UAVs' positions are at higher altitudes and follow the losses according to the standards in [33] and [34]. Our research presumes that terrestrial users may likewise be in fixed locations or can change their positions according to mobility models [35]. The small cells are configured with an antenna height of 10 m, typically seen in urban environments.

Table II displays the four different experimental setups we created, in which basically multiple combinations of mobility for UAV attackers and/or terrestrial users are considered. During the simulations, as further explained in Section V, we vary the scenarios to account for different mobility/speed options, as well as different distances between the small cells and authenticated UAVs, UAV attacker power, number of UAV attackers, and number of terrestrial users.

The authenticated UAVs try to identify if there are any attackers attempting to disrupt the communication link by using the proposed DAtR mechanism, which is fed with the RSSI and SINR measurements that are available in the receiver. For each scenario listed in Table II, we create a dataset with 600 files, including up to four attackers and thirty terrestrial users connected at the same time. We group them together to form a complete dataset composed of 2400 files split into RSSI and SINR parameters in constant LoS condition. Then, we change the channel condition in the dataset and check if it is possible to identify the attackers in persistent NLoS condition, and in randomly combined LoS and NLoS conditions through the 3rd Generation Partnership Project (3GPP) stochastic models in [33] and [34]. In the end, we have three datasets with 2400 files each, corresponding to LoS, NLoS, and hybrid LoS/NLoS

TABLE I
ABBREVIATION LIST

Abbreviation	Definition	Abbreviation	Definition
ASA	Azimuth Spread of Arrival	LSTM	Long Short-Term Memory
ASD	Azimuth Spread of Departure	MVA	Majority Voting Algorithm
A2G	Air to Ground	NLoS	Non-Line-of-Sight
CAT	CatBoost	OFDM	Orthogonal Frequency Division Multiplexing
CDL	Clustered Delay Line	RF	Random Forest
CNN/Conv1D	Convolutional Neural Network	RSSI	Received Signal Strength Indicator
CPU	Central Processing Unit	SINR	Signal to Interference plus Noise Ratio
C-RAN	Cloud Radio Access Network	SISA	Self-Identifying Solution against Attacks
DAtR	Deep Attention Recognition	SVD	Singular Value Decomposition
DL	Deep Learning	SVM	Support Vector Machines
DNN	Deep Neural Network	TSA	Time Series Augmentation
GNB	Gaussian Naive Bayes	UAV	Unmanned Aerial Vehicle
GNSS	Global Navigation Satellite System	UMi	Urban Micro Scenario
MH-DNN	Multi-Headed Deep Neural Network	URD	Uniformly Random Distributed
ML-IDS	Machine Learning Intrusion Detection System	XGB	eXtreme Gradient Boosting
LoS	Line-of-Sight	ZSA	Zenith Spread of Arrival
LR	Logistic Regression	ZSD	Zenith Spread of Departure

TABLE II
SPEED CONFIGURATION SCENARIO

Scenario	Attackers configured with speed	Users configured with speed
None Speed	N	N
Attackers Speed	Y	N
Users Speed	N	Y
Both Speed	Y	Y

conditions. Additional information on the dataset's development and possible applications are available in [36], [37]. The study of the attacks in urban environments is an intriguing problem due to the fact that in LoS cases, channel variations and terrestrial users increase the difficulty of self-identifying attacks. The deep network must distinguish grounded users from intruders considering the channel variations due to speed and environment changes over time. Under the NLoS condition, the lower received power makes it more challenging to recognize the UAV attackers. Finally, let us notice that the connection link between the authenticated UAV and the small cell exists during the entire simulation, even in low SINR circumstances.

B. Communication Model

We consider an A2G connection between the small cell and the authenticated UAVs, as depicted in Fig. 1. The scenario consists of an urban environment where buildings, trees, and other structures may cause significant path loss and shadowing degradation. We define the A2G large-scale effect with two components, i.e., path loss and shadowing, as follows:

$$L^{\alpha}(d,f) = PL^{\alpha}(d,f) + \eta^{\alpha}\ [\text{dB}], \tag{1}$$

where $PL^{\alpha}(d,f)$ is the path loss at distance d from the authenticated UAV to the respective small cell (in km) when transmitting over the carrier frequency f (in MHz), η^{α} is the shadowing (in dB), and α reflects the LoS and NLoS conditions, i.e., $\alpha \in \{\text{LoS}, \text{NLoS}\}$.

In A2G communications, the path loss $PL^{\alpha}(d,f)$ in (1) depends on the high/low altitude configurations and the LoS/NLoS conditions. We compute it as follows:

$$PL^{\alpha}(d,f) = \begin{cases} PL^{\text{LoS}}(d,f) & if\ \text{LoS} \\ PL^{\text{NLoS}}(d,f) & if\ \text{NLoS}. \end{cases} \tag{2}$$

For urban UAV scenarios, the path loss in the LoS condition is given by the maximum between high/low altitude path loss computations:

$$\begin{aligned} PL^{\text{LoS}}(d,f) &= \max(PL_h(d,f), PL_l(d,f)), \\ PL_h(d,f) &= 20\log(d)+20\log(f)+20log(4\pi/c), \\ PL_l(d,f) &= 30.9+(22.25-0.5\log(h))\log(d)+20\log(f), \end{aligned} \tag{3}$$

where c is the speed of light (in m/s), h is the altitude (in m), $PL_h(d,f)$ is the free space path loss for high altitudes, and $PL_l(d,f)$ is the low altitude path loss.

Under NLoS condition, the path loss is given by the maximum between the LoS path loss and the NLoS path loss expression:

$$\begin{aligned} PL^{\text{NLoS}}(d,f) &= \max\left(PL^{\text{LoS}}(d,f), PL_n(d,f)\right), \\ PL_n(d,f)_{\alpha} &= 32.4+(43.2-7.6\log(h))\log(d)+20\log(f). \end{aligned} \tag{4}$$

In our scenario, we assume that all the UAVs fly with a height within the margin of 22.5 m $< h <$ 300 m. With that in mind, the remaining shadowing component (η^{α}) in (1) is defined by 3GPP as an additional variation over the path loss with a certain standard deviation, depending on LoS/NLoS conditions as well. Table III includes the shadowing characterization for LoS and NLoS.

TABLE III
SHADOWING FOR UAVS IN UMI [33], [34]

	Std. deviation (dB)	Altitude (m)
LoS	$\max(5 \times \exp(-0.01h), 2)$	$22.5 < h < 300$
NLoS	8	$22.5 < h < 300$

To determine the LoS or NLoS condition for each communication link, 3GPP uses a stochastic model. The probability of being in LoS (p_{LoS}) is given by:

$$p_{\text{LoS}} = \frac{d_1}{d_{2D}} + \exp\left(\frac{-d_{2D}}{p}\right)\left(1 - \frac{d_1}{d_{2D}}\right), \tag{5}$$

where $p = -233.98 \log_{10}(h) - 0.95$, h is the height of the UAV, $d_1 = \max(294.05\ log_{20}(h) - 432.94, 18)$, and d_{2D} is the 2D distance between the UAV and the small cell. Accordingly, the probability of being in NLoS is $p_{\text{NLoS}} = 1 - p_{\text{LoS}}$. For small-scale fading, we adopt CDL models, as in [34] and [33]. 3GPP defines in tabular mode the parameters that model the fading, including the powers, delays, Angle of Arrival (AoA), and Angle of Departure (AoD) that contain spreads in Azimuth Spread of Arrival (ASA), Azimuth Spread of Departure (ASD), Zenith Spread of Arrival (ZSA), and Zenith Spread of Departure (ZSD) of each cluster for the UAV scenario. The scenario assumes large and small-scale fading in the link between the UAVs and the small cells. Given this model, the received power at the UAV with no jammers or interferences can be expressed as:

$$P_{uav} = P + G - L^{\alpha}(d, f) - S(n, m), \tag{6}$$

where P is the transmission power, G is the overall antenna gain in the link considering UAV and small cell antenna gains, i.e., $G = (G_{uav} + G_{sc})$, and $S(n, m)$ is the small-scale fading effect, which corresponds to the superposition of n clusters with m rays in the communication link, as per [33], [34]. Our model considers single antenna elements in the small cell and the UAVs. The simulation in this work uses CDL-A and CDL-D models for small-scale fading in the NLoS and LoS conditions. In this case, each CDL comprises 23 clusters with 20 multi-path components (rays) each. Each cluster has an AoA and an AoD. These values are used to create the rays' AoAs/AoDs according to the azimuth/zenith arrival/departure spreads (ASA/ASD, ZSA/ZSD), respectively.

The SINR, Γ_{uav}, between the authenticated UAV and the small cell at distance d, in the presence of interference coming from jammers and terrestrial users, is given by:

$$\Gamma_{uav} = \frac{P_{uav}}{\zeta^2 + \sum_{i=1}^{U} P^i_{\text{user}} + \sum_{j=1}^{J} P^j_{\text{jammer}}}, \tag{7}$$

where P^i_{user} and P^j_{jammer} represent the received power at the UAV coming from the i-th user and the j-th jammer, respectively, which act as interfering signals (including the channel gain with the authenticated UAV, ζ^2 is the noise power, U is the total number of terrestrial users transmitting at the same time as the authenticated UAV, and J is the number of jammers transmitting in the scenario. $\wedge$ is the RSSI which includes the linear average of the total received power in Watt from all sources, including co-channel serving and non-serving cells, adjacent channel interference, thermal noise, etc. Considering $\wedge_0$ as the RSSI value at a reference distance, we have

$$\wedge = \wedge_0 - 10\rho \log(d), \tag{8}$$

where $\rho = L^{\alpha}(d, f) + S(n, m)$ includes path loss and fast fading components, and d is the link distance.

We considered the inclusion of additional parameters, such as the Reference-Signal-Receive-Power (RSRP). However, our experimental analysis revealed that RSRP parameter did not make a significant contribution to the overall results. This outcome was expected, as RSRP and SINR are closely related to each other.

C. Problem Formulation and Dataset

The SISA goal for the authenticated UAV is to quickly identify malicious changes in the received power caused by UAV jammers in the environment. For that, we use a small deep network, where the number of trainable parameters T is smaller than 100 k ($T < 10^5$), that is composed of a combination of layers, including CNNs, Attention, Dropout, and Batch Normalization, among others. The details of the DNN architecture are provided in Section III.

First, we study the case where UAV attackers try to disrupt communication when the UAV and the small cell can directly see each other (LoS condition). Then, we simulate the NLoS condition, where buildings and other elements in the city may block the direct communication between the UAV and the small cell. Finally, we study a probabilistic combination of LoS and NLoS conditions. As such, we assume the following in the three datasets we create for the experiment:

- *LoS:* The UAV is always in LoS condition throughout all the simulations available in the dataset;
- *NLoS:* The UAV is in NLoS condition for the entire time during all the simulations included in the dataset;
- *LoS and NLoS:* The link between the UAV and the small cell is in either LoS or NLoS condition with a probability of p_{LoS} and $p_{\text{NLoS}} = 1 - p_{\text{LoS}}$ (according to (5)) for all the simulations in the dataset.

Table II describes the four scenarios in each dataset. The differences between the scenarios inside the dataset relate to the following parameters: the UAVs' and terrestrial users' mobility and speed, the distance between the small cell and the authenticated UAVs, the number of attackers and their power, and the number of terrestrial users in the network. It is important to note that the scenarios in the dataset, such as `Attackers' Speed`, `Users' Speed`, `Both speed`, and `None Speed` are unbalanced, meaning that the proportion between attackers and no attackers in the raw data is different. For example, the dataset has data for 1, 2, 3, and 4 attackers, while for no attacks, there is 0 attacker data. Therefore, to avoid bias toward the classification, it is necessary to implement countermeasures to balance the data during the pre-processing phase. Our deep network design aims to achieve maximum performance. To this

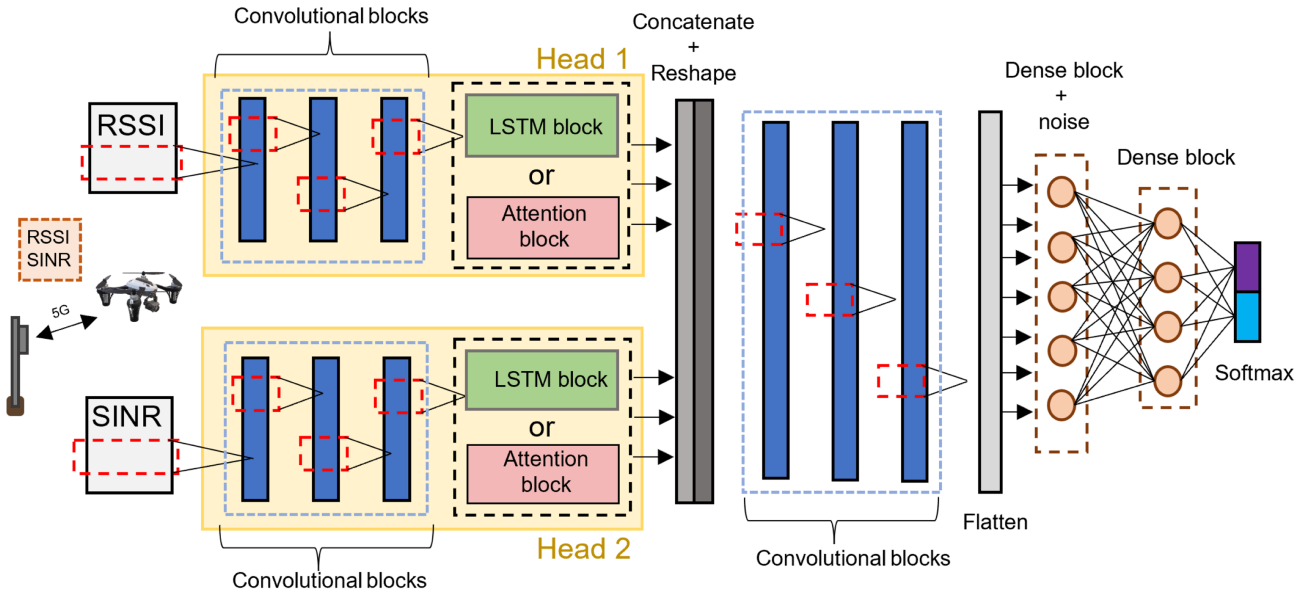


Fig. 2. Multi-Headed Deep Neural Network (MH-DNN) architecture. Note the switch from LSTM to Attention layers.

end, we compare the use of LSTM and Attention layers. We improve the capabilities of the Multi-Headed Deep Neural Network (MH-DNN) by integrating TSA and MVA techniques, which results in the proposed DAtR. We benchmark our DAtR with six other well-known ML algorithms and analyze other parameters, such as the optimum window size, the attack accuracy when the deep network sees the data for the first time during the test, and the latency added due to the DAtR processing time.

III. Convolutional Attention-Based Attack Detection

The proposed SISA model is based on an MH-DNN. The proposed architecture is shown in Fig. 2. It contains (i) three CNN blocks and (ii) an Attention or an LSTM block in each head. The body of the deep network consists of: (i) a Concatenate and Reshape layer, (ii) three CNN blocks, (iii) two Fully connected blocks, and (iv) the output layer (Softmax) for two classification classes. Although RSSI and SINR measure different parameters from the telecommunication perspective, both values may be related. For example, when RSSI increases, SINR may decrease; The multi-headed structure of the MH-DNN allows the extraction of the essential characteristics of the RSSI and SINR separately before combining both signals in the MH-DNN body. Also, it enables scalability when considering other telecommunication parameters such as Reference Signal Received Power (RSRP) by adding another head with the same structure and using transfer learning of the existing RSSI and SINR heads. This method can save the training process in the future for a new M-headed DNN while utilizing the advantage of the current pre-trained DNN.

Using our proposed MH-DNN, we can simultaneously extract features from both parameters in each head at each window size. The window size defines the length of each sequence that the deep network will receive as an input in each head. Fig. 3 presents the components of each block illustrated in Fig. 2. Each

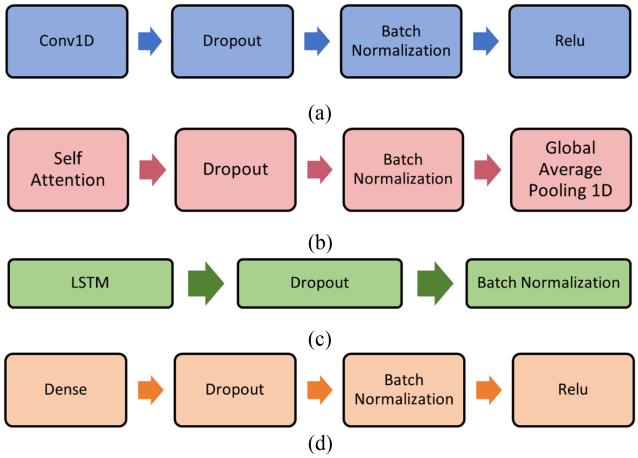


Fig. 3. Detailed block components in the proposed MH-DNN model. (a) Convolutional block, (b) Attention block, (c) LSTM block, (d) Dense block.

Convolutional block sequentially aggregates Conv1D, dropout, batch normalization, and the Relu layers. The Attention block contains Self-Attention, dropout, and batch normalization layers followed by the global average pooling 1D layer. The LSTM block includes the LSTM, the dropout, and the batch Normalization layers, and the Dense block encloses the same structure as the Convolutional block, except that the Conv1D layer is replaced by the Dense layer. Each block component performs an essential function to facilitate the MH-DNN head and body integration. The supplementary layers also keep the output sizes consistent and reduce the over-fitting chances. For example, adding dropout immediately after the main layers (i.e., Conv1D, Self-Attention, LSTM, and Dense) is one of the techniques that we used to avoid the MH-DNN over-fitting. The dropout configuration value D is the same for all blocks ($D = 0.4$). It defines the probability of each output node to be enabled

TABLE IV
COMPARISON BETWEEN TRAINABLE AND TOTAL PARAMETERS IN MH-DNN WITH ATTENTION OR LSTM BLOCKS

	Trainable parameters	Total parameters
Attention (4, 4)	57,936	58,540
Attention (8, 8)	**64,368**	**64,972**
Attention (16, 16)	90,096	90,700
LSTM (16)	**59,984**	**60,588**

temporally and randomly during the training process. In other words, it prevents the deep network from memorizing the input parameters instead of learning the patterns in the sequences. The Batch normalization layer speeds up convergence by normalizing data for the next input layer. Note that batch normalization is applied after the dropout layer to prevent information leakage from one layer to another.

In the Convolutional block, we convolute both signals in each head in the three CNN layers, as Fig. 2 indicates. Each layer creates a Convolution kernel that is convoluted with the layer input over a single temporal dimension to produce a tensor of outputs. Thanks to the configuration of strides and kernels, this operation returns a single tensor with several channels (i.e., $1 \times$ width $\times$ $channel$). The Convolution operation extracts different features from the time series sequences available in each head. The result from the Convolutional blocks is computed in parallel in the Attention layer.

The Attention layer utilizes an auxiliary vector to selectively weight the input features by computing a set of Attention weights based on the information from the previous and current states. These weights are then used to adjust the importance of different input parts when making predictions [38]. For example, the Attention mechanism may look for parts of the signal which might contain attack characteristics. The input tensor in the Attention block has the shape of $batch\ size$ by $width$ by $filters$ (i.e., $32 \times 50 \times 16$), and the global average pooling layer reduces dimensionality to $batch\ size$ by $filters$ (i.e., 32×16). The $width$ dimension is related to the window size of input sequences of each head of MH-DNN. In a similar architecture, we use the LSTM block with 16 units for the LSTM layer instead of the Attention block to compare these two different blocks in performance. A fair comparison between LSTM and the Attention layer's overall performance requires that both blocks' output create almost the same tensor size. The LSTM layer with 16 units creates the same output tensor shape as the Attention block (i.e., 32×16). Another metric to compare is the number of MH-DNN parameters generated with Attention or LSTM blocks. For example, Table IV compares trainable and total parameters of the MH-DNN configured with LSTM blocks (16 units) and Attention configured with different heads and keys. According to Table IV, the first two Attention settings (4×4) and (8×8) produce a number of parameters close to the MH-DNN embedded with the LSTM (16). However, there is a high leap when the MH-DNN uses the Attention blocks with (16×16) heads and keys. Therefore, based on the knee (elbow) rule, we choose the Attention configuration (8×8), the highest configuration before the leap in the amount of trainable parameters related to the attention layer. Notice that, even though Attention produces more trainable parameters than LSTM, the benefits in accuracy in NLoS scenarios compensate for this difference.

TABLE V
MH-DNN CONFIGURATION PARAMETERS

Deep network Parameters	Values
Number of input heads	2
Base learning rate	2.5×10^{-2}
Base batch size	32
Optimizer	Adam
——————- Heads ———————	
Conv1D (filters, kernel size, stride)	8, 6, 2
Conv1D (filters, kernel size, stride)	16, 6, 1
Conv1D (filters, kernel size, stride)	16, 5, 2
Self-Attention (heads, keys)	8, 8
(or LSTM)	(16)
——————- Body ———————	
Conv1D (filters, kernel size, stride)	8, 3, 1
Conv1D (filters, kernel size, stride)	16, 2, 1
Conv1D (filters, kernel size, stride)	16, 2, 1
Fully connected (Dense)	100
Gaussian noise	0.3
Fully connected (Dense)	50
Softmax	2
——————- blocks ———————	
Dropout layers	0.4
L2 regularization for Conv1D, and LSTM layers	1×10^{-6}
L2 regularization for Dense and Attention layers	1×10^{-5}

The concatenation procedure merges the features extracted from RSSI and SINR in each head, and the reshape method prepares them for the following CNN blocks. After using the CNN blocks in the body, we apply two Dense blocks. The first one is followed by an additive Gaussian noise N ($N = 0.3$). Additive noise injection during the training process increases our model's stability and robustness. Moreover, it performs as a regularizer to prevent over-fitting and improve generalization [39]. We ended our MH-DNN with a Softmax layer with two nodes for binary classification and the categorical cross entropy as a loss function. Table V shows the main parameters for the MH-DNN. Notice that we did not employ padding for any of the Conv1D layers, since it decreases the output $width$ after each Convolutional block. We apply L2 regularization only in the Convolutional, Attention, LSTM, and Dense layers weights with no bias decay. Also, we use the batch normalization layers with no regularization, as recommended by [40].

IV. IMPROVEMENTS IN MH-DNN ROBUSTNESS

In this section, we introduce the TSA method combined with the MVA to improve the performance of our deep neural network under the NLoS condition, which tends to present lower total received power compared to the LoS condition. Fig. 4 summarizes the significant additions to the MH-DNN to include

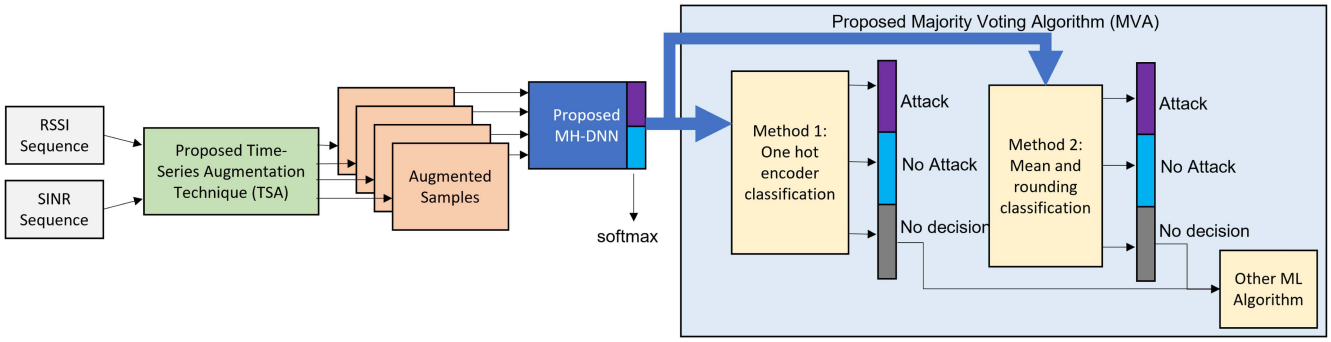


Fig. 4. Deep Attention Recognition (DAtR), including TSA and MVA Techniques - methods 1 and 2.

TABLE VI
OUTPUT OF THE TSA

	RSSI Sequence	SINR Sequence
Sample 1	Same	Same
Sample 2	Same	Flipped
Sample 3	Flipped	Same
Sample 4	Flipped	Flipped

these two new methods. After incorporating both techniques into the system, we named the new system DAtR.

A. Time Series Augmentation Technique

TSA aims to supplement the original dataset with additional augmented samples for the MH-DNN to process further. We create the additional data using data augmentation and flipping techniques applied in the training set to increase data diversity and prevent over-fitting during the training process. Also, we use this technique in both training and test sets combined majority voting method, which converts binary classification into three classes in Section IV-B. As Fig. 4 shows, we transform the input samples into four augmented samples. In Table VI, we display an example of generating the four new expanded instances according to TSA.

By randomly inverting each RSSI and SINR sequence, we can generate four different augmented samples from each occurrence. Other data augmentation strategies could also be considered to generate the extended data. After pre-processing the dataset, which converts the data to augmented samples with an appropriate rolling window, each augmented instance has two data sequences representing the RSSI and the SINR. Then, we feed the extended samples to MH-DNN, as in Fig. 4

B. Proposed Majority Voting Algorithm

DAtR uses TSA and MVA as pre-processing and post-processing techniques, respectively. After feature classification is done in the Softmax layer, we use the MVA to reclassify the features to have better accuracy.

MVA divide into two methods (see Fig. 4). In Method 1, MVA uses one hot encoding probability values between 0 and 1 as input from the MH-DNN classification prediction and rounds them. This process applies to all augmented instances made from the previously explained TSA method for each sample. Next, the mean of all four results is calculated and used to classify the sample into three classes. Suppose the sample is classified in class 1 (attack) or 2 (no attack). In that case, the code finishes, the classification achieves high accuracy, minimal false alarms, and the number of features in class 3 (no decision) is low. However, if the feature is classified in class 3, we try to reclassify using other ML algorithms. In Method 2, we try to classify the samples as class 1 or 2 by inverting the algorithm order. Instead of rounding them first and then calculating the mean, we calculate the mean of probability values and then round them. If after Method 2, the feature can not be classified in class 1 or 2, we apply other well-known ML algorithms to classify the features that methods 1 or 2 could not classify. Notice that although the proposed DAtR results are efficient in LoS channel conditions (as will be demonstrated in Section V), the motivation for using pre-processing and post-processing techniques in MH-DNN arises from the fact that the attack detection accuracy might decrease in cases of low received power conditions, as they happen in NLoS channel conditions. As such, we target to increase accuracy by applying TSA and MVA. In the end, DAtR proved to be efficient also in LoS conditions. Algorithm 1 illustrates the details of methods 1 and 2, where τ is the primary sample, and Aug represents the four augmented samples for the τ example. When categorizing features into classes in the Softmax layer is impossible, the algorithm tries to classify them. For example, a sample classifies as a specific class 1 or 2 if 3 of

Algorithm 1: Majority Voting Algorithm.

Require: τ, Aug
Ensure: Assign τ to Classes 1 or 2 or 3
  $Class\ 1\ ||\ Class\ 2 \leftarrow$ Classify Aug
  if $3Aug/4 \geq Class\ 1$, **then**
    $Class\ 1 \leftarrow$ Classify τ
  else if $3Aug/4 \geq Class\ 2$, **then**
    $Class\ 2 \leftarrow$ Classify τ
  else if $Aug/2 == Class\ 1$ and another
   $Aug/2 == Class\ 2$ **then**
    $Class\ 3 \leftarrow$ Classify τ
  end if

TABLE VII
NETWORK PARAMETERS

Scenario Parameters	Values
Terrestrial Users	0, 3, 5, 10, 20, and 30
Authenticated UAVs	1
Small Cells	10
Small cell height	10 m
Attackers	0, 1, 2, 3, and 4
Speeds	10 m/s
Modulation scheme	OFDM
Small cell power	4 dBm
Authenticated UAV power	2 dBm
Attackers power	0, 2, 5, 10, and 20 dBm
Authenticated UAV position	URD*
Attackers position	URD*
Small cells position	URD*
Scenario	UMi
Distance	100, 200, 500, and 1000 m
Simulation time	30 s

*URD - Uniformly Random Distributed.

its four augmented instances classify in the same class. In the case of a draw, the feature goes into class 3.

V. SIMULATION RESULTS

In this section, we present the performance evaluation of the proposed DAtR. In particular, we provide five experimental outcomes related to the robustness of the DAtR. First, we conduct a comparative study on the efficacy of different layers, such as Attention and LSTM, in the MH-DNN architecture. Then, we study the effect of the window size on the DAtR's accuracy. In addition, we examine the performance of the proposed DAtR when we remove parts of the dataset from training, and we benchmark the DAtR's accuracy against six machine learning alternatives. All these experiments evaluate LoS and NLoS channel conditions separately. To evaluate the DAtR's performance, we compare the overall accuracy based on the various parameters available in the dataset. Initially, we analyze the accuracy as a function of the number of attackers and attackers' power. After that, we analyze the accuracy as a function of the attackers' distance and power. These simulations set all three conditions presented in the article: LoS, NLoS, and a combination of both. For this section, we adopt attacker amount N_{att}, attacker power P_{att}, users amount N_u, and distance d. Table VII presents the parameters used in the simulation. The speed remains the same for all scenarios, and the distances in Table VII refer to the distances between the small cell and authenticated UAVs.

A. The Window Size Impact

Fig. 5(a) and (b) show the window size impact on the final accuracy for LoS and NLoS conditions using the MH-DNN (no improvements, no TSA, and no MVA).

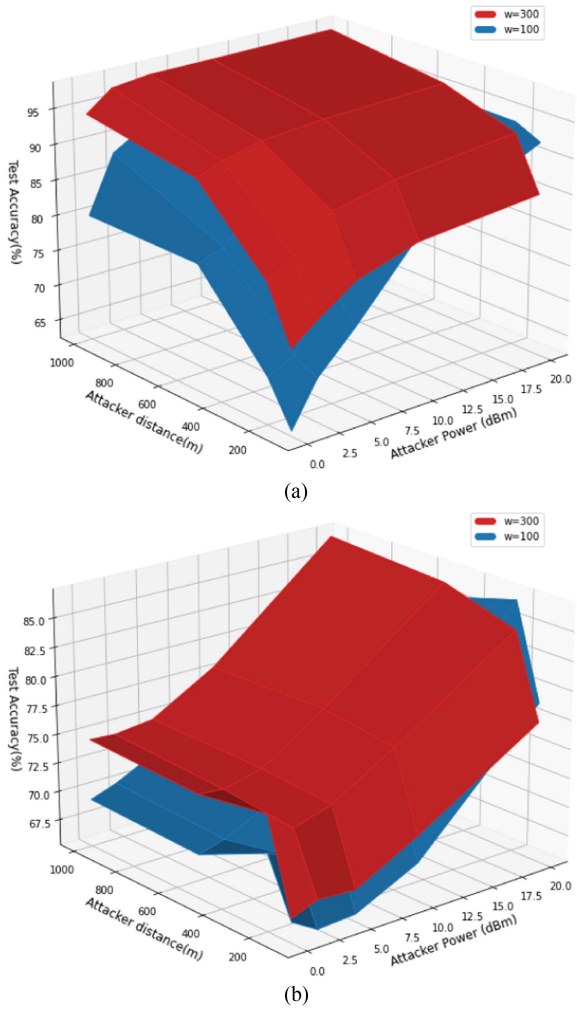


Fig. 5. Impact of the window sizes $w = 100$ and $w = 300$ (a) In LoS condition, (b) In NLoS condition.

Fig. 5(a) indicates that the accuracy range for $w = 100$ is roughly 65% to 90%, whereas the range for $w = 300$ is approximately 75% to 95%. In the NLoS case (see Fig. 5(b)), the MH-DNN achieves a range of about 67% to 85% when $w = 100$, and the percentage ranges from 70% to 87% when $w = 300$. Both figures demonstrate that the accuracy is directly proportional to the window size, independently of the channel condition. It is worth noting that there is a small trade-off between the time it takes to calculate the estimate for each class and the available resources, as will be demonstrated later in Fig. 12.

B. Attention vs. LSTM

Both the LSTM and Attention layers are trying to solve the same problem. They keep track of the old input sequences in the current node or state. For example, the information flowing from t_0 to $(t - n)$ is available in a modified/partial form in the state at time t. The algorithm uses the modified form to establish a relationship with the incoming data. We compare LSTM and Attention regarding window size and final accuracy

TABLE VIII
DIFFERENCES IN THE OVERALL ACCURACY FOR EACH CONDITION AND FOR EACH WINDOW SIZE (w)

			w = 50	100	200	300
DNN	LoS	**Attention**	**82.26**	83.04	**88.35**	**89.59**
		LSTM	79.62	**84.67**	86.51	88.06
	NLoS	**Attention**	**72.58**	**73.00**	**74.12**	**75.60**
		LSTM	69.43	71.46	65.76	68.67
	Both	**Attention**	**76.31**	**79.59**	**79.19**	**82.77**
		LSTM	76.07	78.19	77.10	77.29
DNN+Method 1	LoS	**Attention**	**83.88**	84.31	**88.48**	**89.98**
		LSTM	83.65	**84.38**	87.10	88.34
	NLoS	**Attention**	**82.81**	82.53	**82.94**	**83.07**
		LSTM	81.87	**83.05**	81.27	80.19
	Both	**Attention**	**80.50**	**81.27**	**79.13**	**83.66**
		LSTM	79.82	79.67	78.95	79.02
DNN+Method 2	LoS	**Attention**	**84.10**	84.77	**89.99**	**90.80**
		LSTM	81.34	**86.26**	88.47	89.49
	NLoS	**Attention**	**75.66**	**76.07**	**77.13**	**79.00**
		LSTM	72.20	73.85	68.60	73.10
	Both	**Attention**	**78.61**	**81.52**	**80.51**	**84.65**
		LSTM	78.28	80.11	79.22	79.59

TABLE IX
ACCURACY MEASUREMENTS USING THE XGB ALGORITHM FOR EACH CONDITION WITH DIFFERENT WINDOW SIZES (w)

w	50	100	200	300
LoS	**83.27**	**83.69**	**85.57**	**86.33**
NLoS	83.04	82.58	83.41	80.58
Both	79.65	79.47	78.40	78.85

improvements in LoS and NLoS conditions for each proposed algorithm in the article.

The trainable parameters do not change between the different window sizes or conditions. In our example, the MH-DNN configured with LSTM has 59,984 trainable parameters compared to 64,368 in the one with the Attention. However, most well-known deep neural networks, such as VGG [41] and ResNet [42], employ more than one million trainable parameters in their architectures, which increases the overall training time and, consequently, the prediction time. Also, they require more computation capabilities. Therefore, we only interchange the Attention and LSTM layers using Table V settings and the proposed DAtR. Table VIII shows the differences in the overall accuracy between the Attention and LSTM layers for different window sizes (ranging from $w = 50$ to $w = 300$), various channel conditions (LoS, NLoS, and both), and the three proposed methods (MH-DNN, MH-DNN + Method 1, MH-DNN + Method 2). Table IX compares results to the reference XGB algorithm for different window sizes and channel conditions. The XGB performs poorly when the hybrid dataset is applied to the algorithm in contrast to the results obtained with the DNN and DNN with methods 1 and 2. In comparing the LSTM with Attention, except for four states, better results are almost seen in the Attention layer. For example, in MH-DNN + Method 1 in NLoS condition with window size $w = 100$, LSTM performs slightly better, where its difference with Attention is around 0.52%.

Moreover, we notice that an increase in the window size positively impacts the overall accuracy when using Attention layers. For LSTM in NLoS conditions, it has the opposite effect when $w > 100$. Pattern recognition in NLoS is generally hard to extract due to the low power received in the authenticated UAV. Still, for this particular case, when $w > 100$, it decreases the overall accuracy. Concerning the LoS, NLoS, and Both conditions, LoS presents the best accuracy because there is no decrease in the received power due to obstacles and objects between the authenticated UAV and the small cell. Therefore, the deep network could learn the attacker pattern even in cases with channel variations and more users in the network. The combined condition presents the second-best results; as expected, NLoS shows the worst. Notice that by adding more nodes and layers, the deep network can learn this pattern; however, there is a trade-off in terms of memory and energy consumption, which is outside the scope of this work. The most significant impact of the MVA and TSA in the DNN is in NLoS conditions. Method 1 increases the overall accuracy by more than 10% when using LSTM and by approximately 10% with Attention. Among the methods in the study, the MH-DNN + Method 2 performs better for LoS, whereas the MH-DNN + Method 1 performs better for NLoS conditions. Fig. 6 depicts the accuracy against the distance between the authenticated UAV and the small cell in the network for two different window sizes using Attention and LSTM layers for (a) LoS and (b) NLoS channel conditions. For each condition, we present the results for MH-DNN with no additional methods. Fig. 6(a) shows that, for LoS, both Attention and LSTM configurations with window size 300 ($w = 300$) outperform the structures with window size 50 ($w = 50$). In the NLoS condition, see Fig. 6(b), the DNN embedded with the Attention layer performs better independently of the window size.

C. Comparison With Other Machine Learning Classifiers

Fig. 7 compares the proposed DAtR (composed by MH-DNN, Method 1, and Method 2) with three other machine learning methods, namely RF, CAT, and XGB, over the distances between the small cell and the authenticated UAV available in the dataset, in LoS and NLoS conditions, separately.

We eliminate GNB and LR from the charts because they fail to achieve 70% accuracy across the range of distances and SVM because its performance is comparable to the other ML algorithms for shorter distances but dropped to 75% accuracy for those with d > 200 in LoS conditions. In Fig. 7(a), we show that even our primary classifier, which is the MH-DNN embedded alone with the Attention layer, consistently outperforms well-known classifiers such as RF, CAT, and XGB, while Method 1 and 2 present an additional improvement, especially for considerable distances. CAT and XGB perform similarly, while RF decreases its accuracy for significant distances. Compared to all the accuracies obtained from other algorithms, the proposed DAtR

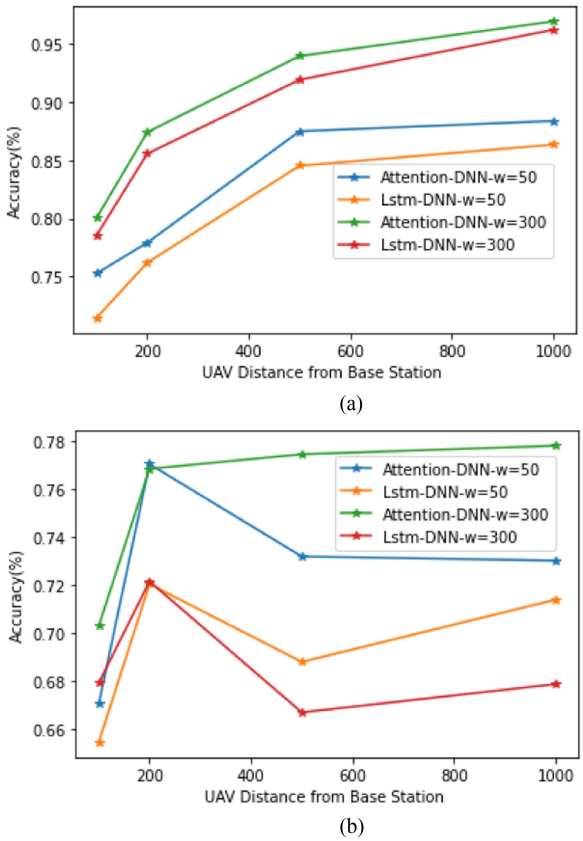


Fig. 6. Comparison between Attention and LSTM algorithms for $w = 50$ and $w = 300$, $N_u = 20$, $N_{att} = 2$, and $P_{att} = 5$ dBm. (a) In LoS condition, (b) In NLoS condition.

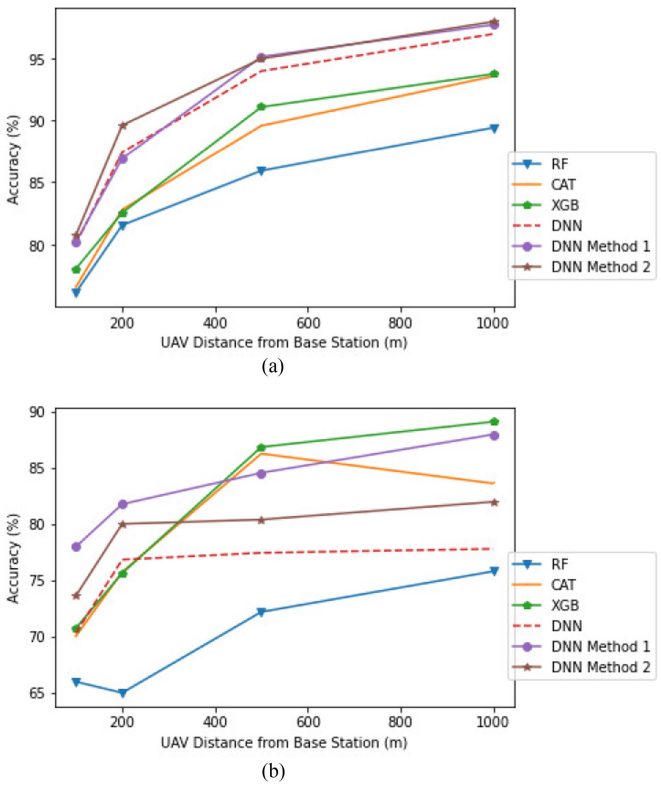


Fig. 7. Comparison between the proposed MH-DNN with MH-DNN + Method 1, MH-DNN + Method 2, RF, CAT, XGB. $w = 300$, $N_u = 20$, $N_{att} = 2$, $P_{att} = 5$ dBm. (a) In LoS condition, (b) In NLoS condition.

achieves an accuracy range from 80% up to 95% overall distance ranges. The mean accuracy that the DAtR achieves is 89.97%, while the RF, CAT, and XGB achieved 83.24%, 85.60%, and 86.33%, respectively. Fig. 7(b) presents the results for the NLoS channel condition. This Fig. shows that Method 1, in this case, is more effective in short distances. However, note that the DAtR and Method 2 outperform the benchmark schemes for short distances but lose accuracy for higher distances. As such, Method 1 appears to achieve a good compromise between small and large distances. Comparing both charts, DAtR can more easily identify attackers in LoS, but it can also be implemented in NLoS or mixed conditions depending on the link distance.

D. Attacker Number and Power

Fig. 8 presents the accuracy over the number of attackers and their power in (a) LoS, (b) Combined, and (c) NLoS conditions. If we look closely at the individual charts, we see that the accuracy increases with more attackers and more power for LoS and combined conditions. In the NLoS case, the low accuracy is centered in the scenario with two attackers when both are configured with power less than 5 dBm. After that, it increases for more and fewer attackers, and as the attacker increases, power rises.

In the LoS case, the scenario with one attacker is the hardest for the proposed algorithms to learn. In the Combined condition, 0 and 1 attacker scenarios are complicated for the algorithms to understand, and for the NLoS condition, the most complex scenario is with two attackers. In LoS and Combined cases, the changes in the power presented improvements in the accuracy of around 5%. The low accuracy when there are fewer than three attackers in the scenario might be justified by the stochastic channel models available in 5G UAV cases where the channel adjustments experienced by the UAV can change approximately 30 dB from one channel update to another. The amount of users affects the total received power reducing the DAtR's overall accuracy. In the NLoS case, the fact that no straight rays are feeding into the receiver impacts the overall power received and decreases the accuracy of results. By comparing all the results, the NLoS simulation presents the lowest overall accuracy from all conditions, but the best accuracy it can achieve is 93% with four attackers configured with 20 dBm power.

E. Confusion Matrices

Fig. 9(a) and (b) illustrate the confusion matrices resulting from the proposed algorithms: MH-DNN, MH-DNN + Method 1 + ML algorithm, and MH-DNN + Method 2 + ML algorithm, for LoS and NLoS, respectively. In addition, we utilize the XGB as an ML algorithm for Methods 1 and 2.

We compare the results of MH-DNN with Method 1 and Method 2 with the results of MH-DNN alone. We notice that MH-DNN + Method 2 + XGB increases the accuracy in LoS scenarios, while MH-DNN + Method 1 + XGB is more suitable

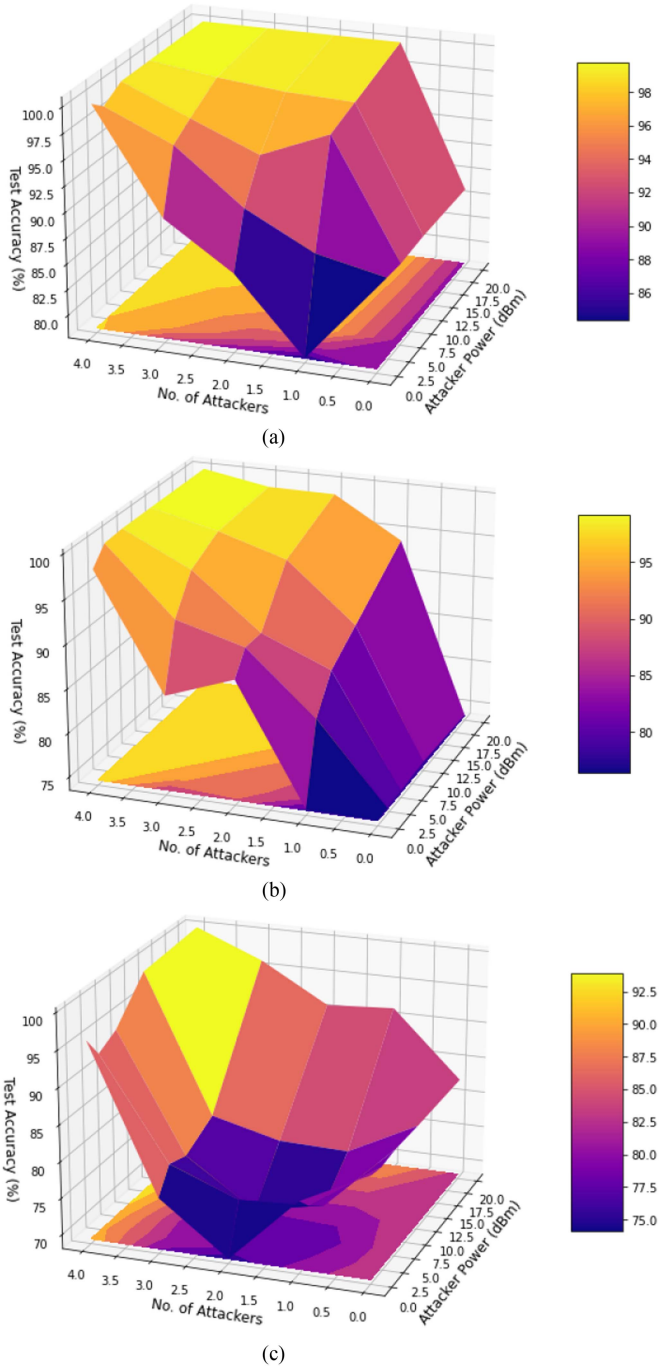


Fig. 8. Accuracy vs. Attackers Number and Attacker Power, $N_u = 20$, $d = 100$ m, $w = 300$, (a) In LoS condition only, (b) In LoS and NLos conditions, (c) In NLoS condition only.

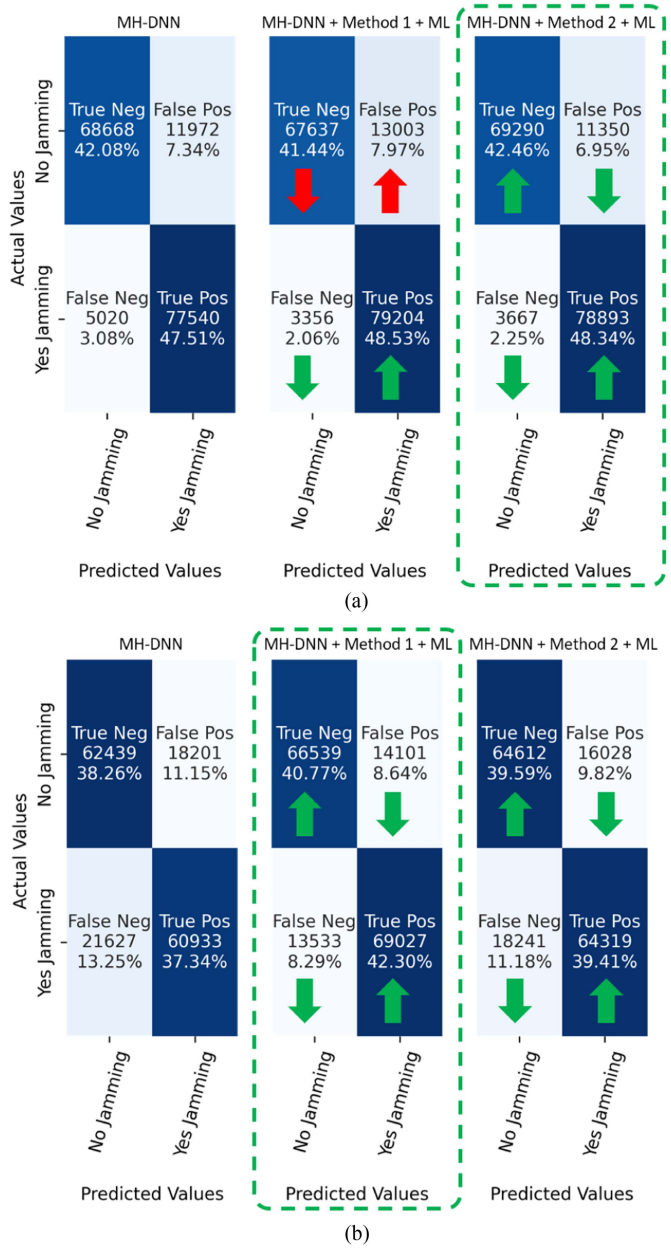


Fig. 9. Overall Confusion Matrices of the proposed MH-DNN, MH-DNN + Method 1 and ML algorithm, and MH-DNN + Method 2 and ML algorithm, $w = 300$, (a) In LoS condition, (b) In NLoS condition. Green arrows indicate enhancement, while the red ones refer to reduction.

for NLoS settings. For example, Fig. 9(a) highlights the difference between the two True Negative (`True Neg`) when we subtract Method 1 and Method 2 values from the MH-DNN. Method 1 + XGB results in -0.64% less accuracy, while with Method 2 + XGB, there is +0.38% better accuracy. Also, Method 1 increases the chances of False Positive (`False Pos`) by +0.63%, while Method 2 decreases the likelihood of `False Pos` by -0.39%. We see the opposite effect in Fig. 9(b). Method 1 + XGB has better values for `True Neg` and `False Pos` than Method 2 + XGB when comparing both to the Deep Network. Regarding LoS, the MH-DNN + Method 2 performs better than the other approaches in the research, but the MH-DNN + Method 1 is the clear winner when it comes to NLoS. Taking into account the best outcomes that we have so far, specifically, MH-DNN configured with Attention + Method 2 for LoS or + Method 1 for NLoS and XGB algorithm, except when explicitly mentioned, we use this configuration to show detailed performance evaluation considering all cases and parameters available in the dataset using DAtR. In the combined condition, we used MH-DNN configured with Attention + Method 1 for NLoS and XGB algorithm. The accuracy presented in the confusion

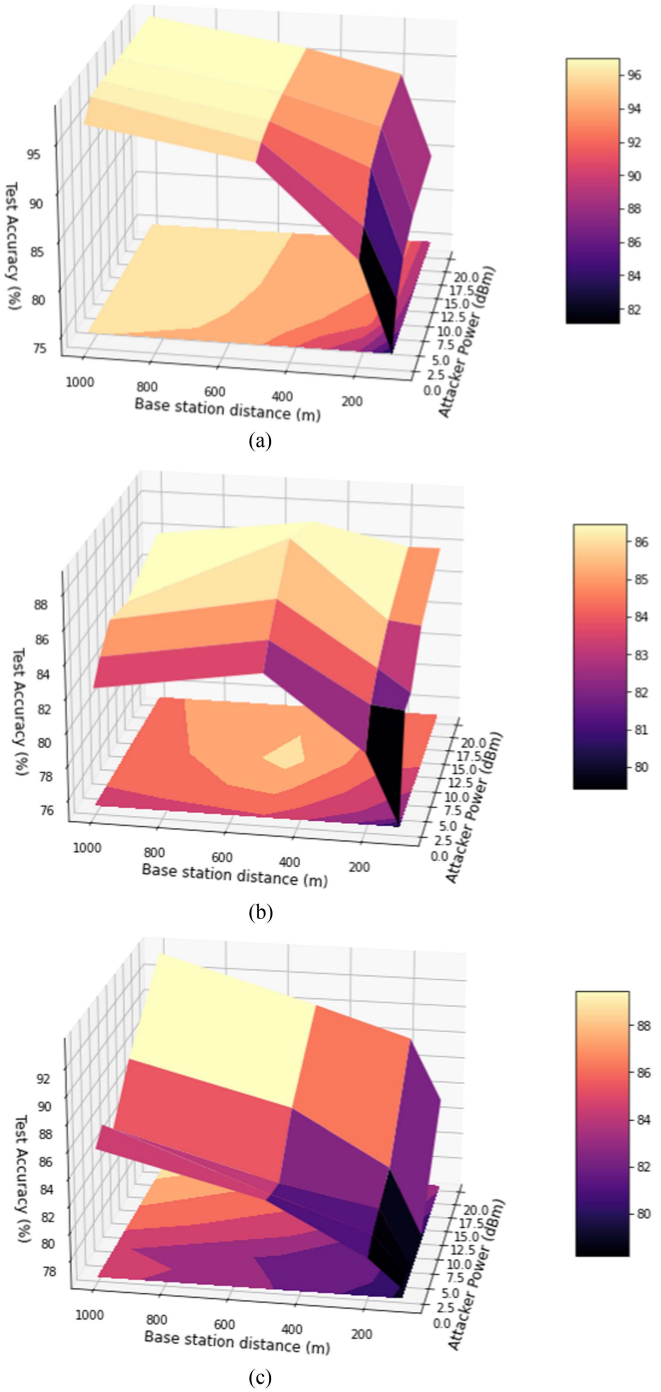


Fig. 10. Accuracy vs. Attackers Power and Attacker Distance test data, window size $w = 300$, $N_{att} = 2$, $N_u = 20$ (a) In LoS condition only, (b) In LoS and NLoS condition, (c) In NLoS condition only.

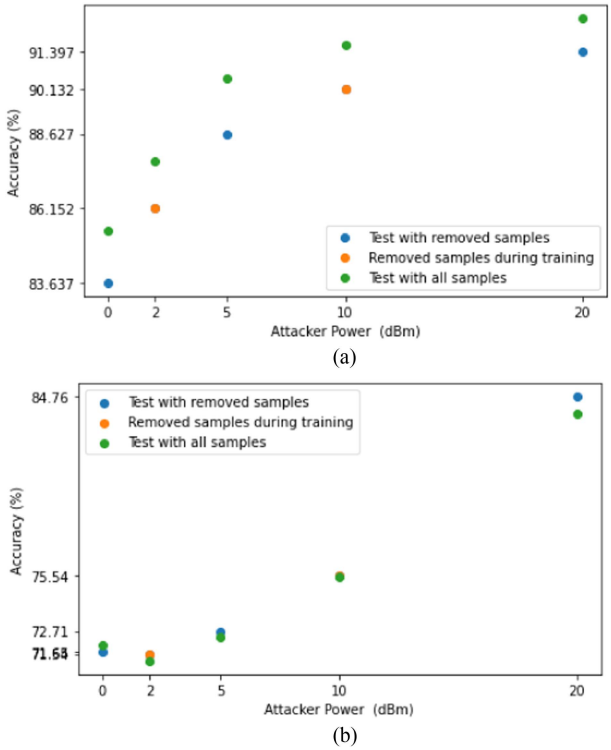


Fig. 11. Comparison with data that is not in the training $N_{att} = 2$, $N_u = 20$, and $d = 500$ m (a) In LoS condition, (b) In NLoS condition.

matrix is the average accuracy from all the scenarios in the dataset. It significantly impacts the specific cases, as shown in the following sections.

F. Attacker Power and Distance

Fig. 10 shows the accuracy over distance and attackers' power ratios during training for the three conditions: LoS, Combined, and NLoS. In the three conditions, attackers with lower power are more challenging for the deep network to recognize. In the LoS conditions, the deep network can identify attacks even though the base station is 1000 m away from the authenticated UAV and the attacker power is lower than 5 dBm with 96% accuracy. Of course, there are improvements when the power increases, but we achieve better results when increasing distance. In addition, the user interference decreases at this position so that the deep network can achieve high accuracy. In the Combined condition, we see the impact of power on accuracy more clearly than in LoS. For example, when the attacker power is set to 15 dBm, the accuracy is 85% when the distance between the authenticated UAV and the Base station is 100 m. However, we see a peak accuracy when the distance is 500 m and the attacker power is 15 dBm. While it is easier to identify attackers for the other conditions when the attacker power is higher than 5 dBm, in the NLoS condition, the attacker power needs to be adjusted to 15 dBm so the deep network can have approximately 84% accuracy.

G. Comparison With Data That is not in the Training

Fig. 11(a) and (b) depict the accuracy results based on the attacker power when the network users are $N_u = 20$, for a distance of 500 m, and two attackers. We remove the data related to the attacker power of 2 dBm and 10 dBm from the training. Therefore, the deep network sees both these pieces of data for the first time during testing. We executed this simulation for LoS and NLoS conditions.

TABLE X
PREDICTION TIMING VERSUS WINDOW SIZE (w) FOR THE PROPOSED DEEP NETWORK AND THREE OTHER ML CLASSIFIERS

w	50	100	200	300
DNN-Attention	30.9 ms ± 248 μs	30.9 ms ± 335μs	31.9 ms ± 656μs	30.8 ms ± 391μs
DNN-LSTM	31.3 ms ± 1.03 ms	31 ms ± 351 μs	31.2 ms ± 311 μs	30.5 ms ± 393 μs
CAT	**0.52** ms ± 561 ns	0.82 ms ± 744 ns	1.49 ms ± 939 ns	2.19 ms ± 2.02 μs
RF	71.6 ms ± 1.28 ms	74.8 ms ± 1.63 ms	76.6 ms ± 1.66 ms	79.4 ms ± 1.76 ms
XGB	0.66 ms ± 22.4 μs	**0.67** ms ± 22.5 μs	**0.68** ms ± 23.9 μs	**0.74** ms ± 21.6 μs

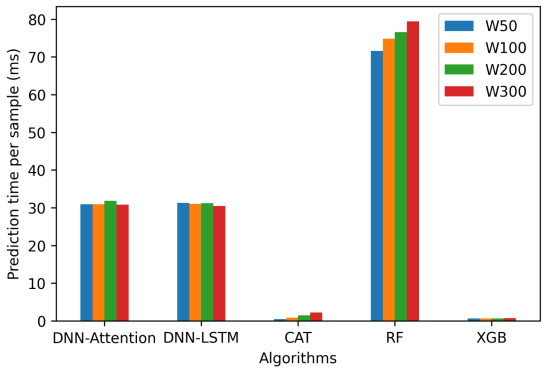


Fig. 12. Average processing time for each classifier.

Fig. 11(a) demonstrates the outcomes for LoS. A comparison of training with all and removed samples noticed a proportional decrease in all instances. This difference is around 1.5%. For the NLoS case, illustrated in Fig. 11(b), there is a difference more significant than 0.5% only when the attacker was set up with 20 dBm power. There are no significant differences for the other cases, which shows the robustness of our proposed algorithm.

H. Average Processing Time

Fig. 12 compares the average prediction time after training for the three baseline classifiers (RF, CAT, and XGB) and the proposed MH-DNN configured with Attention or LSTM for different window sizes to classify each sample. Table X shows the average values with their respective standard deviations. The prediction time is essential because it shows the latency in discovering attacks using such UAV algorithms. All timing tests were done using an Nvidia RTX 3090 GPU system.

In Fig. 12, we can see that the window size has a negligible effect on the XGB and the MH-DNN configured with Attention or LSTM. However, it has a more significant impact on CAT and RF. For example, the prediction time for CAT increases four times when the window size is 300 ($w = 300$). For RF, the impact of the window size is smaller than CAT, but it still increases by approximately 10% for the same window size ($w = 300$). There is a minor difference between the LSTM and Attention prediction times. The RF algorithm displays the highest prediction time. Our proposed method has a good trade-off between accuracy and prediction time.

VI. CONCLUSION

This article studied the attacks Self-Identifying problems in 5G UAV networks assuming scenarios with LoS, NLoS, and a probabilistic combination of both conditions. Specifically, we proposed a small deep network system denominated DAtR, that can cope with the attack Self-Identifying problem, and we verified its accuracy through extensive simulation campaigns. Along with the application and deep network design, our work innovates by combining both RSSI and SINR signals within the deep network and incorporating two novel pre and post-processing methods to increase accuracy. Our research examined five major implementation issues related to the deep network: how the key parameters, such as the window size, impact the deep network accuracy, the impact of different layers on the overall performance (i.e., Attention vs. LSTM), its performance compared to other machine learning alternatives for classification, the robustness of our deep network using data that is not available in training, and the prediction timing for the proposed DAtR. Compared to six popular classifiers available in the literature, we showed that the proposed system is a competitive option for the attack classification for all distance ranges in LoS conditions and for short-range distances in NLoS conditions. The comparison between LSTM and Attention shows that increasing the window size in the LSTM setup reduced the performance, while with Attention, it boosted performance. Attention layers in DAtR outperformed the same system configured with LSTM. Finally, we present the performance graphs we created for each case study. Results have demonstrated that our deep network reliably identifies attacks across all possible configurations. Identifying attacks in simulations with three or more attackers, fewer users, and a power of 10 dBm or higher was more straightforward. The identification accuracy was also affected by the three-dimensional distance between the small cell and the authenticated UAV. Here, the chances of identification improved with increasing distances since there was less interference.

REFERENCES

[1] B. K. S. Lima et al., "Aerial intelligent reflecting surfaces in MIMO-NOMA networks: Fundamentals, potential achievements, and challenges," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 1007–1024, 2022.
[2] W. Jin, J. Yang, Y. Fang, and W. Feng, "Research on application and deployment of UAV in emergency response," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emerg. Commun.*, 2020, pp. 277–280.
[3] G. Geraci et al., "What will the future of UAV cellular communications be? A flight from 5G to 6G," *IEEE Commun. Surv. Tut.*, vol. 24, no. 3, pp. 1304–1335, Thirdquarter 2022.
[4] X. Wang et al., "Joint flying relay location and routing optimization for 6G UAV & IoT networks: A graph neural network-based approach," *Remote Sens.*, vol. 14, no. 17, 2022, Art. no. 4377. [Online]. Available: https://www.mdpi.com/2072-4292/14/17/4377
[5] H. Kang, X. Chang, J. Mišić, V. B. Mišić, J. Fan, and J. Bai, "Improving dual-UAV aided ground-UAV bi-directional communication security: Joint UAV trajectory and transmit power optimization," *IEEE Trans. Veh. Technol.*, vol. 71, no. 10, pp. 10570–10583, Oct. 2022.

[6] M. M. Azari, F. Rosas, and S. Pollin, "Cellular connectivity for UAVs: Network modeling, performance analysis, and design guidelines," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3366–3381, Jul. 2019.
[7] B. Li, Z. Fei, and Y. Zhang, "UAV communications for 5G and beyond: Recent advances and future trends," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019.
[8] M. Vaezi et al., "Cellular, wide-area, and non-terrestrial IoT: A survey on 5G advances and the road toward 6G," *IEEE Commun. Surv. Tut.*, vol. 24, no. 2, pp. 1117–1174, Secondquarter 2022.
[9] B. Li, Q. Li, Y. Zeng, Y. Rong, and R. Zhang, "3D trajectory optimization for energy-efficient UAV communication: A control design perspective," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 4579–4593, Jun. 2022.
[10] B. Li, J. Zhang, L. Dai, K. L. Teo, and S. Wang, "A hybrid offline optimization method for reconfiguration of multi-UAV formations," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 1, pp. 506–520, Feb. 2021.
[11] K. L. Teo, B. Li, C. Yu, and V. Rehbock, *Applied and Computational Optimal Control*. Berlin, Germany: Springer, 2021, doi: 10.1007/978-3-030-69913-0.
[12] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "AFRL: Adaptive federated reinforcement learning for intelligent jamming defense in FANET," *J. Commun. Netw.*, vol. 22, no. 3, pp. 244–258, 2020.
[13] N. Liu, X. Tang, R. Zhang, D. Wang, and D. Zhai, "A DNN framework for secure transmissions in UAV-relaying networks with a jamming receiver," in *Proc. IEEE 20th Int. Conf. Commun. Technol.*, 2020, pp. 703–708.
[14] N. Souli, P. Kolios, and G. Ellinas, "An autonomous counter-drone system with jamming and relative positioning capabilities," in *Proc. IEEE Int. Conf. Commun.*, 2022, pp. 5110–5115.
[15] D. Darsena, G. Gelli, I. Iudice, and F. Verde, "Detection and blind channel estimation for UAV-aided wireless sensor networks in smart cities under mobile jamming attack," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11932–11950, Jul. 2022.
[16] O. Sharifi-Tehrani, M. F. Sabahi, and M. Danaee, "GNSS jamming detection of UAV ground control station using random matrix theory," *ICT Exp.*, vol. 7, no. 2, pp. 239–243, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959520303040
[17] D. Su and M. Gao, "Research on jamming recognition technology based on characteristic parameters," in *Proc. IEEE 5th Int. Conf. Signal Image Process.*, 2020, pp. 303–307.
[18] M. Cheng, Y. Ling, and W. B. Wu, "Time series analysis for jamming attack detection in wireless networks," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–7.
[19] Y. Shi, X. Lu, Y. Niu, and Y. Li, "Efficient jamming identification in wireless communication: Using small sample data driven naive Bayes classifier," *IEEE Wireless Commun. Lett.*, vol. 10, no. 7, pp. 1375–1379, Jul. 2021.
[20] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Trans. Mobile Comput.*, vol. 13, no. 8, pp. 1746–1759, Aug. 2014.
[21] J.-C. Li, J. Liu, B.-G. Cai, and J. Wang, "Jamming identification for GNSS-based train localization based on singular value decomposition," in *Proc. IEEE Intell. Veh. Symp.*, 2021, pp. 905–912.
[22] A. Krayani, A. S. Alam, L. Marcenaro, A. Nallanathan, and C. Regazzoni, "Automatic jamming signal classification in cognitive UAV radios," *IEEE Trans. Veh. Technol.*, vol. 71, no. 12, pp. 12972–12988, Dec. 2022.
[23] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," in *Proc. IEEE Int. Conf. Inf. Netw.*, 2020, pp. 459–464.
[24] M. P. Arthur, "Detecting signal spoofing and jamming attacks in UAV networks using a lightweight IDS," in *Proc. IEEE Int. Conf. Comput., Inf. Telecommun. Syst.*, 2019, pp. 1–5.
[25] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5G cloud radio access networks," in *Proc. IEEE Int. Symp. Netw., Comput. Commun.*, 2020, pp. 1–5.
[26] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, Jul. 2019.
[27] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010. [Online]. Available: http://papers.nips.cc/paper/7181-attention-is-all-you-need
[28] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, 2017.
[29] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.
[30] Y. Li et al., "Jamming detection and classification in OFDM-based UAVS via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16859–16870, 2022.
[31] J. Gao, M. Wang, L. Chen, B. Hui, C. Wang, and H. Fan, "DRFM jamming mode identification leveraging deep neural networks," in *Proc. IEEE Int. Conf. Control, Automat. Inf. Sci.*, 2021, pp. 444–449.
[32] F. Ruo-Ran, "Compound jamming signal recognition based on neural networks," in *Proc. IEEE 6th Int. Conf. Instrum. Meas., Comput., Commun. Control*, 2016, pp. 737–740.
[33] "3GPP - Technical specification group radio access network; study on enhanced LTE support for aerial vehicles," 2018. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3231
[34] "3GPP - Technical specification group radio access network; study on channel model for frequencies from 0.5 to 100 GHz," 2020. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173
[35] L. F. Henderson, "The statistics of crowd fluids," *Nature*, vol. 229, no. 5284, pp. 381–383, 1971. [Online]. Available: https://doi.org/10.1038%2F229381a0
[36] J. Viana et al., "A convolutional attention based deep learning solution for 5G UAV network attack recognition over fading channels and interference," in *Proc. IEEE 96th Veh. Technol. Conf.*, 2022, pp. 1–5.
[37] H. Farkhari, J. Viana, P. Sebastião, L. Bernardo, S. Kahvazadeh, and R. Dinis, "Accurate and reliable methods for 5G UAV jamming identification with calibrated uncertainty," in *Proc. 17th Int. Conf. Research Challenges Inf. Sci.*, 2023. [Online]. Available: http://hdl.handle.net/10071/28846
[38] S. Ruder, "Deep learning for NLP best practices," 2017. [Online]. Available: http://ruder.io/deep-learning-nlp-best-practices/
[39] N. I. Levi, I. M. Bloach, M. Freytsis, and T. Volansky, "Noise injection node regularization for robust learning," in *Proc. 11th Int. Conf. Learn. Representations*, 2023. [Online]. Available: https://openreview.net/forum?id=gmSZ-GPNY6
[40] T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of tricks for image classification with convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 558–567.
[41] Keras, "VGG16 and VGG19," 2022. [Online]. Available: https://keras.io/api/applications/vgg/
[42] Keras, "RESNet and RESNetv2," 2022. [Online]. Available: https://keras.io/api/applications/resnet/

Joseanne Viana received the bachelor's degree in telecommunication engineering from the University of Campinas, Campinas, Brazil. He is working toward the Ph.D. degree with Signal Processing and Communications Department, University Carlos III de Madrid, Getafe, Spain She is an Early-Stage Researcher in the project TeamUp5G, a European Training Network in the frame of (MSCA ITN) of the European Commission's Horizon 2020. Her research focuses on wireless communications applied to interconnected systems such as UAVs, aerial vehicles, and non-terrestrial devices.

Hamed Farkhari is working toward the Ph.D. degree with ISCTE - Lisbon University Institute,Lisbon, Portugal. He is an Early-Stage Researcher with the TeamUp5G group, a European Training Network in the frame of (MSCA ITN) of the European Commission's Horizon 2020. He is also a Researcher and Developer with PDMFC Company, Lisbon, Portugal. His research interests include cybersecurity, machine learning, deep learning, data science, meta-heuristic, and optimization algorithms.

Pedro Sebastião received the Ph.D. degree in electrical and computer engineering from IST. He is currently a Professor with Information Science and Technology Department, ISCTE – University Institute of Lisbon, Lisbon, Portugal. He is also the Board Director of AUDAX-ISCTE - Entrepreneurship and Innovation Center, ISCTE, responsible for the LABS LISBOA Incubator and Researcher with the Institute of Telecommunications. He has oriented several master's dissertations and doctoral theses. He is the author or co-author of more than 200 scientific articles, and is responsible for several national and international Research and Development projects. He is an expert and evaluator of more than one hundred national and international Civil and Defense Research and Development projects. He was the recipient of scientific, engineering, and pedagogical awards. He has also organized or co-organized more than 55 national and international scientific conferences. He planned and developed several postgraduate courses in technologies and management, entrepreneurship and innovation, and transfer of technology and innovation. He supported several projects involving technology transfer and the creation of startups and spinoffs of value to society and the market. He developed the professional activity in the National Defense Industries, initially in the Office of Studies and later as the Board Director of the Quality Department of the Production of New Products and Technologies. He was also responsible for systems of communications technology in the Nokia-Siemens business area. His research interests include monitoring, control, communications of drones, unmanned vehicles, planning tools, stochastic processes (modeling and efficient simulations), the Internet of Things, and efficient communication systems.

Luis Miguel Campos received the B.Tech. degree from the Instituto Superior Técnico, Lisbon, Portugal, in 1992, the M.S. degree in information and computer science from the University of California, Irvine, CA, USA, in 1995, and the Ph.D. degree in information and computer science, in 1999. With 25 years of experience managing companies from the startup stage to medium size, he is focused on creating a self-sustainable virtuous cycle ecosystem of business angel funds, venture capital funds, active investors, researchers, and entrepreneurs, which will cover all stages of creation and growth until IPO. He has founded and led several companies, some of which have been sold to large companies, namely ZPX Interactive Software. He currently leads the Research and Development Team, PDMFC, Lisbon. He is involved in 12 European-funded research projects (Horizon2020) and five national research projects (Portugal2020).

Katerina (Aikaterini) Koutlia received the B.Sc. degree in electronics engineering from the Technological Institution of Thessaloniki, Thessaloniki, Greece, in 2009 and the M.Sc. degree with distinction in wireless communication systems from the Brunel University, Uxbridge, U.K., in 2011. In 2016, she received the Ph.D. degree with honors (supported by a grant from the Spanish Ministry of Education, Culture, and Sport) from the Polytechnic University of Catalonia (UPC), Barcelona, Spain. She was a Postdoctoral Researcher with the Mobile Communication Research Group, UPC, where she has been involved in several European and National Projects. In 2018, she was with CTTC, where she is currently a Researcher. Her main activities include developing and studying existing and novel 3GPP 4G, 5G, and B5G standard-compliant features using the LENA/5G-LENA system-level simulators and the maintenance and extension of the simulators under the framework of European International and Industrial projects.

Biljana Bojovic received the M.Sc. degree in electrical and computer engineering from the Faculty of Technical Sciences, Novi Sad, Serbia, in 2008 and the Ph.D. degree in networking engineering from the Polytechnic University of Catalonia, Barcelona, Spain, in 2022. She is the Developer and Maintainer of the LTE, NR, and NR-U modules of the ns-3 network simulator and the Principal Author of the LAA and LTE-U modules. She held LTE and NR module tutorials at the ns-3 workshops in 2016 and 2022 and CONFTELE in 2021. In addition, she was a Mentor of ns-3 GSoC on several occasions. In 2020, she was the recipient of the ACM SIGCOMM Networking System Award. She worked on many research projects for industrial clients, such as Wi-Fi Alliance, SpiderCloud, Interdigital, US Department of Defense, and NIST, Facebook. She is the Co-Author of one patent application (US20200314906A1). Her research interests include XR traffic enhancements for 5G-Advanced, MIMO simulation models for ns-3, and unlicensed/shared spectrum.

Sandra Lagén (Senior Member, IEEE) received the Telecommunications Engineering, M.S., and Ph.D. degrees from Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, in 2011, 2013, and 2016, respectively. In 2017, she was with CTTC, Castelldefels, Spain, where she is currently a Senior Researcher and Head of the Open Simulations (OpenSim) research unit. She has participated in outstanding projects within the industry, leading to the design and development of the open-source end-to-end 5G-LENA simulator. Her research interests include wireless communications, spectrum and interference management, and optimization theory. She was the recipient of the dissertation best national Ph.D. thesis on high-speed broadband mobile communications (2017) and a Special Doctoral Award from UPC (2019). She was also the recipient of IEEE WCNC 2018 and WNS3 2020 best paper awards. Since 2021, she has been a Member of the executive board of the ns-3 consortium.

Rui Dinis (Senior Member, IEEE) received the Ph.D. degree from the Instituto Superior Técnico (IST), Technical University of Lisbon, Lisbon, Portugal, in 2001. He was a Researcher with Centro de Análisee Processamento de Sinal, IST, from 1992 to 2005 and Instituto de Sistemas e Robótica from 2005 to 2008. He is currently a Senior Researcher with Instituto de Telecomunicações, Aveiro, Portugal and a Full Professor with FCT, Nova University of Lisbon, Lisbon. He is or was the Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE OPEN JOURNAL ON COMMUNICATIONS, and *Physical Communication* (Elsevier). He is an IEEE VTS Distinguished Speaker and an IEEE ComSoc Distinguished Lecturer.

2.4. Article #4: A Convolutional Attention-Based Deep Learning Solution for 5G UAV Network Attack Recognition

This paper introduces a novel deep learning approach for detecting attacks in 5G UAV networks using a convolutional attention-based architecture. The research focuses on realistic scenarios that include multiple terrestrial users, interference, and static and moving attackers.

Key Contributions

- Development of an attention-based deep learning architecture for attack detection;
- Comprehensive analysis of UAV networks with multiple terrestrial users;
- Evaluation of both static and moving attack scenarios;
- Implementation of efficient processing techniques for real-time detection.

The proposed architecture processes two key parameters:

- RSSI
- SINR

Performance Evaluation

The system demonstrated robust performance in various scenarios:

- Overall accuracy of 84% during training and 74% during testing;
- Effective detection with up to 20 terrestrial users;
- Successful identification of static and moving attackers;
- Performance maintained across different power levels and distances.

A significant innovation of this work is the implementation of the attention mechanism, which reduced the trainable parameters by approximately 50% compared to LSTM-based solutions while maintaining comparable performance. This makes the solution particularly suitable for resource-constrained UAV platforms.

Experimental Validation

The experimental validation included the following configurations:

- Various attacker configurations (0–4 attackers).
- Different power levels (0–20 dBm).
- Multiple user scenarios (0–20 users).
- Both static and dynamic scenarios.

Article Details

- **Title:** A Convolutional Attention-Based Deep Learning Solution for 5G UAV Network Attack Recognition
- **Date:** 2022
- **Authors:** Joseanne Viana, Hamed Farkhari, Luis Miguel Campos, Pedro Sebastião, Katerina Koutlia, Sandra Lagén, Luis Bernardo, Rui Dinis

- **Status:** Accepted in a Conference
- **Conference:** IEEE VTC Fall 2022
- **DOI:** 10.1109/VTC2022-Fall57202.2022.10012726

This research advances the field of UAV security by demonstrating the effectiveness of attention mechanisms in attack detection while maintaining computational efficiency. The comprehensive evaluation in realistic 5G scenarios with multiple users and moving attackers provides valuable insights for practical implementations.

A Convolutional Attention Based Deep Learning Solution for 5G UAV Network Attack Recognition over Fading Channels and Interference

Joseanne Viana [†‡], Hamed Farkhari[*†], Luis Miguel Campos [*], Pedro Sebastião [†‡],
Katerina Koutlia [¶], Sandra Lagén [¶], Luis Bernardo [§‡], Rui Dinis[§‡],
[†]ISCTE – Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
[*]PDMFC, Rua Fradesso da Silveira, n. 4, Piso 1B, 1300-609, Lisboa, Portugal
[‡]IT – Instituto de Telecomunicações, Av. Rovisco Pais, 1, Torre Norte, Piso 10, 1049-001 Lisboa, Portugal
[§]FCT – Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal
[¶] CTTC - Centre Tecnològic de Telecomunicacions de Catalunya (CERCA);
Emails: joseanne_cristina_viana@iscte-iul.pt, Hamed_Farkhari@iscte-iul.pt, luis.campos@pdmfc.com,
pedro.sebastiao@iscte-iul.pt, {kkoutlia, slagen}@cttc.es, rdinis@fct.unl.pt

2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall) | 978-1-6654-5468-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/VTC2022-Fall57202.2022.10012726

***Abstract*—When users exchange data with Unmanned Aerial Vehicles - (UAVs) over Air-to-Ground - (A2G) wireless communication networks, they expose the link to attacks that could increase packet loss and might disrupt connectivity. For example, in emergency deliveries, losing control information (i.e., data related to the UAV control communication) might result in accidents that cause UAV destruction and damage to buildings or other elements. To prevent these problems, these issues must be addressed in 5G and 6G scenarios. This research offers a Deep Learning (DL) approach for detecting attacks on UAVs equipped with Orthogonal Frequency Division Multiplexing - (OFDM) receivers on Clustered Delay Line (CDL) channels in highly complex scenarios involving authenticated terrestrial users, as well as attackers in unknown locations. We use the two observable parameters available in 5G UAV connections: the Received Signal Strength Indicator (RSSI) and the Signal to Interference plus Noise Ratio (SINR). The developed algorithm is generalizable regarding attack identification, which does not occur during training. Further, it can identify all the attackers in the environment with 20 terrestrial users. A deeper investigation into the timing requirements for recognizing attacks shows that after training, the minimum time necessary after the attack begins is 100 ms, and the minimum attack power is 2 dBm, which is the same power that the authenticated UAV uses. The developed algorithm also detects moving attackers from a distance of 500 m.**

***Index Terms*—Cybersecurity, Convolutional Neural Networks, Deep Learning, Jamming Detection, Jamming Identification, Unmanned Aerial Vehicles, 5G;**

I. Introduction

Unmanned Aerial Vehicles - (UAVs) will integrate into 5G and 6G networks to provide delivery services, security, general and risky inspections, emergency services, and other functions inside and outside the network. The logistics industry will benefit first from using UAVs in their ecosystem, followed by all other vertical industries. In addition to coverage, high throughput, and low latency requirements, there is an increasing demand for secure and reliable connections with powerful data protection [1]. We expect that emergency and high-value transportation, whose success depends on the capacity to communicate reliably and securely, will employ UAVs to provide high-quality services at lower costs [3]. Due to their aerial nature, UAVs provide faster and more flexible network services at higher data rates since they have complete control over their movement and a high probability of establishing robust Line-of-Sight (LoS) communication links. However, the vulnerability of wireless Air-to-Ground - (A2G) communication links make UAVs susceptible to attacks that increase packet loss or, even worse, completely lose communication. In order to keep UAV communications safe, it is crucial to detect potential risks and implement countermeasures. There is extensive research on Anti-Jamming techniques. Two established approaches to identify jamming are: analyzing the packet delivery ratio and the received signal strength. Both mechanisms deal with a high amount of lost information before detecting the attack. In ultra-dense networks, the overall amount of connected devices might hide the presence of local jammers. Finding other ways to address security issues in UAV networks is vital.

Currently, researchers are adopting machine learning techniques for sequence prediction problems with spatial inputs and pattern recognition [2]. As a part of machine learning, Deep Learning (DL) research exploits algorithms to make models with high-level data abstractions by using multiple processing layers with complex structures. Deep Neural Networks (DNNs) such as Convolutional Neural Networks (CNNs) [3], [4] with Long Short-Term Memory (LSTM) or attention layers are used for temporal modeling, and to define universal functions in complex wireless scenarios. [5] [4]. These characteristics make them suitable for applications that deal with time series and spatial data, such as interference identification in wireless networks. The signal under analysis uses specific features to detect anomalies. The authors in [6] add an attention layer in their CNN to track long temporal variations in the time domain gradients. Some pre-trained networks do not require re-design because they use transfer learning methods to learn classification procedures. For example, the authors in [7] use pre-trained networks (i.e., AlexNet, VGG-16, ResNet-50) to identify jamming using spectral images of the received signal in the UAV. These networks can be vast and require extensive processing to sort information, making them unsuitable for use by UAVs.

Even though embedded deep network techniques in the cloud or edge can monitor and evaluate channel degradation due to interference, fading, and jamming attacks, anti-jamming procedures and non-traditional approaches to avoid jamming are the focus of most research on this topic rather than recognizing attacks. As a result, there is a lack of publicly

available research on attack detection in UAV communications. We intend to detect attacks against authenticated UAVs when the UAVs are providing delivery services in highly complex environments such as the realistic ones in big urban centers. We aim to add terrestrial and aerial users connected to small cells that produce interference and simulate blockages that represent buildings. In this environment, the UAV is equipped with a unique deep network design with fewer layers than pre-trained networks typically use to identify attacks.

We organize this paper as follows: Section II details the system model. It explains the channel model, the dataset, and the architecture of the intended deep network. Section III summarizes the results of the performance evaluation of the deep network, and Section IV concludes this paper.

A. Contributions and Motivation

There is a lack of research and data on the detection and prevention of jamming using deep network techniques in UAV scenarios. In order to expand the literature on this topic, we present the following key contributions of this paper:

- A comprehensive case research model that assumes interference and blockages in the scenario with authenticated UAVs in 5G networks and the presence of other UAV attackers;
- An analysis of the identification of static and moving attackers in the network with and without terrestrial users;
- A smaller Convolutional Neural-Attention Based Network - (CNN-Attention) architecture to detect jamming;
- Insights into the deep network hyperparameters configuration;
- Comparison between deep network performances using attention or LSTM layers;
- Results on attack detection accuracy with and without terrestrial users in both static and moving scenarios;

Finally, we offer a visual representation of the confusion matrix for both the training and test datasets.

II. System Model

We consider a deep learning approach to detect attacks over UAV networks when there are V ($l \in \mathbb{N} \triangleq \{1, 2, ..., V\}$) authenticated UAVs connected to private networks in A2G links, S small cells serve U ground users, and M attackers exist with a fixed index $i \in \mathbb{N} \triangleq \{1, 2, ..., S\}$, $j \in \mathbb{N} \triangleq \{1, 2, ..., U\}$, $k \in \mathbb{N} \triangleq \{1, 2, ..., M\}$, respectively. The attackers are in unknown locations in the air and they can deliberately jam the signal received by the authenticated UAVs. The X-Y-Z Cartesian coordinate established between the small cells and the authenticated UAVs are defined as $||p_{bs} - p_{uav}||^2$. All the elements in the network follow slow fading and fast fading propagation characteristics according to [8] and [9]. The users are in random fixed positions and they can move when the proper configuration is set up. The small cells follow the same random location positioning strategy as the users. Fig 1 illustrates a top view of the simulation scenario. We define a total (1.0kmX1.0km) area that includes buildings of different sizes and heights, which are represented by rectangles. The "x" identifies the fixed S small cells available for connection. Some of the small cells are on the tops of the buildings. " • " represents the authenticated terrestrial users, "+" illustrates the attackers, and a variety of colors distinguishes the authenticated UAVs from the attackers. For the sake of simplicity, the authenticated UAVs stay connected to the same base station during the entire simulation. We assume there is sufficient space between all devices and other objects in the city in order to avoid collisions and that all devices are in outdoor locations. The small cells do not overlap coverage signals, and all the authenticated terrestrial users are always connected to the closest small cell available.

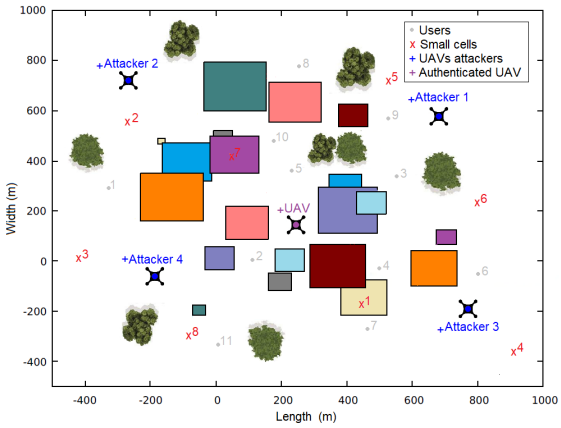


Fig. 1: Scenario Top View.

The authenticated UAVs connect to the small cells and generate downlink signals while the attackers attempt to jam the link. The jamming UAVs are able to adjust their power and position throughout the simulation. They aim to disrupt the UAV signal using decentralized capabilities and the least resources possible. The attackers utilize the same propagation models as the authenticated UAVs. They move towards the target UAVs when they are set up with moving capabilities.

TABLE I: Network Parameters.

Scenario Parameters	Value
Terrestrial Users	$0, 3, 5, 10, 20$
Authenticated UAVs	1
Small Cells	10
Small cell height	10 m
Attackers	$0, 1, 2, 3, 4$
Speeds	10 m/s
Modulation scheme	OFDM
Small cell power	4 dBm
Authenticated UAV power	2 dBm
Attackers power	0,2,10,20 dBm
Authenticated UAV position	random
Attackers position	random
Small cells position	random
Scenario	UMi
Distance	$100, 200, 500$ m
Simulation time	20 s

We define an urban scenario for our experiment based on the complex interference and obstruction patterns that we find in a city. Additionally, this is the most common environment for the UAV emergency delivery use case. The channel between the UAVs and the small cells use a wireless fading model which is modeled after A2G channels, and the transmission uses the OFDM modulation scheme. The deep network takes the fading and interference of wireless data into consideration using s subchannels for a total of Ni time slots, where

$s \leq Ni$ (in practice, we typically have $s \lll Ni$). The urban scenarios limit the small cell's height. We use the heights and distances in Table I and we create scenarios with both LoS and Non-Line-of-Sight (NLoS) conditions. Note that the distance in Table I is the distance between the small cell and the authenticated UAV.

A. Channel Model

The 3GPP standards [8] and [9] describe the losses and fading in 5G UAV wireless communications. Specifically, small-scale and large-scale fading in rural and urban scenarios. The UAV norm adds the logarithm's height component losses to the overall calculation to differentiate the UAV links from well-known wireless connections. Regarding the mathematical modeling of the small-scale fading effect, known as fast fading, there are two channel models available in the standard: the Tapped Delay Line (TDL) and the Clustered Delay Line (CDL). The second model comes from the first one. UAV fast fading models are frequently described as CDL channels. In addition, there is another subcategory for LOS and NLoS conditions in the model highlighted by the letters "ABCDE" after the model's name. For example, the CDL-D includes line-of-sight components while "ABC" represent models with NLoS components. Due to the line-of-sight characteristics of the UAV links, they are usually modeled using CDL-D. The major difference compared to terrestrial wireless links is that the UAV is at substantially higher altitudes considering the average rooftop height in a city, whereas the antenna is at positions below the same reference points, which means that the angular spreads in the departure and arrival devices swap.

B. The Dataset

As an extension of our previous work [10], we study deep network identification and generalization algorithms for jamming attacks under fading and interference when UAV attackers have static and moving configurations. We use sequential datasets, such as the time-series network parameters that the authenticated UAV generates during its mission. Specifically, we analyse two observable parameters: the Received Signal Strength Indicator (RSSI) and the Signal to Interference plus Noise Ratio (SINR) as inputs. Both parameters are collected from the authenticated UAV's receiver side. The dataset contains 2400 folders. Each folder has two files, one for RSSI data and one for SINR. The folders are classified into four configuration groups namely: $None\ Speed$, $Attacker\ Speed$, $User\ Speed$, and $Both\ Speed$. The $None\ Speed$ group collects RSSI and SINR data when there are no changes in the initial position of the elements in the network over time. In the $Attacker\ speed$ configuration, the attackers are able to change their speed according to Table I. In the third case named $User\ speed$, only the users are able to move in the simulation over time, and in the $Both\ speed$ case, both elements (the attackers and the users) are able to move according to predefined speeds during the simulation time. The RSSI parameter defines the total interference power in the network, and the SINR parameter measures the link quality (with the ratio of useful signal power over interference plus noise power). Both parameters are available in the authenticated UAV after the initial access synchronization.

For classification purposes, the dataset use the following nomenclature: " Yes Jamming", "No Jamming", "Moving Jamming", and "Fixed Jamming". Yes Jamming implies that at least one attacker has been discovered by the authenticated UAV in the network. No Jamming indicates the absence of jamming and suggests that the network is secure. Moving Jamming denotes that the jammer is approaching the authenticated UAV over time and Fixed Jamming suggests that the jammer is in a fixed position over time. The jammers change their transmission power values during the experiment.

C. The Designed Deep Network

In the this section, we describe the deep network characteristics that recognize attacks in realistic scenarios. The motivation behind the use of a deep network solution is to learn the characteristics of networks while it is under attack. The two headed DNN solution receives two sequences from the observable signals RSSI and SINR and then it produces just a single classified output. The architectural design contains the following in both of the two heads: (i) three CNNs layers, (ii) a LSTM or Multi-headed-attention layer, and (iii) a drop-layer. The body of the deep network consists of: (i) three convolutional layers, (ii) a Drop out layer, (iii) a fully connected layer, and (iv) the output layer for two classes classification as in figure 2. After the first classification, we run another deep network with the same structure to classify the moving and non-moving jammers.

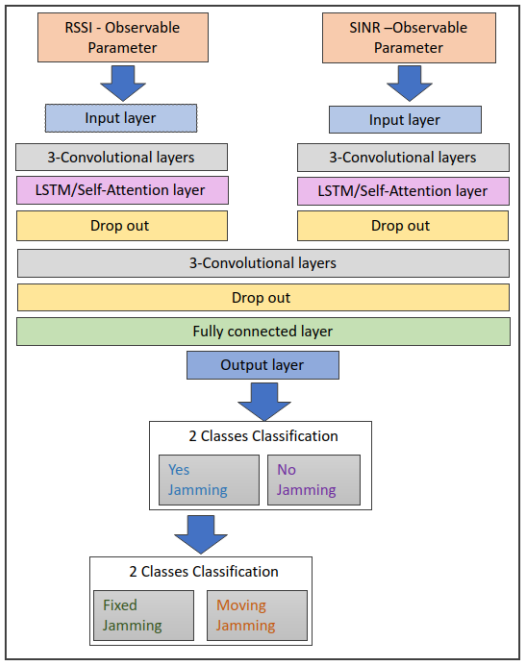


Fig. 2: Multiheaded Deep Network proposed architecture with two configurations: all structure with LSTM layer or all structure with self-attention layer

First, we use the auto correlation function to find the window size hyperparameter for the time-series data. For the other hyperparameters such as the number of CNN filters, kernel sizes, stride lengths, batch size, learning rate, and the number of regularization terms we use the grid search algorithm and other insights from our previous paper [10]. Our training data is fed into the Adam optimizer, which reduces the classification error for each new batch of commands. The Attention layer is applicable in this context because it provides the ability to capture temporal information since the

nodes in the layers are weighted by the sum of the row vectors that hold the information over several time steps, which increases robustness similar to the LSTM layer, but with fewer trainable parameters. At the end, we trained the deep network with 5-fold cross validation technique to assure correctness and prevent overfitting. The deep Network model is trained and tested on a computer system with a Nvidia RTX 3090 that has a 25GB RAM Graphics Processing Unit (GPU). All the Convolutional layers, self-attention, and drop out groups follow the same structure and parameters mentioned in Table II. The main deep network parameters are available in Table II.

TABLE II: Deep Network Configuration Parameters.

Deep network Parameters	Value
Base learning rate	$2.5\text{x}10^{-2}$
Base batch size	32
Conv-1 filters, kernel size, strides	8, 8, 2
Conv-2 filters, kernel size, strides	8, 4, 2
Conv-3 filters, kernel size, strides	8, 3, 1
Self-Attention head-number, key-dimensions	8, 8
(or LSTM)	50
Drop-out	0.4
Fully connected layer	100
Softmax	2

III. Experimental Results

In this section, we present the results of our synthetic UAV attack dataset executed in our designed deep network. Except when explicitly mentioned, all the network and deep network parameters used are described in Table I and in Table II, respectively. For each attacker number, we ran a simulation based on the attacker power, distance, and users amount, which generated 4800 files (2400 for RSSI and 2400 for SINR). We fed this data into the deep network and we analysed the classification results. First, we calculated the overall accuracy considering all the scenarios. Our deep network was able to correctly classify approximately 84% of the scenarios regarding $Yes\ Jamming$ and $No\ Jamming$ labels in the training and 74% in the test.

The training results showed that the deep network mis-categorized 11,407 training samples from $No\ Jamming$ to $Yes\ Jamming$ and vice-versa out of a total of 72288 training samples generated from all folds cross validation. For validation, we used roughly 14,000 samples. During the testing, we observed 80,537 misclassifications out of a total of 315,800. The relatively high misclassification number found in the training can be justified by taking into account the abrupt changes in the stochastic channel model and the random nature of the simulation. Moreover, the fact that we did not use samples from the same configuration in the test as the ones that we used in the training might justify the increased number of miscategorized samples. Figure 3 presents the confusion matrix for all scenarios.

Tables III and IV illustrate additional information regarding the experiment's accuracy and f-score parameters for training and testing, respectively.

In order to simplify the 2-steps-classification in the deep network architecture in figure 2, we tried a 1-step classification with 3-classes, the labels were $No\ Jamming$, $Fixed\ jamming$, and $Moving\ Jamming$, but the accuracy results decreased approximately 11% reaching 72%. In the binary classification test, it became clear that the CNN layers were critical to reduce the number of trainable parameters. After replacing the LSTM with the self-attention layers, we noticed that the number of trainable parameters reduced down to approximately half of the initial amount (i.e., from 43000 to 22000), but maintained the same good performance in training and validation. We observed that accuracy increased roughly by 2% (i.e., 73.15 to 75.13) during testing using the attention layers.

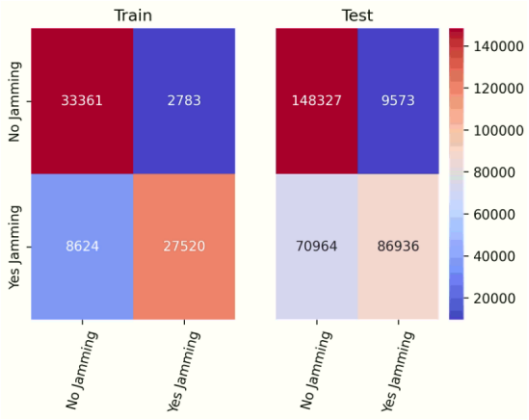


Fig. 3: The binary classification Confusion Matrix for all scenarios.

TABLE III: Precision, recall, and f1-score in training.

	precision	recall	f1-score	support
No Jamming	0.79	0.92	0.85	36144
Yes Jamming	0.91	0.77	0.83	36144

TABLE IV: Precision, recall, and f1-score in testing.

	precision	recall	f1-score	support
No Jamming	0.68	0.94	0.79	157900
Yes Jamming	0.90	0.55	0.68	157900

Figure 4 depicts the $accuracy$ results based on the number of terrestrial users connected to the network for training and testing. The overall $accuracy$ decreased according to the number of users connected to the network. For example, when there were no users in the network, training $accuracy$ was about 90%, but with 20 users it was around 83%. The $accuracy$ decreased because of the interference generated by the connection between the users and the small cells over time. The slow fading values changed when the users were configured to move and the additional users made it hard to differentiate whether the RSSI and SINR changes were caused by attackers or users. The accuracy of the 5-users simulation was lower (roughly 75%) compared to the other cases with fewer and/or more users because the related data was new to the deep network. These results assured the robustness of our deep network with respect to data that was not in the training.

Figure 5 shows the accuracy over distance and attackers power ratios during training. Attackers with lower power are harder to be identified by the deep network. Both simulations where the attacker power was configured to 2 dBm and the

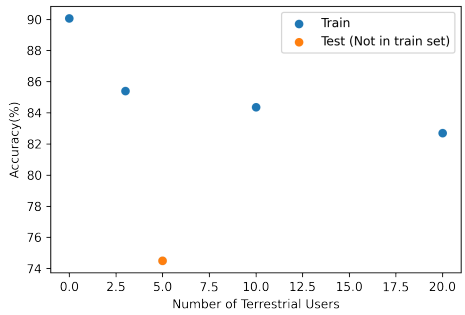


Fig. 4: Accuracy versus the number of users in the network in the 5-fold Cross validation training and test.

distance was set up to 200 m were removed from the training.

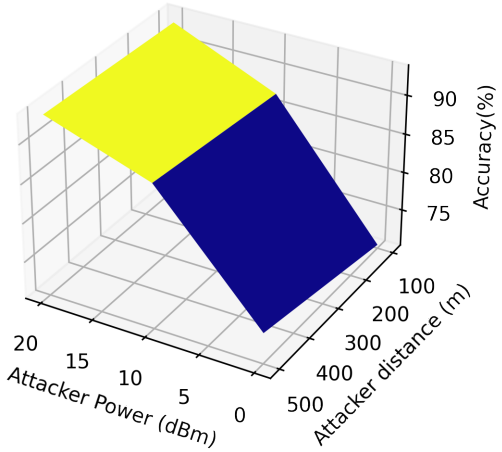


Fig. 5: Accuracy over power and distances.

Table V presents the overall results of each scenario considering all distances, powers, and attackers. The deep network achieved the highest accuracy in the Attacker speed scenario (when only the attackers move toward the authenticated UAV), but the accuracy difference compared to the other scenarios was small (i.e., 0.25% compared to None speed, 1.15% in User Speed, and 3.24 % in Both speed).

TABLE V: Accuracy in fixed and moving scenarios.

Scenario	Accuracy (%)
None speed	77.35
Attacker speed	77.60
User speed	76.45
Both speed	74.36

Figure 6 depicts the accuracy across the number of attackers and their respective power. It is difficult for the deep network to identify a small number of attackers or an attacker with limited power because The CDL channel model can fluctuate 30 dB depending on the configurations in the fast fading parameters.

IV. Conclusion

This article offered a solution based on deep networks for identifying jamming attacks in UAVs networks. We were able to embed the deep network with self-attention layer in the UAV because after training and testing the processing capacity of the generalized deep network matched the limited processing capacity of the UAV. In general, our deep network was able to

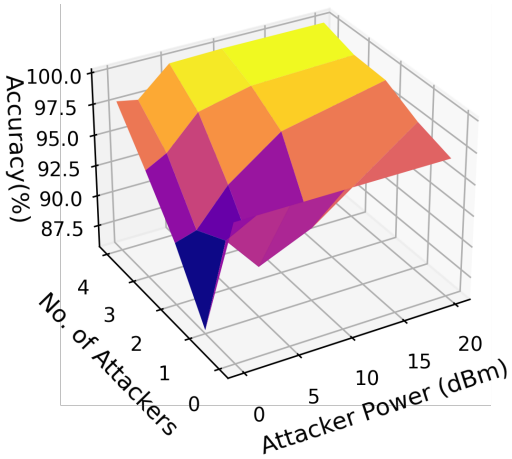


Fig. 6: Accuracy over the number of attackers in the network.

recognize attacks in all scenarios' configurations. Simulations with 3 or more attackers, fewer users, and power greater than 10 were easier to be identified. Furthermore, the 3D distance between the small cell and the authenticated UAV impacted the identification accuracy. In our case, as the distance grew, the chances of identification increased because the interference decreased.

Acknowledgment

This research received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Project Number 813391 and ANEMONE (PID2021-126431OB-I00) project by the Spanish Government.

References

[1] M. Mahdi Azari, Fernando Rosas, and Sofie Pollin. "Cellular Connectivity for UAVs: Network Modeling, Performance Analysis, and Design Guidelines". In: *IEEE Transactions on Wireless Communications* 18.7 (2019), pp. 3366–3381. DOI: 10.1109/TWC.2019.2910112.

[2] Yann Lecun and Yoshua Bengio. "Convolutional Networks for Images, Speech and Time Series". In: *The Handbook of Brain Theory and Neural Networks*. Ed. by Michael A. Arbib. The MIT Press, 1995, pp. 255–258.

[3] Bendong Zhao et al. "Convolutional neural networks for time series classification". In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169.

[4] Hassan Ismail Fawaz et al. "Deep learning for time series classification: a review". en. In: *Data Min. Knowl. Discov.* 33.4 (July 2019), pp. 917–963.

[5] Ashish Vaswani et al. "Attention is all you need". In: *Advances in Neural Information Processing Systems*. 2017, pp. 5998–6008.

[6] Chun-Jie Chiu Kai-Jui Chen An-Hung Hsiao and Kai-Ten Feng. "Self-Attention based Semi-Supervised Learning for Time-varying Wi-Fi CSI-based Adjoining Room Presence Detection". In: *IEEE 95th Vehicular Technology Conference:, VTC Spring 2022, Helsinki, Finland, 19-22 June, 2022* (2022).

[7] Yuchen Li et al. "Jamming Detection and Classification in OFDM-Based UAVs via Feature- and Spectrogram-Tailored Machine Learning". In: *IEEE Access* 10 (2022), pp. 16859–16870. DOI: 10.1109/ACCESS.2022.3150020.

[8] *3GPP - Technical Specification Group Radio Access Network; Study on Enhanced LTE Support for Aerial Vehicles*. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3231.

[9] *3GPP - Technical Specification Group Radio Access Network; Study on channel model for frequencies from 0.5 to 100 GHz*. URL: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3173.

[10] Viana J et al. "Two methods for Jamming Identification in UAVs Networks using New Synthetic Dataset". In: *(IWAANETS 2022) in the 2022 IEEE 95th Vehicular Technology Conference* (2022). URL: https://arxiv.org/pdf/2203.11373.pdf.

2.5. Article #5: Two Methods for Jamming Identification in UAV Networks Using a New Synthetic Dataset

This paper presents two complementary approaches to detect jamming attacks in UAV networks: a statistical method using STL and a CNN-LSTM architecture. The research addresses the critical vulnerability of UAV transmissions to jamming attacks due to the open nature of air-to-ground (A2G) wireless communication networks.

Key Contributions

The key contributions of this work include:

- Development of a new synthetic dataset that incorporates realistic channel effects;
- Statistical model based on time series analysis for jamming detection;
- Simplified CNN-LSTM architecture requiring minimal computational resources;
- Comprehensive performance evaluation across various scenarios.

Statistical Approach

The statistical approach uses STL to decompose the received signals into trend, seasonal and residual components. This method achieved 84.38% accuracy in identifying attacks when the attacker was 30 meters from the UAV. Although its effectiveness decreased with distance and lower jamming power ratios, the low computational requirements of the method make it suitable for real-time UAV applications.

Deep Learning Approach

The CNN-LSTM solution demonstrated superior performance, achieving 99.99% accuracy for jamming powers greater than 2 dBm and distances less than 200 meters. The key features of the architecture include the following:

- Three convolutional layers with optimized filter configurations;
- An LSTM layer for temporal pattern recognition;
- An efficient design with only 53k trainable parameters.

Experimental Validation

The experimental validation considered various factors, including:

- Jamming power ratios (1–20);
- Distances between UAV and base station (10–350 meters);
- Channel conditions and shadowing effects.

Significance

The combined approach offers a practical solution for UAV networks. The statistical model provides rapid detection with minimal resources, while the deep learning model offers higher accuracy when computational resources are available. This research demonstrates the feasibility of implementing effective jamming detection in resource-constrained UAV environments.

Article Details

- **Title:** Two Methods for Jamming Identification in UAV Networks Using a New Synthetic Dataset
- **Authors:** Joseanne Viana, Hamed Farkhari, Luis Miguel Campos, Pedro Sebastião, Francisco Cercas, Luis Bernardo, Rui Dinis
- **Status:** Accepted at a Conference
- **Conference:** IEEE VTC Spring 2022
- **DOI:** 10.1109/VTC2022-Spring54318.2022.9860816

The significance of this work lies in its practical approach to UAV security, providing lightweight and sophisticated solutions that can be implemented based on available resources. The comprehensive evaluation across different scenarios and conditions establishes a strong foundation for future research in UAV network security.

Two methods for Jamming Identification in UAV Networks using New Synthetic Dataset

Joseanne Viana [†‡], Hamed Farkhari*[†], Luis Miguel Campos *, Pedro Sebastião [†‡], Francisco Cercas [†‡], Luis Bernardo [§‡], Rui Dinis[§‡],
[†]ISCTE – Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
*PDMFC, Rua Fradesso da Silveira, n. 4, Piso 1B, 1300-609, Lisboa, Portugal
[‡]IT – Instituto de Telecomunicações, Av. Rovisco Pais, 1, Torre Norte, Piso 10, 1049-001 Lisboa, Portugal
[§]FCT – Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal;
Emails : joseanne_cristina_viana@iscte-iul.pt, Hamed_Farkhari@iscte-iul.pt, luis.campos@pdmfc.com, pedro.sebastiao@iscte-iul.pt, francisco.cercas@iscte-iul.pt, lflb@fct.unl.pt, rdinis@fct.unl.pt

2022 IEEE 95th Vehicular Technology Conference (VTC2022-Spring) | 978-1-6654-8243-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/VTC2022-Spring54318.2022.9860816

***Abstract*—Unmanned aerial vehicle (UAV) systems are vulnerable to jamming from self-interested users who utilize radio devices to disrupt UAV transmissions. The vulnerability occurs due to the open nature of air-to-ground (A2G) wireless communication networks, which may enable network-wide attacks. This paper presents two strategies to identify Jammers in UAV networks. The first strategy is based on a time series approach for anomaly detection where the available signal in the resource block is decomposed statistically to find trends, seasonality, and residues. The second is based on newly designed deep networks. The combined techniques are suitable for UAVs because the statistical model does not require heavy computation processing, but is limited to generalizing possible attacks that might occur. On the other hand, the designed deep network can classify attacks accurately, but requires more resources. The simulation considers the location and power of the jamming attacks and the UAV position related to the base station. The statistical method technique made it feasible to identify 84.38% of attacks when the attacker was at a distance of 30 m from the UAV. Furthermore, the Deep network's accuracy was approximately 99.99 % for jamming powers greater than two and jammer distances less than 200 meters.**

***Index Terms*—Cybersecurity, Convolutional Neural Networks (CNNs), Deep Learning, Jamming Detection, Jamming Identification, UAV, Unmanned Aerial Vehicles, 4G, 5G;**

I. Introduction

When it comes to the 5G communication system, the deployment of unmanned aerial vehicles (UAVs) is a game-changer. They allow faster and more flexible network services in the sky at higher data rates because they have complete control over their movement and a high probability of establishing robust line-of-sight (LoS) communication links[1], [2] and [3]. Yet, UAV transmission is vulnerable to attacks and interference because of the open nature of air-to-ground (A2G) wireless communication links and the A2G channel connections that may present opportunities for attacks in the network. As a result, it is critical and vital to identify threats and protect UAV communications[4]. In wireless communications, encryption and encoding techniques are commonly employed to ensure security by preventing unwanted access and intentional interference. Nevertheless, maintaining encryption systems requires a lot of effort and resources [5]. Consequently, encrypted UAVs' communication may not be feasible. Therefore, attack identification mechanisms become fundamental in UAV networks. Commonly used jamming detection algorithms such as packet delivery ratio and received signal strength with a missed detection rate are presented by [6]. Statistical models have lately been recognised as feasible methods for monitoring network activity in wireless communications and detecting suspicious attacks via the use of wireless channel properties rather than encryption keys. In [7], the authors propose a jamming detection method using a Naive Bayes classifier trained on a limited sample of data that considers only transmission noise effects in wireless scenarios. Cheng et al. [8] describes a Bayesian method for jamming detection. In [9], the authors offer a jamming detection strategy for GNSS-based train localization that makes use of singular value decomposition (SVD). Lu et al. [10] present a technique for detecting jamming in power networks that is both efficient and resilient. Most of the studies' computations did not take channel impacts into account. Considering machine learning and Deep Networks, Youness et al. [11] analyze the signal properties that may be used to detect jamming signals, and create a dataset based on these parameters. They utilize the random forest method, the support vector machine algorithm (SVM), and a neural network algorithm to classify the features extracted by the jamming signal. Li et al. [12] also identify jamming samples using signal-extracted features, but the author adds another way to detect attacks that utilizes 2D samples and pretrained networks (i.e. AlexNet, VGG-16, ResNet-50). Although, with pre-trained networks, we may utilize transfer learning to adapt the network to a new dataset without having to design it from scratch. Certain pre-trained networks are enormous and need significant computational processing in order to categorize information which may be unsuitable for UAVs. While embedded statistical models and deep network techniques in the cloud or at the edge can monitor and analyze channel degradation caused by jamming attacks, there is a lack of publicly available research on attack detection in UAV communications. Rather than identifying attacks, most research in this field focuses on prevention, namely anti-jamming measures and non-traditional ways to avoid jamming.

We aim to demonstrate that it is feasible to identify attacks in the receiver block of the UAV by combining a Seasonal Trend Decomposition (STL) time series analyzer with a unique deep network architecture that has much fewer layers than the well-known pretrained networks and does not rely on transfer learning techniques.

The remainder of the paper is structured as follows: First, the detailed system model is presented in Section II. It describes the dataset used, the statistical approach, and the suggested deep network architecture. The assessment of both approaches is summarized in Section III. Finally, section IV

concludes this paper.

Notations: Scalar variables are denoted by lower-case letters (a, b,...), vectors are denoted by boldface lower-case letters (**a, c,...**), and matrices are denoted by boldface capitals (**A, B,...**). Lower case letters denote time-domain variables, whereas upper case letters indicate frequency-domain variables.

A. Contributions and Motivation

With regard to jamming detection and the associated challenges utilizing deep networks and statistical methods, there is a lack of public research and accessible data. Taking this into consideration, the following highlights some of the contributions made by this paper:

- A general case study model that takes into account the jamming power and distances between the jamming attacker and the base station in relation to the authenticated UAV that uses average received signal power in the resource block, Signal-To-Noise-Ratio (SNR), average-noise, average-transmitted-power, path-loss, and shadowing.
- A statistical model for jamming detection using data from the UAV's reception resource block.
- A simpler Convolutional Neural Network-Long Short Term Memory (CNN-LSTM) architecture for jamming detection.
- Simulation results for both of the presented techniques.
- A comparison of two jamming detection technique performances in terms of accuracy over attacker distance and power.

Additionally, we offer a table representation of the confusion matrix for both the training and test data sets. We devise a strategy that increases performance while using the fewest CNN layers.

II. System Model

We analyze a UAV jamming scenario in which a communication link exists between the base station and the UAV, referred to as an air-to-ground (A2G) connection and there are jamming attackers in unknown locations on the ground or in the air that can deliberately jam the signal received by the authenticated UAV. Although, we use the Single Carrier (SC) transmission scheme, the jamming detection algorithms are applicable to any transmission technique. We investigate the reception power in the authenticated UAV using two distinct approaches: one of which relies on time series statistical models and the other on deep networks. Each component is explained as follows:

A. dataset

The data set simulates the received signal in the UAV resource block considering slow fading effects in the transmission channel, specifically (pathloss and shadowing). The frequency domain channel $H_{(k,d)}$ is represented as in 1,

$$H_{(k,d)} = \frac{\sum_{i=1}^{N_{rays}} \alpha_i(\tau) + \exp(-j2f\pi\tau)}{\sqrt{(PLS)}} \quad (1)$$

For $H_{(k,d)}$, f is the frequency band, α is the attenuation of the multipath ray, and τ is the propagation delay. We adopt the Rician model to describe the multipath rays. The path loss is estimated using the UAV and base station locations $p_{id,t} = [x_{uav,t}, y_{uav,t}, z_{uav,t}]^T$, $[x_b,\ y_b,\ z_b]^T$ (in meters) and the 3D Euclidean distance equation $||p_{bs} - p_{uav}||^2$ respectively. The reference point is at $d = 10m$ and S the shadowing is a random variable modeled as $S|_{db} \sim \mathcal{N}(0,\ \sigma^2)$. The received power $Y_{(k,d)}$ is calculated using 2,

$$Y_{(k,d)} = H_{(k,d)}X_k + N, \quad (2)$$

X_k is the frequency domain representation of the transmitted signal x_k^R, and $N \sim \mathcal{N}(\mu,\ \sigma_k^2)$ is the noise in the channel, while k is an available frequency in the bandwidth.

The jamming signal takes into account the same properties of the reliable signal considering path loss. In the experiment, the jammer focuses on $F_k \ll F$ frequencies inside the bandwidth B available for transmission with the gain $(P/P_J)I$, where I is the percentage of the slot occupied by the jammer. The jammer is more powerful than the signal in the majority of the dataset samples (i.e. $P_J > P_S$). The formula in 3 shows how the total noise is affected by the jamming power received, where N_k is the noise without interference.

$$E[|N_{k,Tot}|^2] = \frac{F}{F_{kJ}}\frac{P_j}{P_S}E[|N_k|^2]. \quad (3)$$

The dataset contains 483,540 transmission blocks or samples Sa, with N steps each $\{Sa_k; k = 0, 1, ..., N-1\}$ where N is the FFT size in the frequency domain. The received signal is then classified according to the following categories: Good-Normal, Bad-Normal, Good-Jamming, and Bad-Jamming. "Normal" defines a non-jamming signal while good and bad channels are distinguished by $SNR = 20$ and $SNR = 1$, respectively. Additionally, we vary the jammer and base station positions as well as the jammer power in the experiment. Fig 1 depicts two jamming samples available in the dataset. The top illustrates a jamming sample in a good channel. The bottom shows a jamming signal in a bad channel. The dataset contains only the received power categorized in the four classes previously mentioned.

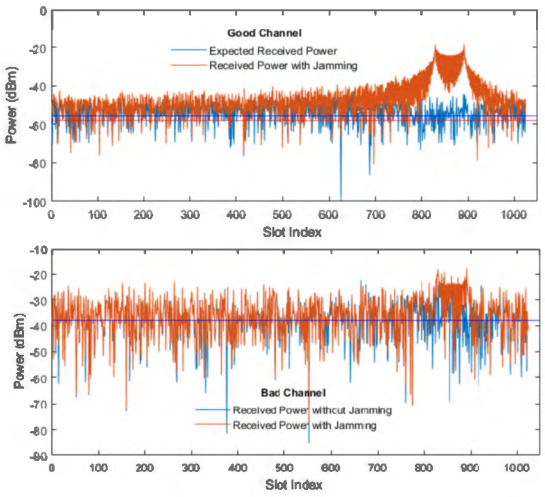


Fig. 1: Received signal with jamming in Good and Bad Channels.

Table I compares the mean received power differences at the UAV when the signal is jammed and when the signal is not jammed with the same power as the authenticated signal at a distance of 30m from the UAV. The base station is about

the same distance from the UAV. According to the table, the jammer modifies the mean power in the resource block, depending on the jammer's power and position relative to the base station and the authorized UAV.

TABLE I: Mean Difference Between Channels with and Without Jamming

Mean Power difference (dBm)	No Jamming	Jamming
Good Channel	0	4.189
Bad Channel	0	0.592

B. *Statistical methods*

The statistical model chosen was STL [13]. It takes into account the decomposition of the signal into trends, seasonal, and residuals. Figs 2 and 3 illustrate a representative sample of both jammed and unjammed decomposed signals. The top chart in both figures shows a combination of three samples from the dataset in a sequence. In fig 2, the jamming power of the attacker is five times greater than the signal received from the base station. The jamming location is $30m$ away from the UAV, while the UAV placement is $90m$ away from the base station. In fig 3, there is no jamming attacker and the base station location is identical to that in fig 2.

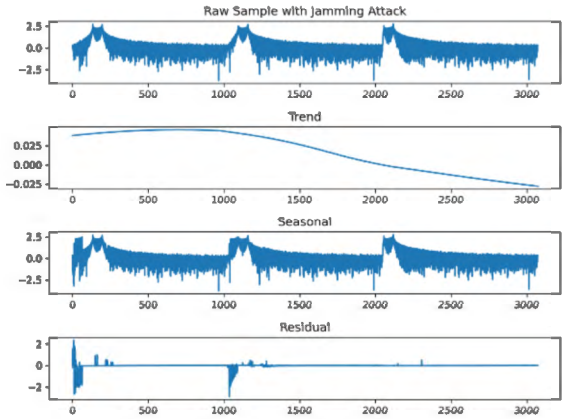


Fig. 2: STL decomposition of a normalized sample in the present of jamming attacker.

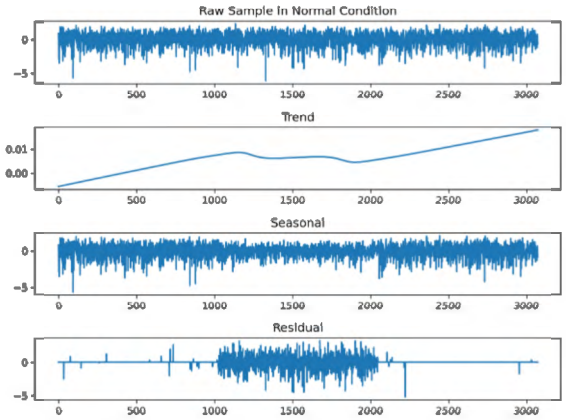


Fig. 3: STL decomposition of a normalized sample without jamming signal.

STL is an acronym for "Seasonal and Trend decomposition based on locally weighted regression (Loess)". This is a technique for decomposing time series data into trend, seasonal, and residual components. After removing the current trend estimate, the seasonal component of the cyclic sub-series is calculated using seasonal smoothing. Next, the predicted seasonal component is smoothed using lowpass smoothing. Finally, the deseasonalized series is smoothed once more with trend smoothing in order to provide an estimate of the trend component. This procedure is repeated numerous times in order to improve the component estimates' accuracy[13].

In the experiment, each sample contains elements that degrade communication, such as noise and slow fading components. The channel and the noise effects are independent of each other. We assume that the jamming effect will manifest itself in multiple samples in the case of a jamming attacker. Then, we exploit the jamming attack's periodicity to identify it using STL. We assume the jamming attack is applied in a certain narrow channel bandwidth from 8% to 10% . After the third resource block reception, first, we combine three samples in a sequence. Second, we apply the STL decomposition and then reconstruct the signal again using 4. Finally, we calculate the sample's error as in 5. If there is a pattern in the sample, the number of errors is smaller; consequently, the sample is classified as a jamming signal. In a normal situation without a jamming effect, there is no specific repeated pattern, and we see more errors in the STL decomposition reconstructed signal. In order to classify the signals correctly, we use root mean square error (RMSE) for binary classification to determine the presence or absence of jamming effects in the experiment as in 6. Lastly, we apply Support Vector Machine (SVM), Logistic Regression, and Random Forest algorithms to split the features in the classes. The difference between the dataset described in the previous section and the one used in the STL is the concatenation of three resource blocks in one sample. While developing the experiment, we noticed it is fundamental to accurately define the *period* in order to get good results from STL.

$$S_a = T + S + R \tag{4}$$

$$Error = S_a - S_r \tag{5}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N} Error_i^2} \tag{6}$$

In 4, 5 and 6, S_a and S_r are the original and reconstructed samples, and T, S, and R stand for trend, seasonal, and residual components, respectively. N is the length of a sample and we use $RMSE$ as a feature for binary classification.

C. *Convolutional Neural Networks*

In the experiment using the convolutional neural network (CNN) joined with Long short-term memory (LSTM), we developed an architecture capable of achieving 99 % accuracy with a small number of CNN layers, number of filters, and kernel sizes in the convolutional layers. Our CNN architecture uses three convolutional layers: one LSTM, one drop-out, a fully connected layer, and the output layer for classification as Fig 4 illustrates. Max-pooling is a common layer used in

CNNs, but it might result in the loss of critical information in certain topologies. According to Geofrey Hinton "pooling is a mistake" and for these reasons, we replace the pooling layer with strides in our convolutional layers. Authors in [14] and [15] reported that using very small weight decay (L2 regularization) values such as $5x10^{-4}$, and $4x10^{-5}$ in convolutional layers are critical for performance purposes and they should be precisely chosen. After executing the grid search algorithm, we found the an optimal L2 regularization and used it for all CNN layers in the experiment.

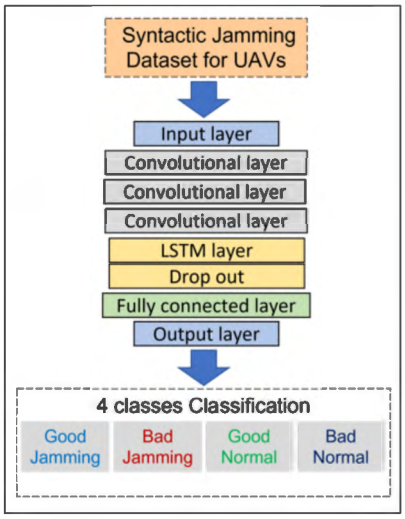


Fig. 4: Proposed CNN-LSTM Architecture.

For the deep network, we use a single sample dataset. Initially, we partition the dataset into 70% for training and 30% for testing. In the first phase, we divide the training section into two sub-sections: training and validation. We then apply the grid search algorithm to determine the deep network's hyperparameters. The hyperparameters are as follows: the number of CNN filters, the kernel sizes, the strides, the batch size, the learning rate, the number of regularization terms, and the drop out percentage. In the second phase, we employ a 5-fold cross validation procedure during the training phase to ensure accuracy and avoid overfitting.

III. Experimental Results

The results of the suggested algorithms are detailed below. First, we look at the statistical data for jamming detection. Then we'll look into deep networks for classification, loss, and accuracy in training and testing. The CNN model is trained and tested in a system with a Nvidia RTX 3090 GPU. The jamming attacker signal power ratio ranges between one and twenty. The distances between the UAV and the base station, and the UAV and jamming attacker varies between 10 and 350 meters. The shadowing variance was adjusted to 4 [16].

A. Statistical model

Fig 5 depicts the RMSE between original signal and reconstructed one after STL decomposition in the BoxPlot. The diagram shows that both distributions can be split using binary classification. We merge three resource block into a sequence where each of them is the size of $N = 1024$ in length, and we use N as a period parameter for the STL decomposition algorithm. After calculating the reconstructed signal, we use RMSE as a feature to detect the jamming attack with respect to distance and power. The overall accuracy of this method for all scenarios using the three different classifiers is about 70%, as is shown in Fig 6, and varies according to the jamming power ratio and distance of the UAV from the base station and the attacker. In some cases, depending on the jammer and base station location relative to the UAV, the accuracy can increase up to 84.38% by employing an SVM classifier that outperforms the other two classifiers.

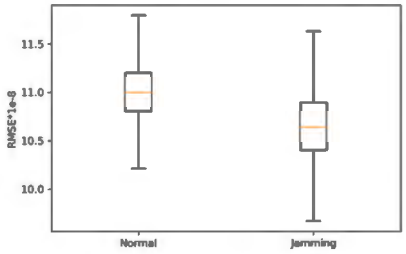


Fig. 5: Boxplot of RMSE between original signal and reconstructed signal for two classes.

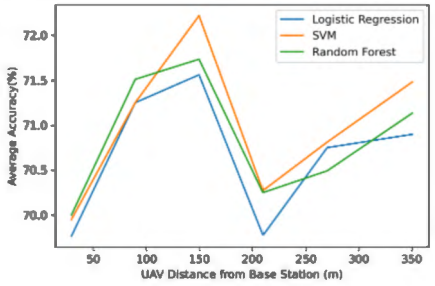


Fig. 6: Overall statistical average accuracy for different attacker powers and distances per fixed UAV distance from base station.

Fig 7 (a) and (b) illustrate the *accuracies* of the STL model at various attacker distances and power ratios using the SVM classifier. In (a), the accuracy decreases with increasing jammer distance, i.e. when the jammer is 350m away, the accuracy in the statistical model is reduced. When the jamming power P_j decreases as specified in (b), it is difficult for the algorithm to differentiate low-power jammers and prominent channel effects such as fading, path loss, and shadowing. Due to the statistical model's low computational requirements, it may readily be implemented in UAVs for user packet transmission and Command and Control (C2) links.

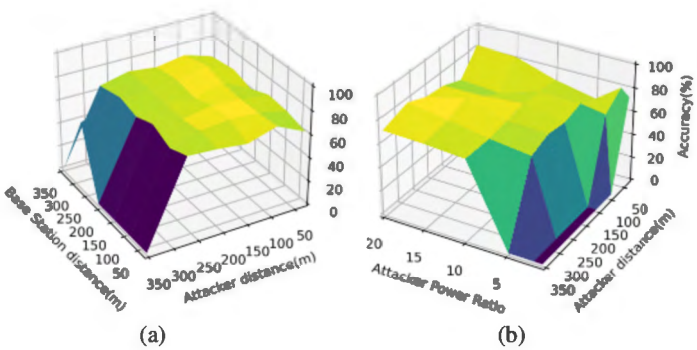


Fig. 7: (a) Accuracy for $P_j = 5P_s$. (b) Accuracy for $Bs_d = 350m$.

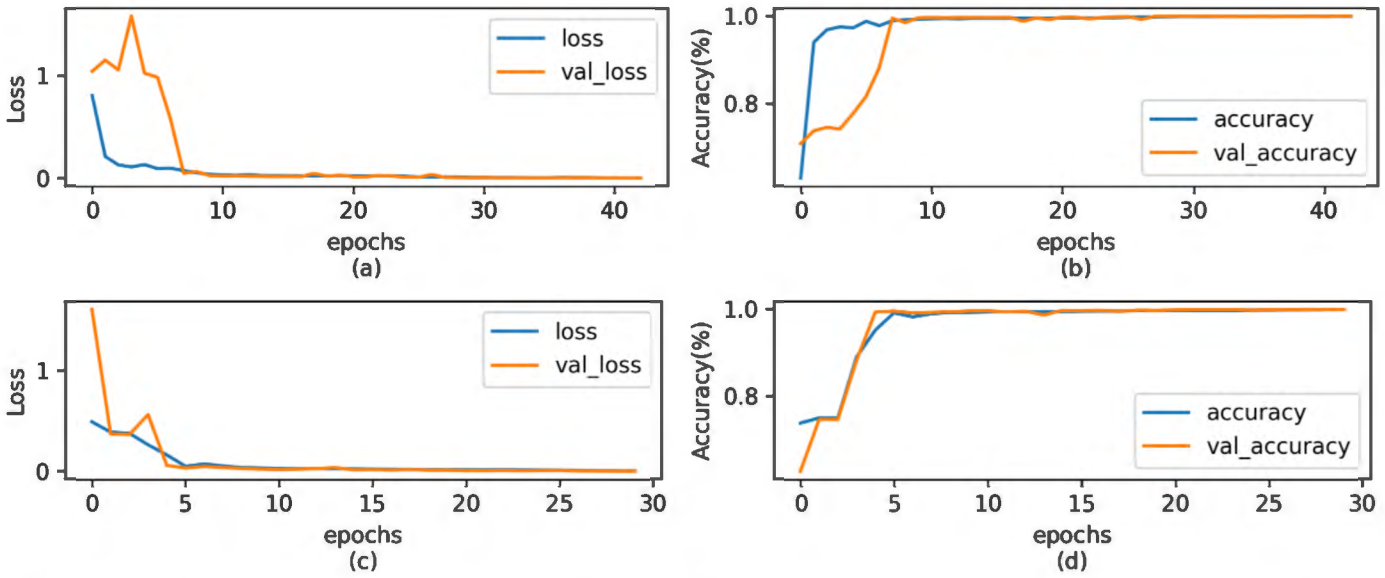


Fig. 8: Convergence of deep network during training with 4 steps. a) loss of model 1. b) *accuracy* of model 1. c) loss of model 2. d) *accuracy* of model 2.

B. Convolutional Neural Networks

In the CNN simulation, the slot size is set to N=1024. The only parameter the experiment uses for jamming detection is the signal received in the resource block. During the CNN performance configuration phase, we notice that adding layers is preferable to increasing the number of filters in each layer. Additionally, the regularization component of the CNN and the fully connected layer are crucial for achieving performance improvements since, in both cases, the parameters demand appropriate adjustment. The LSTM layer is added to take advantage of the sequence memory characteristics in order to increase robustness. Also, the filter numbers and the kernel sizes are implied in the overall trainable parameters. We achieve the same performance by employing two fully connected layers of 50 nodes each rather than a single layer of 100 nodes. These adjustments result in a decrease in the total number of trainable parameters from around 100k to 53k. Table II presents the hyperparameters of our deep network.

TABLE II: Deep Network Configuration Parameters.

Deep network Parameters	Value
base learning rate	$3.16\text{x}10^{-3}$
base batch size	32
conv-1 filters, kernel size, strides	4, 8, 4
conv-2 filters, kernel size, strides	4, 4, 2
conv-3 filters, kernel size, strides	4, 3, 1
LSTM	100
drop-out	0.4
dense	100
softmax	4

We use L2 regularization terms equal to $1\text{x}10^{-6}$ and $1\text{x}10^{-5}$ in the convolutional layers and in the fully connected layer for both kernels and biases, respectively. The initial batch size is 32 for the deep network. Then the grid search algorithm defines the learning rate as $3.16\text{x}10^{-3}$. After that, we increase the learning rate and batch size to 0.2, and 2048, respectively. The new batch size and learning rate increases GPU (RTX 3090 with 24GB Ram) use from 30% to 92% of the limit of processing capacity. Consequently, the batch size is limited to 2048. Following that, we train our deep network in the different steps for validation accuracy. At each training step, if the performance declines compared with the previous step, the training process is immediately stopped, the previous model weights are loaded, and the training process at that step is repeated with a new lower learning rate and batch size. These steps are used to achieve 80, 90, 95, and 99.99% validation accuracy, and as the training process progresses through each step, we save the model and continue the training process. By employing this strategy, we minimize the overshooting effect in the deep network and shorten the overall training time for five models in 5-fold cross validation. As an example, fig 8 shows the convergence of two models from 5 models in cross validation. It shows 99.99% *accuracy* for all the five models in 5-fold cross validation with a maximum of 40 epochs.

TABLE III: Confusion matrix of 4 classes classification for test data by CNN-LSTM network.

	Good Normal	Bad Normal	Good Jamming	Bad Jamming
Good-Normal	36281	0	0	0
Bad-Normal	0	36219	0	62
Good-Jamming	0	0	36281	0
Bad-Jamming	0	385	0	35896

Table III shows the confusion matrix the test set using the CNN-LSTM algorithm. We obtain the correct classification for all samples with good channels using the designed deep network and there is minimal misclassification in the case of bad channels. Specifically, we have 62 misclassifications in the absence of jamming and 385 when jamming is present which represents less than 1% of the total samples analyses. Table

TABLE IV: The result of test set for 4-Classes classification by proposed deep network.

	precision	recall	f1-score	support
Good-Normal	1.00	1.00	1.00	36281
Bad-Normal	0.99	1.00	0.99	36281
Good-Jamming	1.00	1.00	1.00	36281
Bad-Jamming	1.00	0.99	0.99	36281

IV provides more specific details of the precision and f-score parameters in the experiment.

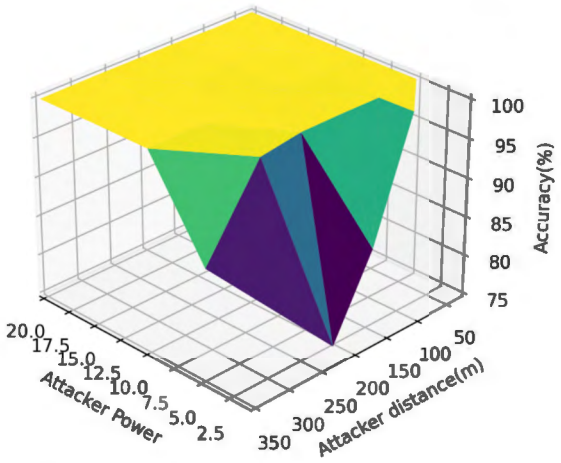


Fig. 9: CNN Accuracy for base station distance 30m.

One of the following setup techniques may be used to reduce the total number of trainable parameters. First, three CNN layers with eight filters each, followed by a LSTM and two fully connected layers with fifty nodes each. Alternatively, three CNN layers with four filters, followed by a LSTM and only one fully connected layer with one hundred nodes can be used. We discover that the second one converges more quickly and with fewer epochs.

Fig 9 depicts the CNN model's performance related to the *accuracy* over a range of attacker distances and power ratios. CNN may struggle to identify low-power jammers depending on the channel situation, the same as in the STL statistical model, but in all other circumstances, CNN obtains 99.99% correct classifications.

IV. Conclusion

This article offered a solution composed of two techniques for identifying jamming attacks in UAV networks. The first one is based on a time series method for detecting patterns using the STL decomposition technique. The second is based on convolutional neural networks. The signal analysed by both approaches relied on the resource blocks received by the UAV. Using the time series analysis, it was possible to identify 84.38% of the attacks when the $SINR$ of the jamming signal was high and the UAV was closer to the attacker than to the base station. While using the deep networks accuracy was 99.99% in the jamming cases and false alarms occurred in less than 1% of the cases. The combined method is appropriate for UAVs since the statistical model is restricted in its ability to classify all conceivable attacks. However, the deep network can classify all attacks, but requires additional resources.

Acknowledgment

This research received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Project Number 813391"

References

[1] Zeeshan Kaleem et al. "UAV-empowered disaster-resilient edge architecture for delay-sensitive communication". In: *arXiv* December (2018), pp. 124–132. DOI: 10.1109/MNET.2019.1800431.

[2] Syed Ahsan Raza Naqvi et al. "Drone-Aided Communication as a Key Enabler for 5G and Resilient Public Safety Networks". In: *IEEE Communications Magazine* 56.1 (2018), pp. 36–42. DOI: 10.1109/MCOM.2017.1700451.

[3] Fei Qi et al. "UAV Network and IoT in the Sky for Future Smart Cities". In: *IEEE Network* 33.2 (2019), pp. 96–101. DOI: 10.1109/MNET.2019.1800250.

[4] Hoon Lee et al. "UAV-Aided Secure Communications With Cooperative Jamming". In: *IEEE Transactions on Vehicular Technology* 67.10 (2018), pp. 9385–9392. DOI: 10.1109/TVT.2018.2853723.

[5] Liang Xiao et al. "User-Centric View of Unmanned Aerial Vehicle Transmission Against Smart Attacks". In: *IEEE Transactions on Vehicular Technology* 67.4 (2018), pp. 3420–3430. DOI: 10.1109/TVT.2017.2785414.

[6] Aleksi Marttinen, Alexander M. Wyglinski, and Riku Jäntti. "Statistics-Based Jamming Detection Algorithm for Jamming Attacks against Tactical MANETs". In: *2014 IEEE Military Communications Conference*. 2014, pp. 501–506. DOI: 10.1109/MILCOM.2014.90.

[7] Yuxin Shi et al. "Efficient Jamming Identification in Wireless Communication: Using Small Sample Data Driven Naive Bayes Classifier". In: *IEEE Wireless Communications Letters* 10.7 (2021), pp. 1375–1379. DOI: 10.1109/LWC.2021.3064843.

[8] Maggie Cheng, Yi Ling, and Wei Biao Wu. "Time Series Analysis for Jamming Attack Detection in Wireless Networks". In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017, pp. 1–7. DOI: 10.1109/GLOCOM.2017.8254000.

[9] Jian-Cong Li et al. "Jamming Identification for GNSS-based Train Localization based on Singular Value Decomposition". In: *2021 IEEE Intelligent Vehicles Symposium (IV)*. 2021, pp. 905–912. DOI: 10.1109/IV48863.2021.9575412.

[10] Zhuo Lu, Wenye Wang, and Cliff Wang. "Modeling, Evaluation and Detection of Jamming Attacks in Time-Critical Wireless Applications". In: *IEEE Transactions on Mobile Computing* 13.8 (2014), pp. 1746–1759. DOI: 10.1109/TMC.2013.146.

[11] Youness Arjoune et al. "A Novel Jamming Attacks Detection Approach Based on Machine Learning for Wireless Communication". In: *2020 International Conference on Information Networking (ICOIN)*. 2020, pp. 459–464. DOI: 10.1109/ICOIN48656.2020.9016462.

[12] Yuchen Li et al. "Jamming Detection and Classification in OFDM-Based UAVs via Feature- and Spectrogram-Tailored Machine Learning". In: *IEEE Access* 10 (2022), pp. 16859–16870. DOI: 10.1109/ACCESS.2022.3150020.

[13] Robert B. Cleveland et al. "STL: A Seasonal-Trend Decomposition Procedure Based on Loess (with Discussion)". In: *Journal of Official Statistics* 6 (1990), pp. 3–73.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[15] François Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357. URL: http://arxiv.org/abs/1610.02357.

[16] Walid Saad et al. *Wireless Communications and Networking for Unmanned Aerial Vehicles*. Cambridge University Press, 2020. DOI: 10.1017/9781108691017.

2.6. Article #6: Latent Space Transformers for Generalizing Deep Networks

This paper introduces a novel concept for interoperability in deep networks using standardized latent space transformers, aimed at facilitating information sharing between different deep learning models.

Key Contributions

- A novel framework for sharing information between trained deep networks;
- A method for combining networks using latent space transformers;
- An approach to reducing retraining requirements in network adaptation;
- Application scenarios for 5G network optimization.

Proposed Approach

The proposed approach involves:

- Splitting deep networks into two parts with standardized latent spaces;
- Using transformer blocks to convert between different latent spaces;
- Enable network combination with minimal retraining;
- Optimizing edge cloud processing in 5G networks.

Applications and Benefits

Applications and benefits of the proposed approach include:

- Reduced training processing costs;
- Improved network security integration;
- Enhanced resource allocation in 5G networks;
- Efficient data handling between edge and cloud systems.

Article Details

- **Title:** Latent Space Transformers for Generalizing Deep Networks
- **Authors:** Hamed Farkhari, Joseanne Viana, Luis Miguel Campos, Pedro Sebastião, Albena Mihovska, Purmina Lala Mehta, Luis Bernardo.
- **Status:** Accepted in a Conference
- **Conference:** 2021 IEEE Conference on Standards for Communications and Networking
- **DOI:** 10.1109/CSCN53733.2021.9686099

Latent Space Transformers for Generalizing Deep Networks

Hamed Farkhari*†, Joseanne Viana †‡, Nidhi ¶, Luis Miguel Campos *, Pedro Sebastião †‡,
Albena Mihovska ¶, Purnima Lala Mehta ¶, Luis Bernardo §
*PDMFC, Rua Fradesso da Silveira, n. 4, Piso 1B, 1300-609, Lisboa, Portugal
†ISCTE – Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
‡IT – Instituto de Telecomunicações, Av. Rovisco Pais, 1, Torre Norte, Piso 10, 1049-001 Lisboa, Portugal
§FCT – Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal;
¶AU – Aarhus University, Denmark
Emails : Hamed_Farkhari@iscte-iul.pt, joseanne_cristina_viana@iscte-iul.pt, nidhi@btech.au.dk, luis.campos@pdmfc.com,
pedro.sebastiao@iscte-iul.pt, amihovska@btech.au.dk, plm@btech.au.dk, lflb@fct.unl.pt

2021 IEEE Conference on Standards for Communications and Networking (CSCN) | 978-1-6654-2349-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/CSCN53733.2021.9686099

***Abstract*—Sharing information between deep networks is not a simple task nowadays. In a traditional approach, researchers change and train layers at the end of a pretrained deep network while the other layers remain the same to adapt it to their purposes or develop a new deep network. In this paper, we propose a novel concept for interoperability in deep networks. Generalizing such networks' usability will facilitate the creation of new hybrid models promoting innovation and disruptive use cases for deep networks in the fifth generation of wireless communications (5G) networks and increasing the accessibility, usability, and affordability for these products. The main idea is to use standard latent space transformation to share information between such networks. First, each deep network should be split into two parts by creators. After that, they should provide access to standard latent space. As each deep network should do that, we suggest the standard for the procedure. By adding the latent space, we can combine two deep networks using the latent transformer block, the only block that needs to train while connecting different pretrained deep networks. The results from the combination create a new network with a unique ability. This paper contributes to a concept related to the generalization of deep networks using latent transformers, optimizing the utilization of the edge and cloud in 5G telecommunication, controlling load balancing, saving bandwidth, and decreasing the latency caused by cumbersome computations. We provide a review of the current standardization associated with deep networks and Artificial Intelligence in general. Lastly, we present some use cases in 5G supporting the proposed concept.**

***Index Terms*—Deep learning, sharing information, latent space, standardization**

I. Introduction

Recommendable advances in Machine Learning (ML) algorithms, computational capacities, processing, preprocessing techniques, and computer hardware have resulted in efficient training methods for Deep Neural Networks (DNNs). In addition, deep feedforward networks have recently provided enhanced acoustic modelling [1]. As a result, the number of use cases for the DNNs in varied fields will witness exponential growth in the future. Increasing demands will make processing time and techniques, parallel computing, and latency highly critical to the connected users. Technologies such as 5G- Ultra-Reliable Low Latency Communications (URLLC), edge, and cloud computing enable the development of applications using deep networks to provide high Quality of Services (QoS) for users with these needs. At the same time, researchers increase the utilization of the mixed deep networks to achieve better performance and higher accuracy. In addition, the innovations with computational techniques and training models will result in evolving neural networks.

To have seamless integration beyond the fifth generation of wireless communications (5G) networks and deep hybrid networks have some open challenges. Standards support innovations, research organizations to build new training models and network architecture to facilitate enormous data and processing capabilities. It is speculated that standardization on latent space will boost research activities towards innovative hybrid networks with reduced or no retraining requirements. Latent spaces define the data representation in another domain space. For instance, it is the space that resulted in modifying some data features like the mathematical transformations. For example, selecting, extracting, and transforming to new domains happens automatically in deep networks, and there are no rules on the number of layers and units per layer. However, because these variables are hyper-parameters and based on the performance achieved. Thus, we propose to use the latent space to reduce the amount of retraining by separating deep networks. The contributions of this paper are the following:

1) A concept for sharing information between several trained deep networks from different fields (e.g. text and speech and images, resource allocation and security algorithms and, others) is able to decrease latency and computation requirements in deep networks applications reducing training processing costs;
2) New concepts for training techniques to create hybrid networks from pretrained networks;
3) New transfer learning methods for using pretrained deep networks with small datasets.

This paper is organized as follows. Section II discusses the state-of-the-art and Section III presents the standardization activities concerning the AI, ML, and deep networks. Section

IV provided limitations of state-of-the-art associated with standardization activities, and Section V addressed the novel concept of reducing the retraining activity and proposes the new concept for sharing information between deep networks. Section VI discusses an integrated view between the proposed idea and the uses cases for 5G networks. Finally, Section VII presents the main conclusions of this work.

II. State-of-The-Art

Artificial Neural Networks (ANN) are networks of connected nodes guided by the associated weights to facilitate the implementation of Artificial Intelligence (AI) to solve real-life problems. ANNs are useful in designing prediction models, automation and control, and applications requiring trained datasets to make decisions or identify patterns. ANNs are adaptive to the learnings from the information they carry. This information is processed using mathematical/ computational models. The nodes are assembled into layers that perform transformation operations to the inputs [2]. Information travels through multiple layers. ANNs are trained by adapting to network parameters and environment. There are various ways to train a network, for example, supervised learning, unsupervised learning, reinforcement learning, self-learning and, so on [3]. Deep Learning (DL) is an approach where the network observes, identifies, and learns the representations required to process and categorize the raw data. A multi-layered ANN capable of modelling complex linear or non-linear relationships is a Deep Neural Network (DNN). DNN formulates compositional models from the structured or unstructured input datasets and extracts features from different layers. These networks are well-versed to create approximate models with the provided data input. The data flow from the input layer to the output layer, and thus, these networks are also called feedforward networks.

The flow of data can be expressed as follows;

1) The weights of neurons in a DNN is initialized by the random numbers.
2) The output is generated using the activation function after multiplying the inputs with the associated weights.
3) An optimization algorithm will update the weights if the desired accuracy is not achieved.

A. Hybrid Deep Networks and Sharing Information

The researchers mention two kinds of hybrid deep networks in the literature—the combination of one deep network followed by machine learning algorithms such as Support Vector Machines (SVMs). For example, in [4], the hybrid combination of Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) with SVM is compared to achieve higher accuracy for sentiment analysis. In different methods like in [5], several deep networks such as CNN, LSTM, Bidirectionally Long Short-Term Memory (BiLSTM), Gated Recurrent Unit (GRU) were used separately to extract the features, and by concatenating all of these features followed by the Softmax layer, the hybrid network was created and was used for sentiment classification. In another way of the combination of deep networks as a series, one followed after another like [6], the authors for analysis Human Activity Recognition (HAR), used the CNN network followed by another RNN network type, e.g., LSTM, BiLSTM, GRU, and Bidirectional Gated Recurrent Unit (BiGRU). In the other fields, such as security, we were using hybrid deep networks increasing. Another hybrid method was used in [7] for attack detection in the Internet of Things (IoT). As in [6], they used the LSTM network after extracting futures by the CNN network. The recent researchers proved that using hybrid deep networks can improve the performance in many different use cases.

This paper proposes a new concept for standardization related to sharing information between the deep networks without changing the last layers, developing a new individual network, or retraining the pretrained networks. Standards for deep networks are critical because several fields use such networks nowadays (e.g., health, telecommunications, gaming). With standardization, the use cases of deep networks are publicly available, which promotes dissemination and broad application. Furthermore, standardization helps prevent market fragmentation, which inhibits growth, and mutually incompatible solutions are avoided.

III. Standardization Activities

Standards are essential to driving research, innovations, policymakers, and industries. They form a set of guidelines that validate requirements specifications and assure quality [8]. In addition, they align various approaches to have interoperable solutions as we are advancing every day with technology and its vast usages.

There are different types of Standards Development Organizations (SDOs) working towards standards for AI applications. SDOs are categorized at International, National, and Regional levels. Some of the renowned SDOs are ISO (International Organization for Standardization), IEC (International Electrotechnical Commission), ITU (International Telecommunication Union), European Telecommunications Standards Institute (ETSI), etc. In addition, organizations like the 3rd Generation Partnership Project (3GPP), Institute of Electrical and Electronics Engineers (IEEE), and oneM2M are examples of Standard Initiatives groups that collaborate and coordinate standardization efforts on different subjects [8].

A. Standards in Artificial Intelligence

Table I summarizes some of the ongoing standardization initiatives concerning AI/ML architectures and techniques. For the AI ecosystem, standards and specifications are indispensable as they ensure a safer and reliable future. Furthermore, the connected, intelligent devices generate enormous data and the information required for the training models. Furthermore, data is critical and essential in intelligent environments as they include personal as well professional details. For instance, in healthcare scenarios, data cannot be shared or used for training purposes [20]. Thus, it is of utmost importance to have a specified requirement to regulate data

TABLE I
STANDARDIZATION ACTIVITIES CONCERNING AI/ML/DL

	Standards	Summarized Activities
1	IEEE P2830, Standard for Technical Framework and Requirements of Shared Machine Learning ITU T Y.3172 [9]	This standard defines the framework and architecture for the training model using multi-source encrypted data in a trusted third-party environment. Its emphasis is on the use of a third-party execution environment to process encrypted data. The standard intends to provide a verifiable basis for trust and security and outlines functional components, workflows, security requirements, technical requirements, and protocols.
2	P3333.1.3/D2-IEEE Draft Standard for the Deep Learning-Based Assessment of Visual Experience Based on Human Factors [10]	This standard is dedicated for defining deep learning-based metrics of content analysis and quality of experience (QoE) assessment for visual content. It targets to contribute to an enhanced user experience. To achieve high QoE, this working group is focused on areas concerning perceptual quality and virtual reality (VR) cybersickness. Its DL models count for affecting human factors, reliable test methodology and a database construction procedure. It also defines cases for deep analysis of clinical and psychophysical data, deep personalized preference assessment of visual contents, and building image and video databases.
3	Focus Group on Machine Learning for Future Networks including 5G (FG-ML5G) [11]	FG-ML-5G is an ITU-T Study Group 13 (SG13) Focus Group on Machine Learning for Future Networks including 5G. It has documented 10 technical specifications for ML for future networks, including interfaces, network architectures, protocols, algorithms, and data formats. It was active from January 2018 until July 2020. Following are some of the relevant contributions from this focus group concerning the proposed work. 1) ITU-T Y.3172: Architectural framework for machine learning in future networks including IMT-2020. 2) ITU-T Y.3173: Framework for evaluating intelligence levels of future networks including IMT-2020. 3) ITU-T Y.3174: Framework for data handling to enable machine learning in future networks including IMT-2020. 4) ITU-T Y.3176: ML marketplace integration in future networks including IMT-2020. 5) Serving framework for ML models in future networks including IMT-2020.
4	ITU-T Y.3172 [12]	ITU-T Y.3172 provides an architectural framework for machine learning in future networks including IMT-2020. It specifies a set of architectural requirements, components and, their integration guidelines. It defines an ML pipeline, ML management and, orchestration functionalities.
5	ITU-T Y.3173 [13]	ITU-T Y.3173 specifies a framework for evaluating the intelligence of future networks including IMT-2020 and introduces a method for evaluating the intelligence levels of future networks including IMT-2020. It defines an architectural view for evaluating network intelligence levels based on the recommendation in ITU-T Y.3172.
6	ITU-T Y.3174 [14]	ITU-T Y.3174 Framework for data handling to enable machine learning in future networks including IMT-2020. It describes the requirements for data collection and processing mechanisms in various usage scenarios for ML and drafts a generic framework for data handling and examples of its realization on specific underlying networks.
7	ITU-T Y.3176 [15]	ITU-T Y.3176 provides ML marketplace integration in future networks including IMT-2020 and provides a high-level requirements and the architecture for integrating ML marketplaces based on the requirements in ITU-T Y.3172.
8	AI Ecosystem Standardization Program at the European Commission Workshop [16]	IEC and ISO organized a workshop on the AI Ecosystem Standardization Program to fully exploit the potential of AI across Europe and guarantee Europe's leading position in AI. It summarizes varied initiatives in individual EU nations and provides an initial snapshot of the European AI landscape.
9	Securing Artificial Intelligence [ETSI GR SAI 005) [17]	ETSI GR SAI 005 focuses on deep learning and explores the existing mitigating countermeasure attacks. It describes the workflow of machine learning models where the model life cycle includes both development and deployment stages.
10	ITU-WHO FG AI4H [18]	The ITU/WHO Focus Group on Artificial Intelligence for Health focuses on creating a standardized assessment framework for AI methods in health. The FG constitutes members from various research organizations, government agencies, healthcare facilities, and many more. FG AI4H is a joint initiative from ITU and World Health Organization (WHO).
11	ITU-WHO FG AI4NDM [19]	The ITU/WMO/UNEP Focus Group on Artificial Intelligence for Natural Disaster Management (NDM) focuses on establishing a roadmap for an effective and secure use of AI methods for NDM. The FG activities include data collection and handling, improving modelling across spatiotemporal scales, and providing effective communication.

sharing and analysis. IEEE is working towards the Ethically Aligned Design for AI [21], and also European Union's (EU) General Data Protection Regulation (GDPR) [22] sets regulations on how the data can be used. AI and ML is an extensive and open area where the details at each level are crucial.

IV. SHORTCOMINGS

In the following, we summarize the limitations in the current standardization related to deep networks, which motivate us to propose new standards related to sharing content between such networks.

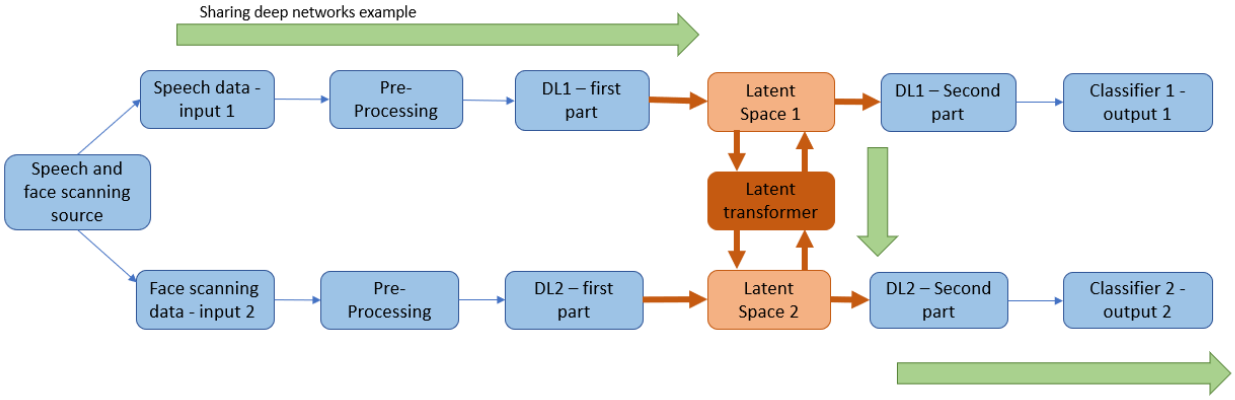


Fig. 1. Latent Transformer to Mix Two Deep Networks with Different Purposes (e.g. speech and face scanning)

A. *Limitations on the State of the Art*

There are limited algorithms like SVMs supported by a solid mathematical background in machine learning and deep networks literature. However, these methods still need user intervention in selecting some hyper-parameters such as kernel-trick mode or the value of regularization, which are difficult to define in deep networks. These parameters are tuned commonly by grid searching on all possible domain values. This limitation causes the training of deep networks to be computationally costly for researchers, and improving the deep networks is possible for only a few organizations or labs that have access to enough hardware and data resources. Based on this lack, the methods like transfer learning were invented to retrain the pretrained deep networks by the new small datasets used by different researchers. However, these replacement methods suffer from complexity and also low compatibility with new other datasets. Based on the number of samples contained in each new dataset and the similarity between the original dataset used to train the deep network and the new dataset, which will be used on the same network, this complexity can be varied. It can cause to change only a few last deep network layers and train only them, or retrain the whole deep network.

B. *Limitations on the Standards*

Studying various standardization activities concerning AI and ML, few standards are available for defining the architecture for preprocessing data and preparing them for machine learning algorithms. Also, there is some guidance for using different blocks and techniques in the typical programming frameworks like Pytorch or TensorFlow for deep networks. However, improving the performance of deep networks needs some standardization beyond the existing ones. The two usual ways for enhancing the existing deep networks are using more data or adding more layers. Nonetheless, methods such as distillation may help reduce the size of deep networks, but they are not enough. Therefore, new standardization needs to separate Deep networks into independent parts. These parts enable researchers with low resources to use the parts, improving or replacing parts of their networks without training. In addition, the combination of different trained parts from several deep networks will be provided.

V. Proposed Idea

According to [23], latent space is a different domain space where data can be decreased to represent new optimal features adequately. The new features may be more distinguishable per each class which facilitates solving classification problems. Typically, when we modify data features, such as some mathematical transformation, those features will be converted to another domain known as latent space In deep networks, features selection and extraction happen automatically. After each layer, the features are converted into a new domain known as latent space. There are no rules on the number of layers and units per layer. Moreover, the result of each layer may depend on the data availability.

Both the number of units and layers are hyper-parameters and, based on the performance achieved in results, will be changed and are varied from one researcher to another. The concept of separating deep networks into at least two parts is to access latent space defined by a specific standard. However, this latent space should follow some rules and standards, and the number of network layers before it should provide some required quality. A practical example of this idea can be the deep hybrid networks using the encoder parts of autoencoders to transfer features in the new latent space, but not standardized, and then feeding to another deep network.

This specification offers a new level of generality for the latent spaces and the network layers before and after them. In addition, these parts of deep networks will be made reusable without needing retraining by the new dataset. By doing these changes, new transfer learning techniques will emerge, and interoperability between deep networks with different datasets will be possible.

Figure 1 depicts the procedure to mix two deep networks with different purposes (e.g., speech and face scanning) by

only training the latent transformer unit, using the elements of two deep networks with the different tasks using the latent transformers. The arrows include the process steps and the parts which were chosen from each deep network. In the first path, the source generates speech data, followed by pre-processing. It is fed in the first part of the deep network - DL1 and then converted to latent space1. Traditionally, the data would be fed to the second part of the deep network 1 - DL1 and, after classification/regression, we would be able to see the results. The face-scanning data would follow the same method in input 2 toward the deep network 2. Using standarized latent spaces, the latent transformer block could convert data from one latent space to another. This conversion makes it possible to create two other paths using the first part from one network and the second part from another network, as highlighted in Figure 1. Using latent transformers and conversion from one latent space to another enables multiple types of data accepted as input or be created as output. By dividing the pipeline into small parts and replacing only some elements may improve the performance and accuracy of the whole process significantly because the raw data is pre-processed and prepared in the deep network's first part. Furthermore, adding new features and maintenance may be more accessible. Besides mixing both networks, we can create mixed data, increasing the feeling analysis. This new technique also works for parallel-connected deep networks. For example, in the ensemble technique, only the first part of each network needs to be used. By ensemble latents, the amount of prediction calculation in ensembling methods will be expected to decrease because there is no need for the second part of the networks while the final performance increases. It causes the latency prediction of networks also to be improved. A critical implementation of standard latent spaces is providing information sharing and transforming between different deep networks. The idea is that instead of training networks for particular purposes, we can use the combination of general networks, and only the transformer units between them should be trained. It will be happening by generalization provided by standardization of latent spaces in deep networks under the same framework.

The importance of using parts of one deep network combined with the elements of another deep network will be revealed while enough edge computation or bandwidth will not be available for the users. In this situation, the different latent spaces of deep networks with various fields produce different data sizes in latent space. So, this combination can make the same result but with lower edge computation or bandwidth for transferring.

With this standard related to latent spaces available in the research community and between European countries, the ability to use the series of deep networks by using latent transformers will be possible. It makes the complicated tasks more manageable than before, which concludes the integration of multi-services. For example, by combining different elements of deep speech transcription, deep translation, profound text to speech, and finally, deep fake technology, we can have the users from various countries and languages communicating their native languages. It is only one example of how we can, with less effort, facilitate the interaction between humans in real-time. To achieve this, the latent transformers need to be trained and create the required compatibility. This process also can make the new generation of transfer learning for deep networks. Recording data of one latent space can further be analyzed by other techniques and deep networks later.

VI. Generalization of Deep Networks in 5G Communications

Deep Networks generalization by standards in Research and Development (R&D) reduces training processing costs, increases investment in security, provides an innovative solution with information advantage over future competitors in 5G markets and provided experience exchange with essential participants in the standardization process. Thus, standardization generates innovation, expands business access, and internationalizes new technological advances.

The development of the telecommunication systems 5G coincides with the emergence of the IoT, extended reality new use cases, and improvements in deep learning techniques, leading to the development of applications combining them in the future to provide high Quality of Services (QoS). Therefore, it affects accessing the high bandwidth with low latency will be required more than before, and cause to creating the massive amount of transferred data from the edge to cloud for processing. Splitting deep network parts between edge and cloud and transferring the represented latent data can provide a new opportunity for developers to create and improve the new cloud services based on the standard latent domains for deep networks.

Conversion data into common latent space should happen on the edge side to remove the redundancy from data, decrease the data dimension, and apply the super compression techniques as illustrated in Figure 2. The deep network that implements resource allocation algorithms is integrated with the security in the edge using latent spaces transformers and the neural network processing computes the result of the mixed deep network in the cloud. These steps of data reduction followed by the existing or new security standards can provide a high level of personal data protection without significantly increasing the amount of final data rather than the initial for transferring to the cloud. Also, other deep networks related to resource allocation and security techniques can be merged creating, a new secure, optimized algorithm. This process is expected to create an innovative competition between several industries to improve and create better standards for latent spaces or improve the following parts based on the existing latent standards.

In the future, we envisage that a latent space with high quality of service will be a commodity that people will rent or buy to provide services. Hence, we can expect different versions of latent spaces with various QoS requested by the users or the available network bandwidth. This concept also fits the described solution to be compatible with different network bandwidths.

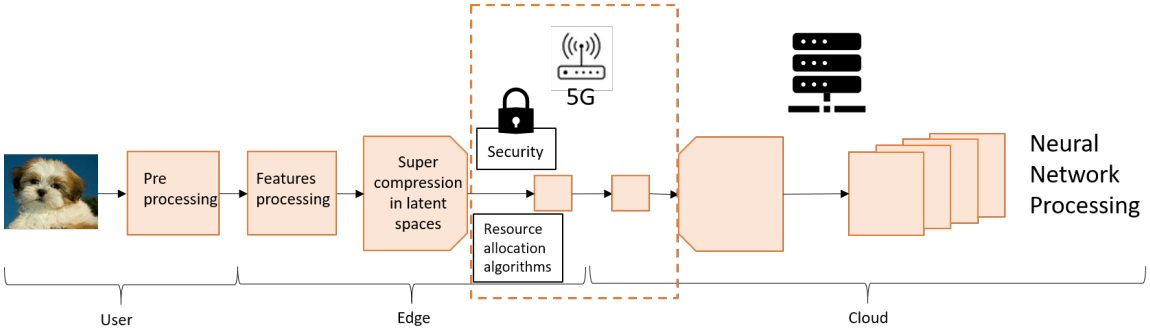


Fig. 2. Use Case for 5G using Deep Networks

VII. Conclusions

Generalizing deep networks by sharing information between them using latent transformers may reduce the costs of the training process. This generalization can be implemented throughout standardization. Additionally, it might create an opportunity for innovation by combining pretrained deep networks to generate other hybrid networks for new purposes, research, and development. There are several standards available for Artificial Intelligence and Deep Learning. However, none of them considers the possibility of using latent transformers blocks for sharing information. Unfortunately, the standards activities are not public, so researchers do not have easy accessibility to all developments and proposed frameworks. Therefore, at this point, assuming the area that we are covering is not in the standards, it is an open research area, and we propose the requirements and related guidelines to develop our concept. Moreover, we showed several use cases applications for this standard (e.g., processing image and sounds, mixing security and resource allocation algorithms in 5G networks and IoT devices, ensembling multiple deep networks and extended reality scenarios).

Acknowledgment

This research received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Project Number 813391.

References

[1] A. R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 14–22, 2011.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[3] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97–116, 2017.

[4] C. N. Dang, M. N. Moreno-García, and F. De la Prieta, "Hybrid Deep Learning Models for Sentiment Analysis," *Complexity*, vol. 2021, 2021.

[5] M. U. Salur and I. Aydin, "A novel hybrid deep learning model for sentiment classification," *IEEE Access*, vol. 8, pp. 58 080–58 093, 2020.

[6] S. Abbaspour, F. Fotouhi, A. Sedaghatbaf, H. Fotouhi, M. Vahabi, and M. Linden, "A comparative analysis of hybrid deep learning models for human activity recognition," *Sensors*, vol. 20, no. 19, p. 5707, 2020.

[7] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Computer Communications*, 2021.

[8] ETSI, "Understanding ICT Standardization - ETSI." [Online]. Available: https://www.etsi.org/images/files/Education/Understanding_ICT_Standardization_LoResWeb_20190524.pdf

[9] IEEE-Standards, "IEEE Draft Standard for Technical Framework and Requirements of Trusted Execution Environment based Shared Machine Learning," *IEEE P2830/D1, October 2020*, pp. 1–21, 2021.

[10] IEEE-SA, "IEEE Draft Standard for the Deep Learning-Based Assessment of Visual Experience Based on Human Factors," *IEEE P3333.1.3/D2, August 2021*, pp. 1–47, 2021.

[11] ITU-T, "Focus Group on Machine Learning for Future Networks including 5G." [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx

[12] ITU, "Architectural framework for machine learning in future networks including IMT-2020," Jun 2019. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3172/en

[13] ITU-T, "Recommendation ITU-T Framework for evaluating intelligence levels of future networks including IMT-2020," Feb 2020. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3173/en

[14] Tsbmail, "Framework for data handling to enable machine learning in future networks including IMT-2020," Feb 2020. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3174/en

[15] ITU-T, "Machine learning marketplace integration in future networks including IMT-2020," Oct 2020. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3176-202009-P

[16] E. Commission. (2020) The European AI Landscape: The Workshop Report. [Online]. Available: https://ec.europa.eu/jrc/communities/sites/jrccties/files/reportontheeuropeanailandscapeworkshop.pdf/

[17] ETSI, "Securing Artificial Intelligence (SAI)," Aug 2021. [Online]. Available: https://www.etsi.org/committee/sai

[18] ITU-T, "Focus Group on Artificial Intelligence for Health." [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/ai4h/Pages/default.aspx

[19] ITU, "Focus Group on Artificial Intelligence for Natural Disaster Management." [Online]. Available: https://www.itu.int/en/ITU-T/focusgroups/ai4ndm/Pages/default.aspx

[20] F. Elmas, "Artificial intelligence." [Online]. Available: https://www.din.de/en/innovation-and-research/artificial-intelligence

[21] J. P. How, "Ethically Aligned Design [From the Editor]," *IEEE Control Systems Magazine*, vol. 38, no. 3, pp. 3–4, 2018.

[22] P. Voigt and A. Von dem Bussche, "The EU general data protection regulation (GDPR)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, p. 3152676, 2017.

[23] A. Oring, Z. Yakhini, and Y. Hel-Or, "Autoencoder Image Interpolation by Shaping the Latent Space," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 8281–8290. [Online]. Available: https://proceedings.mlr.press/v139/oring21a.html

2.7. Article #7: New PCA-based Category Encoder for Efficient Data Processing in IoT Devices

This paper introduces a novel computational preprocessing method designed to convert categorical variables into numerical representations for ML algorithms. The approach is specifically tailored for devices on the IoT with constrained computational resources.

Key Contributions

- Development of a new category encoding method using conditional probabilities combined with PCA;
- The method requires only two hyperparameters: a threshold value and PCA representativeness;
- Validation of the approach using the NSLKDD cybersecurity dataset;
- Comprehensive comparison with 17 existing category encoders in 10 different classifiers.

The proposed method demonstrates superior performance, particularly for categorical variables of high cardiac intensity, while maintaining minimal computational requirements. The core methodology involves the following.

- Converting categorical variables to numerical formats using conditional probabilities;
- Applying PCA to reduce dimensionality while preserving essential information;
- Balancing training and performance testing using harmonic average metrics.

Experimental Validation

The experimental results highlight the following achievements:

- Achieved the highest test accuracy of 89.64% using an SVM classifier;
- Demonstrated an optimal trade-off between training and testing performance;
- Outperformed traditional encoding methods in terms of accuracy and efficiency;
- Provided efficient dimensionality reduction suitable for resource-constrained devices.

Article Details

- **Title:** New PCA-based Category Encoder for Efficient Data Processing in IoT Devices
- **Authors:** Hamed Farkhari, Joseanne Viana, Luis Miguel Campos, Pedro Sebastião, Luis Bernardo
- **Status:** Accepted in a Conference
- **Conference:** IEEE Globecom Workshops 2022
- **DOI:** `10.1109/GCWkshps56602.2022.10008757`

New PCA-based Category Encoder for Efficient Data Processing in IoT Devices

Hamed Farkhari*†, Joseanne Viana †‡, Luis Miguel Campos *, Pedro Sebastião †‡, Luis Bernardo §‡
*PDMFC, Rua Fradesso da Silveira, n. 4, Piso 1B, 1300-609, Lisboa, Portugal
†ISCTE – Instituto Universitário de Lisboa, Av. das Forças Armadas, 1649-026 Lisbon, Portugal
‡IT – Instituto de Telecomunicações, Av. Rovisco Pais, 1, Torre Norte, Piso 10, 1049-001 Lisboa, Portugal
§FCT – Universidade Nova de Lisboa, Monte da Caparica, 2829-516 Caparica, Portugal;
Emails : Hamed_Farkhari@iscte-iul.pt, joseanne_cristina_viana@iscte-iul.pt, luis.campos@pdmfc.com, pedro.sebastiao@iscte-iul.pt, lflb@fct.unl.pt

2022 IEEE Globecom Workshops (GC Wkshps) | 978-1-6654-5975-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/GCWkshps56602.2022.10008757

***Abstract*—Increasing the cardinality of categorical variables might decrease the overall performance of machine learning (ML) algorithms. This paper presents a novel computational preprocessing method to convert categorical to numerical variables ML algorithms. It uses a supervised binary classifier to extract additional context-related features from the categorical values. The method requires two hyperparameters: a threshold related to the distribution of categories in the variables and the PCA representativeness. This paper applies the proposed approach to the well-known cybersecurity NSLKDD dataset to select and convert three categorical features to numerical features. After choosing the threshold parameter, we use conditional probabilities to convert the three categorical variables into six new numerical variables. Next, we feed these numerical variables to the PCA algorithm and select the whole or partial numbers of the Principal Components (PCs). Finally, by applying binary classification with ten different classifiers, we measure the performance of the new encoder and compare it with the other 17 well-known category encoders. The new technique achieves the highest performance related to $accuracy$ and $Area\ Under\ the\ Curve\ (AUC)$ on high cardinality categorical variables. Also, we define the harmonic average metrics to find the best trade-off between train and test performances and prevent underfitting and overfitting. Ultimately, the number of newly created numerical variables is minimal. This data reduction improves computational processing time in Internet of things (IoT) devices connected to future networks.**

***Index Terms*—Categorical Encoders, Dimensionality Reduction, Internet of things, Feature Selection, Machine Learning, NSLKDD, Principal Component Analyses**

I. Introduction

Machine learning (ML) prediction problems require giving the model relevant features to represent the problem accurately. Consequently, data preparation and feature engineering are critical activities for all machine learning algorithms [1]. In Internet of Things (IoT) devices: processing capacity, energy consumption, and resource availability all limit the execution of deep learning algorithms. As the amount of accessible data rises, the degree of diversity of the features increases and this expansion impacts categorical variables. When the number of features grow, the cardinality, which is the number of unique values detected in each feature, increases [2]. The challenge of appropriately and effectively encoding categorical features influence the machine learning model's performance. Handling the conversion from categorical characteristics to numerical features is a well-known issue in data science and machine learning since many methods require numerical input [3]. This problem has several solutions. Specific categorical data encoding schemes are more suitable than others depending on the type of problems i.e. classification or regression. These encoders are critical when processing large volumes of data, especially in IoT devices, at the edge, and in cloud computing because errors and outliers are more common when using these devices to process data. Due to these errors and outliers, reliable statistical estimations are challenging to compute.

One Hot Encoding is the most well-known encoding for low-cardinality categorical features. This yields orthogonal and equidistant vectors for each of the categories. Integers are picked at random because they have no inherent order. An alternate encoding method is Label/Ordinal Encoding, which uses a single column of integers to represent multiple category values. Both encoding techniques present high-dimensional encoding limitations, but Label Ordinal Encoding forces the categories into a particular order. This makes it more difficult for the model to extract valuable information. Regarding the assessment of ML algorithms' success, researchers have used a variety of methodologies such as Recall, Precision, F-Factor area under the curve, true positive rate (TPR), true negative rate (TNR) and $accuracy$ [4]. In most cases, the focus is on specific attributes that matter in the context for which the measure was developed. For example, when Information Retrieval (IR) algorithms are evaluated on Recall, Precision, and F-Factor, erroneous predictions are often overlooked in favor of the accurate ones.

We present a different approach to solving the categorical encoders' modelling problem using conditional probability in supervised learning and Principal Component Analysis (PCA). Further, we compare the performance of our method with several available categorical encoders and classifiers using the same dataset. Finally, we show that our method achieves the best performance by adjusting only two parameters. Our algo-

rithm, which outperforms current machine learning algorithms and reduces the dimensionality of the data, may be a viable choice for IoT devices and cybersecurity algorithms embedded in sensors, UAVs, and other devices. This paper is organized as follows: First, we introduce our novel method to convert categorical to numerical variables considering the probability relationships between categories and target classes in supervised classification. Then, we add a description of the metrics that select the best combinations between them and propose a new metric based on the harmonic averages to highlight the improvements in $accuracy$ during training. After that, we analyze the results from 17 different categorical to numerical encoders using ten different classifiers. We compare the results using the $accuracy$, the $Area\ Under\ the\ Curve\ (AUC)$ and the $harmonic\ averages$, highlighting the improvements in $accuracy$ during training.

A. Contributions and Motivation

Traditional categorical encoders do not provide the parameters to adjust them to the classifiers. Considering this constraint, below is a summary of the main contributions of this paper:

- A new method to encode categorical features using only two hyperparameters: a combination of threshold and PCA that adjusts to different classifiers for maximum performance achievement.
- A supervised category encoder which is suitable for both linear and nonlinear classification algorithms.
- A new metric for measuring training gains in $accuracy$ using $Harmonic\ Averages$ calculations.
- A comparison between the proposed solution and the available categorical encoders using $accuracy$, AUC and the proposed metric based on $harmonic\ averages$.

In high cardinality categorical variables, our method achieves the highest performance using the lowest possible dimensionality, specifically, when the categories exist in the test set and not in the train set. Furthermore, it is possible to prevent or decrease underfitting and overfitting. Also, we define new metrics using a different set of hyperparameters which makes adjustments in the classifiers during the preprocessing steps to improve the performance of our encoder.

II. The Proposed Method

The scheme offers a unique computational preprocessing approach for converting categorical to numerical variables for machine learning (ML) methods. Table I shows the dataset with categorical variables named from $Variable_1$ to $Variable_N$ and each variable contains different numbers of categories. It is required that the target variable defines a binary classification, with two complementary classes.

Using the variables and the target in Table I, we define the conditional probabilities for each unique category using binary classification. The calculation for each category is based on the numbers of its occurrences for each class C_1 and C_2 per its total occurrences as (1) and (2) illustrates :

$Variable_1$	$\cdots$	$Variable_N$	$Target$
$Category_{1,1}$	$\cdots$	$Category_{N,1}$	
$Category_{1,2}$	$\cdots$	$Category_{N,2}$	Class C_1 or C_2 (Binary Classification)
$\vdots$	$\ddots$	$\vdots$	
$Category_{1,j}$	$\cdots$	$Category_{N,j}$	
$\vdots$	$\vdots$	$\vdots$	

TABLE I: Categorical variables with different categories in binary classification.

$$P1_{i,j} = P(Target = C_1 | Variable_i = Category_{i,j}), \quad (1)$$

$$P2_{i,j} = P(Target = C_2 | Variable_i = Category_{i,j}). \quad (2)$$

Before applying the threshold parameter, for each unique $Category_{i,j}$ the following condition holds:

$$P1_{i,j} + P2_{i,j} = 1, \quad (3)$$

where i, j are defined as $\forall i, j | i \in \{1, 2, \ldots, N\}, j \in \{1, 2, \ldots, M_i\}$, N is the number of total categorical variables, and M_i is the number of unique categories for variable i. N and M_i are fixed for each variable. Thus, each category $Variable_i$ will produce two new numerical variables with three states.

$Variable_i$	$New\ Var1_i$	$New\ Var2_i$	Conditions
$Category_{i,j}$	1	0	If $P1_{i,j} > P2_{i,j}$, AND $P2_{i,j} > threshold$.
$Category_{i,j}$	0	1	If $P1_{i,j} < P2_{i,j}$, AND $P1_{i,j} > threshold$.
$Category_{i,j}$	0	0	If $P1_{i,j} < threshold$, OR $P2_{i,j} < threshold$.

TABLE II: Converting each categorical variable to two numerical variables with conditions for each category.

In Table II, $Variable_i$, $NewVar1_i$ and $NewVar2_i$ refers to categorical $Variable_i$, and the first and second newly created numerical variables for $Variable_i$, respectively. New numerical variables will be created based on the probability conditions in (1), (2), and the threshold value. Each categorical value of a database element is converted to the $NewVar1_i$ and $NewVar2_i$ values, where the elements' Category is used to select the value of j in Table II.

A. Threshold

The *threshold* defines the first hyperparameter, which specifies a minimum occurrence probability for a category considered in the binary classification. Probabilities $P1_{i,j}$ and $P2_{i,j}$ are calculated using (1) and (2) based on the classification of C_1 and C_2 of the database samples. Our method creates two new numerical variables for each categorical variable using the equations specified in Table II. Categories with rare elements from one class (with a probability below the threshold) are mapped into $(NewVar_1, NewVar_2) = (0, 0)$. Otherwise, the variables contain the majority class, C_1, $(NewVar_1, NewVar_2) = (1, 0)$, or C_2, $(NewVar_1, NewVar_2) = (0, 1)$.

B. Principal Component Analysis

The second hyperparameter is the number of Principal Components (PCs) available after the PCA processing. The main objective of PCA in our methodology is to remove the correlation between the $2N$ new numerical features in Table II, where N defines the number of categorical variables. The number of PCs, denoted as K, can vary from 1 to $2N$. K can be the minimum number of PCs necessary to capture all data variances, which might be below $2N$ if some numerical variables contain only one unique value for all categories (i.e., only ones or zeroes) or can be written as combinations of other numerical variables. By choosing a lower K, the cumulative data variance will be less than one. We describe the variety of the first and second hyperparameters in the grid search section.

C. Scaling

Usually, scaling is applied before PCA to prevent the feature dominance effect where some features overshadow others because they have different scales. In our method, there is no need for scaling because the new numerical features are normalized between zero and one. However, after the PCA process, the standardization scale using mean and standard deviation is applied for faster convergence in some classifiers, such as Support Vector Machines (SVMs).

D. Dataset

We choose the NSLKDD dataset [5] to test different encoding methods and classifiers because it is common in cybersecurity research (for instance, for network intrusion detection). The NSLKDD is divided into four different partitions: KDDTrain+, KDDTrain+_20Percent, KDDTest+, and KDDTest-21. All partitions are available for downloading from [5]. We use the KDDTrain+ exclusively for training, and the KDDTest+ as a complete test dataset for test purposes which includes all the test instances. A quick analysis of the NSLKDD shows that the KDDTrain+, KDDTest+, KDDTrain+_ 20Percent, and KDDTest-21 contain 125973, 22544, 25192, and 11850 samples, respectively. There are only three categorical variables in the dataset namely: protocol_type, service, and flag. We convert the categorical variables to numerical using different encoders to compare the performance of each method in binary classification.

E. Categorical Encoders Dimensionality

One the the main challenges related to high cardinality categorical variables is their high dimensionality after converting them to numerical features. The One Hot Encoding method presents such constraints. In our proposed method, the number of dimensions of new numerical features varies from a range of one to six. The protocol_type and flag variables in both KDDTrain+ and KDDTest+ sets contain the same cardinality. However, the cardinality of the service variable is greater and different between the train and test sets which may lead to a low performance of the available encoder. Table III shows the differences of dimensionality for newly created numerical features for each of the encoding schemes. The categorical encoders used are from the category_encoders library version 2.2.2. According to Table III, other encoder schemes create at least three dimensions for the new numerical features. In our system, it is possible to reduce them to one.

Encoding Scheme (abbreviation)	Dim.
(Proposed)	**1-5**
Backward Difference Encoder (Backward Difference) [6]	81
BaseN Encoder (BaseN) [7]	13
Binary Encoder (Binary) [8]	13
Cat Boost Encoder (Cat Boost) [9]	3
Count Encoder (Count) [10]	3
Generalized Linear Mixed Model Encoder (GLMM) [11]	3
Hashing Encoder (Hash) [12]	8
Helmert Encoder (Helmert) [6]	81
James-Stein Encoder (James-Stein) [13]	3
Leave One Out Encoder (LOOE) [14]	3
M-estimate Encoder (MEestimate) [15]	3
One Hot Encoder (One Hot) [6]	84
Ordinal Encoder (Ordinal) [6]	3
Polynomial Encoder (Polynomial) [6]	81
Sum Encoder (Sum) [6]	81
Target Encoder (Target) [15]	3
Weight of Evidence Encoder (WOE) [16]	3

TABLE III: Comparing dimensionality of new numerical features created by each Encoding scheme.

F. Classifiers

We use ten classifiers with different configurations in Python v3.6.9 and Sci-kit learn library v0.23.2 to compare the results. Table IV presents the classifiers with hyperparameters. For replication purposes, the seed value of randomness ($random\_state$) in all classifiers is zero.

Classifiers	hyperparameters
Logistic Regression (LR)	solver = 'saga', penalty = 'l2', c = 1.0
Multilayer Perceptron (MLP)	solver = 'adam', alpha = 0.0001, hidden_layer_sizes = 100, activation = relu, learning_rate_init = 0.001('constant'), batch_size=200
SVM 1	kernel = rbf, gamma = 'auto', c=1.0
SVM 2	kernel = poly, gamma='auto', c=1.0, degree=5
SVM 3	kernel = linear, c=1.0
Decision Tree(DT)	max depth=5, split quality measure = 'gini', max features considered for each best split = min(8, number of new numerical features)
Ada Boost Classifier (ADA 1)	base_estimator=DecisionTreeClassifier (max_depth=1), n_estimators=50
Ada Boost Classifier (ADA 2)	base_estimator=DecisionTreeClassifier (max_depth=5), n_estimators=10
Random Forest (Forest)	max depth =5, no. of estimators = 10, split quality measure = 'gini', max features considered for each best split = min (5, number of new numerical features)
Gaussian Naive Bays (GNB)	default sci-kit learn parameters

TABLE IV: 10 Classifiers with hyperparameters used for classification.

G. Metrics

Metrics such as $accuracy$ can simply be measured in multi-class problems. However, other metrics such as precision, recall, FPR, F1-Score, and the sum of the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) cannot be easily calculated [17]. Thus, in practice, $accuracy$ may be enough to check performance in multi-class problems. It is essential to choose the proper metrics to compare the results between the available encoders and the proposed system. We use binary classification and divide the target labels associated with attacks and regular Internet traffic (normal labels). The proportions of attack and normal labels in the train set is 46.54% and 53.46%. In the test set, the ratios are 56.92% and 43.08%, respectively. The percentages of labels in two of the classes show that the number of instances in the train and test sets are balanced. On one hand, balanced classification usually uses $accuracy$ and AUC. On the other hand, unbalanced classification uses precision, F1-score, and other metrics.

H. New Metrics

Commonly, the attacks and normal data in the train and test datasets are not equal. For example, the NSLKDD test set contains only 15% of the total data. The unbalanced test data has an impact on the evaluation of the algorithm's learning capabilities. Even if the test set exhibits an excellent performance, it does not guarantee that the same performance will occur in the training data set and vice versa. We should therefore consider a trade-off between the performances of train and test sets. The effect of changing the amount of data available for the test by 1% is less noticeable that in the train. If the $accuracy$ of the algorithm changes 1% in the test, it affects only 15% of total data, for our data set is 22544 samples. Nevertheless, a 1% change in the training data affects the other 85% of data containing 125973 samples. For the first time, we want to define new metrics to consider both train and test performances because extensive changes may occur in the train when we ignore minimal changes in test performance. In cybersecurity, these changes mean our systems can detect more attacks, and protection increases. We define new metrics and compare our system's performance using both the previous and the new metrics in light of the above explanation. The new metrics are the distance to the ideal point as the error to calculate mean squares errors (MSE) and the harmonic average of the same metrics in the train and test sets. Using only one of these three metrics is adequate for sorting encoder performance and fine-tuning hyperparameters in our proposed encoder. In addition, using these metrics avoids overfitting or underfitting problems, which the following sections discuss. Equations 4, 5, and 6 use the new metrics to estimate performance:

- Mean Square Errors (MSE) to the ideal point for $accuracy$:

$$MSE = 0.5[(100 - a)^2 + (100 - b)^2]; \quad (4)$$

- Mean Square Errors (MSE) to the ideal point for AUC:

$$MSE = 0.5[(1 - c)^2 + (1 - d)^2]; \quad (5)$$

- Harmonic average of the same metrics ($accuracy$ or AUC) in train and test:

$$Harmonic\_avg = \frac{(2.e.f)}{(e + f)}; \quad (6)$$

where in (4), a and b are percentage accuracies in train and test data. In (5), c and d are $AUCs$, for the same data. The harmonic averages in (6) defines e and f using $accuracy$ or $AUCs$ in the data, respectively. The harmonic average is defined to calculate the average between train and test sets for the same metrics. We apply our method to the NSLKDD dataset containing three categorical variables. We use one unique threshold for all of them due to the similar distribution of the classes in the category of the three categorical variables. All threshold values are represented as percentages.

III. Experimental Results

We apply our method to the NSLKDD dataset containing three categorical variables. We use one unique threshold for all of them due to the similar distribution of the classes in the category of the three categorical variables. All threshold values are represented as percentages.

A. Categorical Encoders Comparison

We measure the performance of a combination of 17 different encoders, plus ours from Table III, with the ten classifiers from Table IV, to compare our new proposed encoder algorithm with the other existing encoders. Table V identifies the 18 Encoders by their abbreviations and summarizes their performance results. Each column in the table V associates the encoding scheme with the best suitable classifier according to the train or test for $accuracy$ or AUC. In the fourth column, we use the maximum $harmonic\ averages$ of train and test $accuracies$ to compare the results and sort the encoders from best to worst performance. For example, the test $accuracy$ for the Polynomial encoder is 88.9549%, which is the highest $accuracy$ that this encoder achieves using the GNB classifier. All the encoders are tested with all classifiers and table V presents the classifier with the highest performance. In our method, the hyperparameters Thre(1.87) and PCs(3) represents a threshold of 1.87 % and the top three principal components, respectively. Our algorithm achieves the highest test $accuracy$ of 89.638041 % by feeding only the first principal component to the SVM2 classifier from Table IV, and with the two different thresholds of 3.64 % and 5.45 %. This $accuracy$ is the highest out of all combinations of categorical encoders and classifiers and puts our encoder in the first place. Our method is placed second after Polynomial Contrast coding by choosing the Harmonic average of accuracies as a sorting metric, as is shown in Table V.

Figs 1 and 2 compare the $accuracies$ and $AUCs$ of the 18 encoders with ten different classifiers with respect to the train versus test data. In fig 1, the point at [100, 100] represents

Encoding Scheme	Classifiers with Max. Train *accuracy* (%)	Classifiers with Max. Test *accuracy* (%)	Classifiers with Max. harmonic_avg. of *accuracies* (%)	Classifiers with Max. Train *AUC*	Classifiers with Max. Test *AUC*
Polynomial	ADA2, 96.3167	GNB, 88.9549	GNB, 91.0538	ADA2, 0.9629	GNB, 0.888
Proposed	**All except GNB, Thre(11.9), PCs(1-5), 95.380756**	**SVM2, Thre(3.64, 5.45), PCs(1), 89.638041**	**SVM3, Thre(1.87), PCs(3), 90.6161**	**All except GNB, Thre(11.9), PCs(1-5), 0.953976**	**SVM2, Thre(3.64, 5.45), PCs(1), 0.893252**
Ordinal	ADA2, 96.3151	LR, 83.388	LR, 87.42	ADA2, 0.9629	LR, 0.8514
One Hot	SVM1, 96.3127	DT, 83.7252	ADA2, 87.1133	SVM1, 0.9628	DT, 0.8274
Sum	MLP, 96.3143	ADA2, 79.5245	ADA2, 87.1133	MLP, 0.9628	Forest, 0.814
Target	ADA2, 96.3167	ADA2, 79.5067	ADA2, 87.1081	ADA2, 0.9629	ADA2, 0.808
Backward Difference	ADA2, 96.315	Forest, 80.6112	ADA2, 87.1075	ADA2, 0.9629	Forest, 0.8165
Helmert	ADA2, 96.3127	Forest, 81.2145	ADA2, 87.1038	ADA2, 0.9628	Forest, 0.822
Base-N	SVM2, 96.3127	GNB, 83.6542	ADA2, 87.1035	SVM2, 0.9628	GNB, 0.8534
Binary	SVM2, 96.3127	GNB, 83.6541	ADA2, 87.1035	SVM2, 0.9628	GNB, 0.8534
James-Stein	ADA2, 96.3167	ADA2, 79.4979	ADA2, 87.1028	ADA2, 0.9629	ADA2, 0.808
Cat Boost	ADA2, 96.3159	ADA2, 79.4979	ADA2, 87.1025	ADA2, 0.9629	ADA2, 0.808
GLMM	ADA2, 96.3167	ADA2, 79.4934	ADA2, 87.1001	ADA2, 0.9629	GNB, 0.8148
LOOE	ADA2, 96.3167	ADA2, 79.4934	ADA2, 87.1001	ADA2, 0.9629	ADA2, 0.8079
WOE	ADA2, 96.3151	ADA2, 79.4934	ADA2, 87.0995	ADA2, 0.9629	ADA2, 0.8079
Count	ADA2, 96.3143	ADA2, 79.4535	ADA2, 87.0752	ADA2, 0.9628	ADA2, 0.8074
MEstimate	ADA2, 96.3159	ADA2, 79.112	ADA2, 86.8703	ADA2, 0.9629	ADA2, 0.8046
Hash	MLP, 91.9959	GNB, 77.8921	GNB, 83.4566	MLP, 0.917	GNB, 0.792

TABLE V: 18 different encoders with the best classifier for each one, compared and sorted based on max harmonic average of accuracies. The amount of thresholds for our proposed method are in percentage.

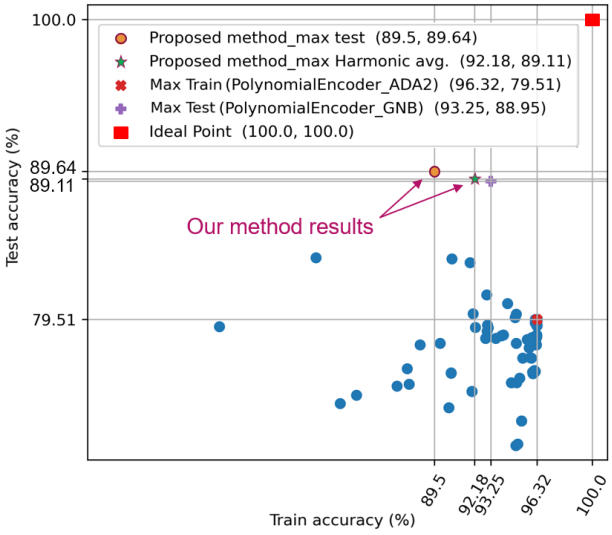


Fig. 1: Scatter of train vs test sets *accuracies* achieved by combination of 18 category encoders.

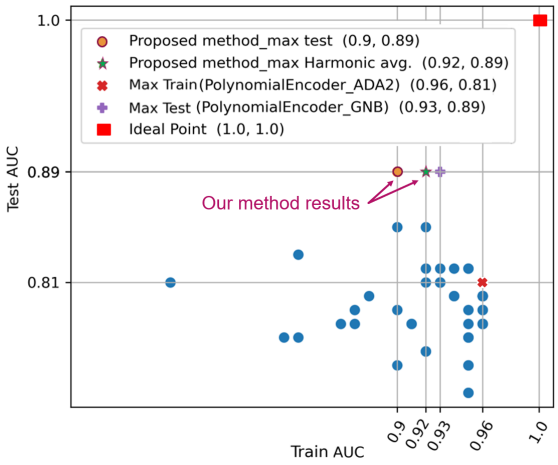


Fig. 2: Scatter of train vs test sets *AUC* achieved by combination of 18 category encoders.

the maximum train and test *accuracy* which is the ideal point of all encoders. After analyzing the available encoders, we discover that the polynomial achieves the greatest train and test *accuracies* with the ADA2(96.31%) and GNB(88.95%) classifiers. Our method achieves the highest **test** *accuracy* (89.64%) with the SVM2 classifier in comparison with the polynomial (88.95%).

Using the harmonic average of accuracies, our method achieves 89.11% during the test phase, which is still the highest test *accuracy*, but lower than the previous test results of 89.64% *accuracy*. However, the amount of train *accuracy* increases from 89.5% to 92.18% and we lose 0.53 % in the test. The difference between the prior test *accuracy* and the harmonic average of accuracies is +2.68 % in the train set and only -0.53% in the test set. The loss between the *accuracy* and the harmonic average metrics for the test set is so minimal and there are significant benefits in the training set which implies that the harmonic average of accuracies is a better metric choice. Fig 2 describes the results based on the *AUC* metric for the same encoders and classifiers. The ideal point is [1.0, 1.0] for the *AUCs* train and test sets. We show that the polynomial and our encoder performances present nearly the same results using the new and previous metrics considering the approximation of two floating points for the test set. The

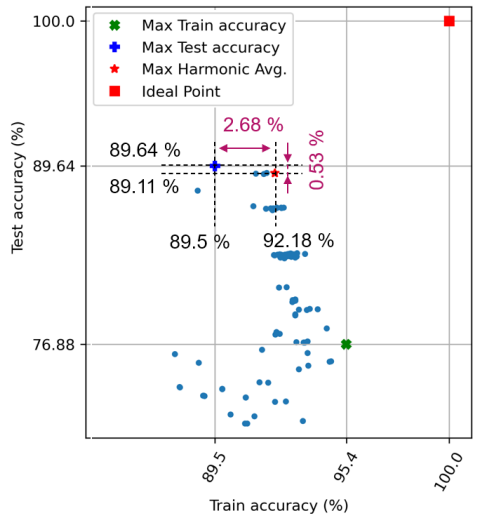


Fig. 3: Scatter of train vs test sets $accuracies$, grid search of two hyperparameters: Threshold and number of principal components followed by 10 different classifiers.

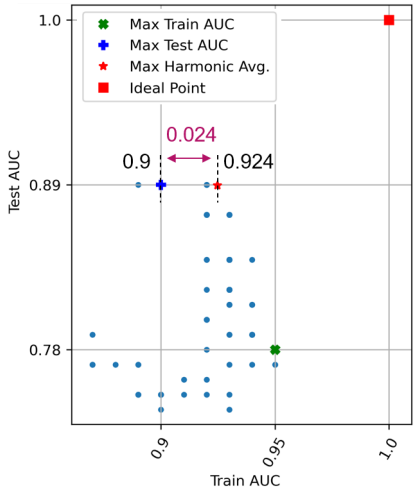


Fig. 4: Scatter of train vs test sets AUC, grid search of two hyperparameters: Threshold and number of principal components followed by 10 different classifiers.

performance for both of them is 0.89 in the test. The difference is in the train in which the polynomial reaches 0.93 while our method achieves 0.92.

B. Grid Search

As previous sections describe, our new proposed category encoder contains two hyperparameters: the threshold and the number of principal components of the PCA. We conduct the grid search for all of the possible combinations of these two parameters to find the best values for each one. For the threshold, we check different values from 0.01% to 50%. Rare categories appear in either less than 1% or less than 5% of all instances. In our results rare categories occur a little more than five percent. We achieve the best test $accuracy$ of 89.64% by choosing 5.45% or 3.64% as the threshold. We check all numbers in the threshold range together with different PC numbers that varies from 1 to 6 as the second hyperparameter.

Figs. 3 and 4 depict the scatter results of $accuracies$ and $AUCs$ for the train versus test sets. Table V shows more information about thresholds, PCs, and classifiers for gaining maximum values for different metrics.

C. Dimensionality

Excluding our encoder, Table III shows the dimensionality output of different category encoders which varies from 3 to 84. We sort the results of each one using different classifiers from Table V. The results based on maximum test $accuracy$ shows that the available encoders with higher dimensionality output have more chances for higher $accuracy$ results. Our method with only one output dimensionality using an SVM classifier defeats all of the other encoders. Prior researchers usually consider the number of PCs that capture 95% or 97% of the variance in the train set for dimensionality reduction problems which means they consider the variance dependency on of the PCs they define. Our results reveal that the number of PCs directly affects the output performance independently of the variance they capture and they should be considered a hyperparameter.

IV. Conclusion

This paper proposed a new method for converting categorical to numerical features, which can be adapted by choosing the correct threshold and number of Principal Components for different classifiers. Furthermore, it produced low dimensional outputs from high cardinality categorical variables. We used $accuracy$ and AUC metrics to compare performances between our method and 17 available encoders. Additionally, we defined new metrics to estimate the trade-off between train and test set performances. Our results overcame the best encoder available for the $accuracy$ test and our method achieved the same result for the AUC test with two floating points approximations. Data preparation and feature engineering are critical steps in every machine learning algorithm. Our encoder can contribute to achieving better performances. Our method involves data compression while translating categorical information, which could be useful in hybrid telecommunication networks such as 5G. Due to the power and resource constraints of IoT devices, our high-performance method may be an attractive solution for particular implementations. We can conclude that the new metrics provide a better trade-off between train and test performances with these results.

Acknowledgment

This research received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie Project Number 813391"

2.8. Article #8: MOOC on "Ultra-dense Networks for 5G and its Evolution": Challenges and Lessons Learned

This paper presents the development and implementation of a Massive Open Online Course (MOOC) focused on ultra-dense networks for 5G and its evolution. The course was created as part of a European MSCA ITN project (TeamUp5G).

Key Aspects

The key aspects of the course include:

- A comprehensive course structure covering 5G technologies and future trends;
- Six modules with five items each, incorporating video content and assessments;
- Collaboration among 15 early-stage researchers and international supervisors;
- A focus on both theoretical and practical aspects of 5G networks.

MOOC Development Process

The development of the MOOC involved the following steps:

- Creating learning materials, including videos and textual content;
- Developing assessment methods and evaluation criteria;
- Ensure quality assurance through a multilayer review process;
- Integrating industry perspectives from non-academic partners.

Course Content

The course content covers the following topics:

- Ultra-dense networks and small cells;
- New transmission technologies;
- Management of interference and energy efficiency;
- Spectrum sharing and carrier aggregation;
- Use cases and prototyping;
- Future technologies and trends.

Article Details

- **Title:** MOOC on "Ultra-dense Networks for 5G and its Evolution": Challenges and Lessons Learned.
- **Authors:** Lopez-Morales et al.
- **Status:** Accepted in a Conference.
- **Conference:** 2022 EAEEIE Annual Conference.
- **DOI:** 10.1109/EAEEIE54893.2022.9819989.

2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE) | 978-1-6654-8442-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/EAEEIE54893.2022.9819989

MOOC on "Ultra-dense Networks for 5G and its Evolution": Challenges and Lessons Learned

Manuel J. Lopez-Morales*, D. Alejandro Urquiza-Villalonga*, Diego Gonzalez-Morin†, Nidhi‡, Bahram Khan§**, Farinaz Kooshki¶, Ahmed Al-Sakkaf*, Leonardo Leyva§††, Hamed Farkhari§, Daniele Medda‖, Ilias-Nektarios Seitanidis‖, Ayman Abu-Sabah§‡‡, Joseanne Viana§ˣ, Pedro Cumino§††, Victor P. Gil-Jimenez*, M. Julia Fernández-Getino García*, Máximo Morales-Cespedes*, Ana Garcia-Armada*, Fernando J. Velez§

*Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain,
†Nokia Bell Labs, Madrid, Spain, §Instituto de Telecomunicações, Portugal
**Universidade da Beira Interior, ††Universidade da Aveiro, ‡‡Universidade Nova de Lisboa, ˣISCTE-IUL, Portugal,
‡CGC Research Lab, Department of Business Development and Technology, Aarhus University, Denmark
¶IS-Wireless R&D, Warsaw, Poland, ‖International Hellenic University, Thessaloniki, Greece
Correspondence should be addressed to Manuel J. Lopez-Morales: mjlopez@tsc.uc3m.es

***Abstract*—Many of the new mobile communication devices will be things that power and monitor our homes, city infrastructure and transport. Controlling drones thousands of miles away, performing remote surgeries or being immersed in video with no latency will also be a huge game changer. Those are some of the few things that make the fifth generation (5G) a revolution expected to be a thrust to the economy. To that end, the design and density of deployment of new networks is also changing becoming more dense, what introduces new challenges into play. What else will it add to previous generations? The MOOC about Ultra-dense networks for 5G and its evolution has been prepared by the researchers of an European MSCA ITN, named TeamUp5G, and introduces the most important technologies that support 5G mobile communications, with an emphasis on increasing capacity and reducing power. The content spans from aspects of communication technologies to use cases, prototyping and the future ahead, not forgetting issues like interference management, energy efficiency or spectrum management. The aim of the MOOC is to fill the gap in graduation and post-graduation learning on content related to emerging 5G technologies and its applications, including the future 6G. The target audience involves engineers, researchers, practitioners and students. This paper describes the content and the learning outcomes of the MOOC, the main tasks and resources involved in its creation, the joint contributions from the academic and non-academic sector, and aspects like copyright compliance, quality assurance, testing and details on communication and enrollment, followed by the discussion of the lessons learned.**

***Index Terms*—Small Cells, energy efficiency, spectrum and interference management, HetNets, IoT, massive MIMO, cell-free, mmWave, VLC, prototyping, UAV, AR/VR, MOOC**

This work has received funding from the European Union Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie ETN TeamUp5G, grant agreement No. 813391.

I. Introduction

A. Motivation and objectives

Massive Online Open Courses (MOOCs) are widely available for everybody with an internet connection. MOOCs are designed to acquire new skills, develop your career, and provide high-quality educational experiences to a large audience in a more affordable and flexible way. Millions of people use MOOCs throughout the world, for professional progress, career transition, and basically any professional training. Several universities and institutions have created and shared their own experience on virtual and remote content creation through MOOC development over the years. Authors from [1] submitted experimental findings from the Virtual Instrument Systems in Reality laboratory. The authors of [2] compared the results of several courses on signal processing and digital communication they had created over the years. In [3], a study about MOOCs' effectiveness in improving undergraduate students' performance in a normal Digital Signal Processing (DSP) class was conducted. There is also a discussion in [4] on the advantages and disadvantages of MOOC courses for microelectronics. The number of discussions in the literature is large, but to the best of our knowledge, there was no MOOC focusing on the 5^{th} generation of mobile communications (5G) and its advancement, which led to the creation of the MOOC addressed in this paper.

The project "New RAN TEchniques for 5G UltrA-dense Mobile networks" (TeamUp5G) [5] is a prestigious Marie Skłodowska-Curie Innovative Training Networks (MSCA ITN) in the frame of the European Commission's Horizon 2020 framework [6], with grant-agreement number 813391. The team is investigating the evolution of the 5G wireless communications and has been preparing an extensive MOOC under the scope of "Ultra-Dense Networks for 5G and Its Evolution". The goal is sharing the recent research advances

and the knowledge about the main technological innovations and new 5G mobile networks applications. Motivated by MOOCs' role in the scope of higher education while providing a positive impact on student's performance, a well-designed, structured, and open comprehensive accessible online course has been prepared by the TeamUp5G team. As an outcome of this join effort, this paper provides the detailed steps and procedures about the methodology adopted and experienced during the preparation of the MOOC, highlighting the experience acquired, challenges, and potential opportunities.

B. Targeted audience

The MOOC was prepared to be simple, understandable and intelligible to the majority of users. In this sense, it can be used for professionals and students who are related to the research and development of 5G New Radio networks and their evolution. Based on the targets for learning outcomes, the transfer of basic concepts is eloquently expressed for beginners and students to make it easier to understand. People with a background in telecommunications can be familiar with the latest objectives and state-of-the-art research areas in which the EU and related companies are willing to invest, research, and develop. Finally, teachers who want to transfer the fundamentals and basic concepts of the 5G networks to their students can also benefit from this MOOC.

C. Content formatting

The "Ultra-dense Networks For 5G And Its Evolution" MOOC [7] is prepared by 14 Early-Stage Researchers (ESRs) under the supervision of an international team of highly qualified professors from different backgrounds and disciplines. The course is divided into six modules and each module contains five different items to cover a wide range of concepts and enabling technologies for the 5G and future 6G. For each item, learning, evaluation, and motivational materials have been created, as shown in Fig.1.

Video-recorded presentations and textual extensions are the main learning materials. The script documents were prepared to assist the presenter in recording the video and to make text transcription easier on the edX platform. The textual extensions have been devised to present students with both reading and hearing information in addition to the video, along with some extra information. The assessment procedure was developed as a method for reviewing the educational material. It is composed of four question types: true/false, multiple-choice, drag and drop, and input number type. Additionally, open questions after each module encourage students to reflect more deeply on the subject through a forum discussion. In this forum, the students and teachers can interact for learning engagement purposes. In total, roughly 2 hours of material was generated for each item, split among 10 minutes of video content each week, 50 minutes of written information to support the video material, plus 1 hour of questions and forum discussion.

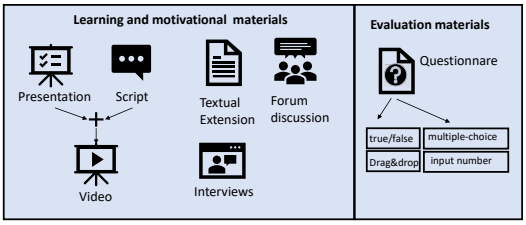


Fig. 1. Block diagram of the content formatting to summarize the contents.

D. Paper Organization

The remainder of the paper is organized as follows: Section II outlines the objective of the MOOC, Section III describes the available resources, Section IV presents the production process, Section V addresses the beta testing and broadcast and Section VI concludes the paper.

II. Objective

The technological development has grown continuously and fast, and the discoveries made by the scientific community and the emergence of new patents bring new challenges at a time when such innovations need to be inserted into people's daily lives. Qualified professionals capable of assimilating new technologies and making good use of them in society are needed. The ESRs, with the help of their supervisors, have observed a gap between innovative evolutionary technologies and the current students' vision over 5G and beyond networks. Addressing this gap is beneficial for students, professionals, and researchers to get updated and understand the latest novel technologies in communication and computer networks. Indeed, after looking more closely at the scope of the necessary road map of the target technologies and their evolution impacting the telecommunications industry, we identified the gaps that could be covered through our MOOC. It is worthwhile to mention that there could be a mismatch between materials provided at university bachelor levels and the online resources from the internet. In general, they do not focus on summarizing the target technologies in a well-developed plan. There is also a mismatch between the research publications which need a prior understanding of the related topic and a very high level of knowledge. They would not be at the level of young students and motivated target researchers.

The goal of this MOOC is to minimize these gaps through efficiently disseminating current research by sharing it into a simple and understandable way to students and young researchers as illustrated in Fig.2. The aim is to deliver this knowledge not only in a high-level view of the 5G mobile network but also exploring the beyond enabling technologies and technical aspects behind each. It is important to consider the need for a creative method for such knowledge sharing to attain effective results. Therefore, the production of the current MOOC package efficiently covers the high-level vision and digs into the technical perspective over the "Ultra-dense Networks for 5G and its Evolution".

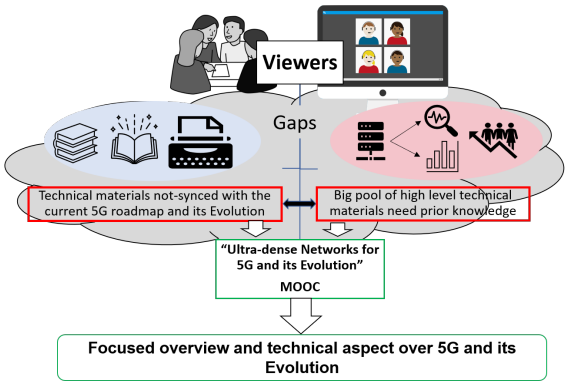


Fig. 2. Showcasing of the objectives (created from free open license CC).

As learning outcomes, this MOOC focuses on understanding, designing and optimizing the 5G heterogeneous small cell ultra-dense networks. Several topics are covered mainly related to 5G and its evolution, its system requirements and new transmission technologies such as beamforming, full-duplex communication, among other. Aspects of interference management and energy efficiency, low power networks, packet and multi-band scheduling, data sensing, spectrum sharing, carrier aggregation, use cases and prototypes (such as augmented and virtual reality), security, unmanned aerial vehicles (UAVs), simultaneous radar and communications (RADCOM), followed by a vision on the future ahead (e.g., 6G and terahertz communications) are also addressed in this MOOC. Students, researchers and practitioners will understand how, motivated by user needs, mobile networks are evolving toward 5G new radios, and how this evolution enable other industrial sectors, such as medical science, transport, entertainment, and education. Practical considerations on these topics are complemented by development and deployment aspects.

Through our produced visionary MOOC, we target the transfer of knowledge in a crystal-clear technical language. After developing a good understanding of the vision for the audience, we share the technical perspective of each key technology player in a smooth manner. It would enable the audience to get the readiness for understanding the latest developments. So, their mind's creativity for contribution in their future careers would be enhanced.

III. Resources

A. Team and project

TeamUp5G is a multi-partner research training network whose beneficiaries come from academic and non-academic sectors to form a structured, international, intersectoral, and interdisciplinary research and training environment for PhD students and young researchers, which is spread in different countries in Europe. It aims to optimize the existing 5G in various domains in terms of throughput, energy and spectral efficiency. Some challenges are the demand for increasing data rates and users served per km^2 and the energy efficiency of the entire system. The goal of the ETN is to propose metrics and develop energy-aware algorithms and protocols to enhance small cells in ultra-dense deployments, making use of massive antenna solutions (mMIMO), millimeter wave (mmWave) bands and Visible Light Communications (VLC), in relevant scenarios, through a combination of analytical work, simulation and prototyping. The details and information regarding our ESRs, their works, and the hosting institutions can be found in [5].

B. University facilities and prior experience

The technical team of Universidad Carlos III de Madrid (UC3M) and some of the involved supervisors had prior experience creating and organizing MOOCs [8]. UC3M provided around 35 different MOOCs both in English and Spanish in the edX and MiriadaX platforms. For instance, the course on mobile communications from the Signal Theory and Communications department at UC3M is published in edX [9]. This MOOC is open to the public, and targets an audience who have no previous knowledge on mobile communications. UC3M experience guided the journey of this MOOC and helped the team to overcome the challenges.

The MOOC was fully recorded at UC3M, utilizing the in-campus Audio/Video (AV) facilities. UC3M has three recording studios, to allow university staff and students to generate teaching materials for various purposes, such as MOOCs or teaching innovation projects. The rooms are provided with all the recording facilities such as HD cameras, a system for mixing and compositing images in HD, special background lighting for generating virtual background, and a teleprompter, as shown in Fig. 3. Concerning the prior experiences in MOOC production, UC3M has experienced staff for editing, mixing, and processing videos. UC3M also provides support for creative process such as covers, course images, and original creation of materials, like animations or even small interactive materials.

Fig. 3. Recording room facilities available at UC3M Leganés Campus.

C. ETN contributions and resources

The MOOC "Ultra-Dense Networks for 5G and its Evolution" results from a great teamwork, supervision and constant guidance. In its production, 14 ESRs and 9 supervisors have participated. From the 14 ESRs, 2 acted as both producers and supervisors of the MOOC, as it happened with 3 of the supervisors of the TeamUp5G project. The other 12 ESRs and the other 6 supervisors acted only as producers and as supervisors of the MOOC, respectively. Each producer

was responsible for the content of the MOOC relevant to one's research area. The contributions by the supervisors were invaluable in coordinating the teams, reviewing and providing continuous insights on improving the content. In Section IV, the MOOC's structure and contents are discussed in detail. To ensure high-quality videos and synchronization, UC3M took the responsibility of recording and coordinating the MOOC. Some producers could not travel to the UC3M premises amid the COVID-19 pandemic. For this reason, some of the producers residing in Madrid recorded most of the videos.

IV. Production of the MOOC

This Section includes information about the timeline of the main tasks, the creation of the material, the copyright compliance, the contribution from the non-academic sector and the quality assurance. Fig. 4 shows an overview of the timeline, involved tasks, copyright, and quality processes of the production of the MOOC.

A. Main tasks and timeline

The kick-off meeting was in early March 2021, when the MOOC structure was defined. The two major goals were to begin the video production phase in late July 2021 and to finish the entire MOOC in January 2022, in order to begin the lessons at the end of February 2022. Six different modules were identified, each one divided into five items, spanning from introductory topics to more technical ones. To structure the overall work, a table of contents for each item was proposed in April 2021. Based on this defined structure, the production of the presentations and scripts of all the modules was carried out during May and June 2021. A common template was used to maintain a homogeneous environment throughout the entire MOOC. We focused on having as less text as possible in the videos, in order to keep an adequate level of attention. Also, a great number of illustrations (both images and schematics) were used to take advantage of visual learning. In the end, this phase has proven to be the most challenging one, both in terms of research and time. Since the maximum duration for each video was set to 7 minutes, the use of written scripts became essential to ensure compliance with this limitation.

The videos were recorded during June, July, and September 2021, supported by the presentations and scripts. Among the parties involved, only the UC3M had adequate facilities for multimedia production (i.e., filming and video production) and the best way to have a centralized quality control was to record all the videos in the UC3M, using a small selected group of people, containing both instructors and speakers. The filming process took about three months.

Apart from the video, a textual extension as additional studying material was provided. The starting point for the textual extensions was the previously written scripts. In addition, some particularly complex topics were further extended to provide a more complete information. In order to provide a homogeneous result, a common template was used for all textual extensions. The textual extensions were created in October and November 2021.

The evaluation questions were also created during these months. Two different evaluation phases were defined: a test related to each item and a more general test for the entire module. The item-wise test contained 6 questions and a starting point topic (including references) to be used for general discussion purposes. The module-wise test featured 10 questions regarding every item included in the module. Both the item and module tests featured different test modalities (true/false, multiple-choice, drag&drop, and numerical answer), to avoid them becoming tedious.

Besides, a forum discussion was proposed in each item to motivate the active participation of all the students of the MOOC. December 2021 was used as a quality assurance month of the contents produced to correct them and to ensure a proper quality. Finally, the beta testing was realized in January and February 2022. Fig. 4 shows the general timeline of the MOOC.

B. Organization and creation of the study material

The content of the MOOC was divided into six modules, each with 5 items:

- Module 1 – "Ultra-dense networks and small cells" introduces to the audience the ultra-dense network, 5G, new scenarios as well as innovative applications. Besides, it introduces the emerging technologies for 5G.
- Module 2 – "New transmission technologies" focuses on the physical layer transmission technologies like massive MIMO, beamforming and full-duplex technologies, as well as VLC.
- Module 3 – "Interference management and energy efficiency" presents scheduling mechanisms, the cell-free paradigm and approaches for energy efficiency.
- Module 4 – "Spectrum sharing and carrier aggregation" introduces the fundamentals of Carrier Aggregation (CA), the coexistence of small cells and Low Power Wide Area Networks and architectures for spectrum sharing.
- Module 5 – "Use cases and prototyping" presents testbeds, the privacy issue in communications and some insight about AR/VR and immersive rendering.
- Module 6 – "The future ahead" introduces emerging technologies like RADCOM, THz communications, and early discussion about what 6G will be. Besides, it summarizes the own experience of the TeamUp5G ETN.

The content for each of the items was created by the ESRs and supervisors within the TeamUp5G ETN. Besides, it is important to highlight that TeamUp5G members are spread all over Europe. Therefore, the pandemic situation originated by COVID-19 highly limited the planning and brainstorming events for the MOOC. This meant that almost all the content creation process was carried out online, mainly with email exchange and teleconference meetings.

After defining the MOOC structure and recording capabilities (i.e., facilities and human resources), the specific content of each item was discussed between the members of each module, targeting coherence, and avoiding content overlap between items. This discussion was a nice experience that

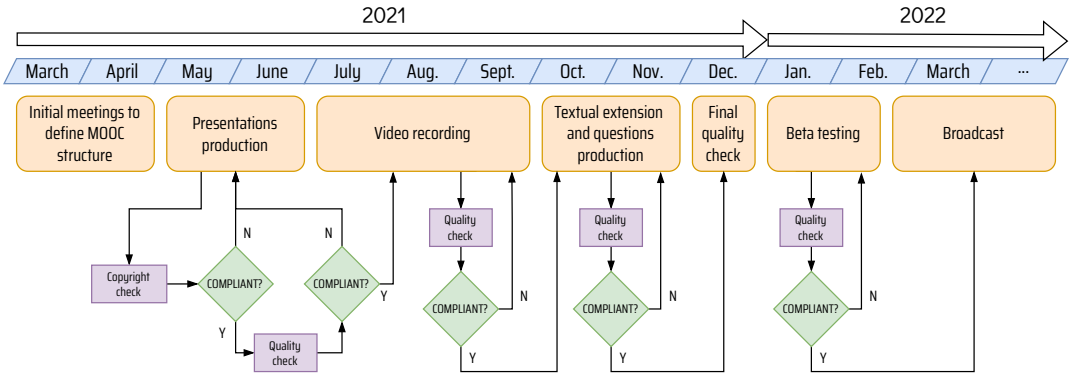


Fig. 4. Overview of the production timeline, involved tasks and copyright and quality processes of the production of the MOOC.

allowed ESRs and supervisors to share knowledge and find ideas for networking. The next step was the writing of the main ideas for the script of the video and initial structure of the items. This initial content was reviewed by the supervisors of each module and feedback was given to the researches in charge of the items. The initial iteration identified several issues like the heterogeneity of the slideshows (e.g., design, animations, fonts, and number of slides) and the use of images with poor quality or subject to copyright. Many of the original images were used in the classroom during teaching activities, so they did not fulfill the required quality for a MOOC. Consequently, in a second iteration, a slideshow/video template and specific guidelines were provided for the content creation, which ensured homogeneity between items. When the slideshows and scripts were ready, the video recording process started, which led to the production team to provide specific guidelines for recording, but induced changes in the already approved slideshows. Some of the main issues found were the use of a large amount of text in the slideshows. Replacing it with illustrations was challenging because of the copyright constraint of the MOOC, explained in the next Section. The videos did not exceed the seven-minute timing constraint and achieved the required presentation quality.

C. Copyright compliance

Any useful lecture requires well designed illustrations to provide useful and complementary visual information to the explained topic. Public and massive lectures as the one in a MOOC not only require the quality and suitability of the selected illustrations to be high, but also to ensure that all of them, with no exception, are copyright compliant. The selection process is more complex, as the content creators not only need to find or produce high quality illustrations but ensure only the ones with appropriate licensing are selected. We mainly used the following sources: commercial or license-free online repositories, proprietary academic or industrial resources, and custom-made illustrations by the MOOC contributors. Even though all the MOOC authors were very cautious with the aforementioned requirements, all the used resources were double checked by the UC3M production team, which validated each resource's license individually.

D. Contribution from the non-academic sector

The TeamUp5G consortium involves multiple non-academic partners which contributed to give the MOOC a practical approach:

- Nokia Bell Labs: the team from Madrid is focused on the study of the most relevant use cases for 5G and beyond ecosystems. Their research is focused on immersive media offloading and industry 4.0. They have produced or revised the lectures related to the description and analysis of 5G use cases.
- PDMFC: a Portuguese company with the goal of providing solutions in areas such as digital transformation, big data, cloud or security. The contributing team have developed the modules related to network security and how it can be improved with the use of machine learning.
- IS-Wireless: a Polish company that targets software-defined 4G and 5G deployments, with a strong support to the Open RAN community. Their knowledge has been gathered in a module focused on cell-free communications.

E. Quality assurance

A successful MOOC requires high quality content, which demands updated and relevant topics, which have to be adequately explained, up to date and with a professional appearance. For this reason, we have followed a multi-layer quality assurance approach. The first quality check came from the authors themselves: we strongly encourage all the authors to make a huge effort to produce high quality content with the goal of reducing the overhead from successive quality checks. Most of the authors were PhD students. Consequently, the second checking layer were their supervisors, which had a crucial role in the development of the MOOC. To add an extra layer for quality checking, we used a peer-to-peer approach, in which the authors and contributors had to check other contributors' work. In every production step, each author had to review at least two other contributions. We believe this process has helped us accelerating the production of the MOOC while ensuring high quality standards. Finally, all the content was checked by the production team, who was in

charge of evaluating the quality from the audiovisual point of view. Each of the mentioned layers involved several iterations: feedback was given, and new versions were produced. Quality assurance requires available time and effort, and in this MOOC we have committed ourselves to both of them.

V. Tests and broadcasts

A. Beta-testing

After the MOOC was uploaded to edX, a beta-testing process was done by the producers of the MOOC, to find possible deficiencies. A total of 2 weeks were allocated to this process, and the work was divided among the beta-testers, with at least 3 beta testers (2 ESRs and 1 supervisor) per module, to ensure enough people to review each module. After feedback was provided, any remaining issues were corrected.

B. Communication and enrollment

The dissemination of the MOOC was mainly conducted via social networks, email messages, and webpage announcement. Announcements were done using the TeamUp5G project social networks, and the researches involved in the creation of the MOOC were also invited to advertise the MOOC. Several colleagues in academia and industry were contacted, and the MOOC was announced via specialized mailing lists, such as that of the IEEE Communications Society. In each outreach event where the TeamUp5G members participated, the MOOC was advertised. The industry actors involved in the creation of the MOOC were also involved in the communication. The enrollment started 3 months before the broadcast, which was scheduled for the 22^{nd} of February 2022, and a strong communication campaign started 3 weeks before this date, i.e. the 1^{st} of February 2022. A total of 144 students were enrolled at the start date of broadcast, and it finished with a bit less than 250 students, with a diverse geographical distribution of about 65 countries/regions and a diverse education distribution from secondary school to doctorate, with the masters being the most representative and the secondary the less representative.

C. Broadcast

The broadcast started on February 22nd, 2022. Two ESRs which were part of the main authors, were actively involved in the forums to respond to doubts and to ensure no inappropriate messages were posted. Active participation among students was suggested and positively followed by them, and supported by the two above mentioned ESRs, with positive feedback. Some corrections were made during the broadcast whenever necessary, by supporting on the comments from students.

VI. Lessons learned and conclusions

The production of a MOOC involves a great amount of work. The most complex task was not only the production of the content itself, but also the coordination of the producers and supervisors. More than 20 people from 5 different countries have been involved in the production of this MOOC and all the work has been carried out online. Therefore, although our project is composed of great professionals, there were some coordination and miscommunication problems between the supervisors and the content producers causing some delays. Besides, the resources to guarantee the recording quality were available at the UC3M premises in Madrid. Hence, some items were not recorded by the authors but by producers residing in Madrid. All these coordination issues implied that efforts had to be doubled to achieve a high-quality outcome.

MOOC planning is a crucial task. From the beginning, it is necessary to have a well-defined structure with all the expected content, and the resources available to produce this content. The deadlines for the production, review, and acceptance of the content with the expected quality should be properly scheduled. Periodic monitoring should be planned to check the work progress and to ensure there are no doubts on the producers. In addition, the active cooperation of all authors of the MOOC is essential. Although, in general, many of the MOOC producers were not initially aware of the work required to create high-quality content that meets the expectations of a well-prepared audience, they all agree that it has been a rewarding learning experience.

To conclude, the MOOC on "Ultra-dense Networks for 5G and its Evolution" has been presented. We have addressed the objectives, the resources that were available, the production of the MOOC itself and its broadcast. Although there have been some mistakes during content creation and recording, lessons have been learned and important conclusions have been drawn to improve future MOOC recordings.

Acknowledgements

We would like to acknowledge the audiovisual team at UC3M in charge of the edition and the support team in charge of copyright checking and uploading the content to the edX platform. We would also like to acknowledge the work of all the people involved in the development of this MOOC who do not explicitly appear as authors of this manuscript.

References

[1] F. García, G. Díaz, M. Tawfik, S. Martín, E. Sancristobal, and M. Castro, "A practice-based MOOC for learning electronics," in *2014 IEEE Global Engineering Education Conference (EDUCON)*, 2014, pp. 969–974.

[2] T. A. Baran, R. G. Baraniuk, A. V. Oppenheim, P. Prandoni, and M. Vetterli, "MOOC Adventures in Signal Processing: Bringing DSP to the era of massive open online courses," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 62–83, 2016.

[3] S. Pertuz and J. Torres, "The impact of MOOCs on the performance of undergraduate students in digital signal processing," in *2016 XXI Symposium on Signal Processing, Images and Artificial Vision (STSIVA)*, 2016, pp. 1–7.

[4] L. Stuchlíková, A. Kósa, P. Benko, and D. Donoval, "Massive open online courses in microelectronics education," in *10th European Workshop on Microelectronics Education (EWME)*, 2014, pp. 31–36.

[5] "TeamUp5G – new RAN TEchniques for 5G UltrA-dense mobile networks – TeamUp5G," https://teamup5g.webs.tsc.uc3m.es/.

[6] Laporma, "Marie skłodowska-curie actions," Feb 2021. [Online]. Available: https://ec.europa.eu/programmes/horizon2020/en/h2020-section/marie-sklodowska-curie-actions

[7] "Ultra-dense networks for 5G and its evolution," https://www.edx.org/course/ultra-dense-networks-for-5g-and-its-evolution.

[8] "UC3M MOOCs." [Online]. Available: https://www.uc3m.es/uc3mdigital/moocs

[9] "Fundamentos de las comunicaciones móviles: En la palma de tu mano." [Online]. Available: https://www.edx.org/es/course/fundamentos-de-las-comunicaciones-moviles-en-la-pa

CHAPTER 3

Conclusions

The articles demonstrate the rapid evolution of approaches to detecting jamming attacks on UAV communications, progressing from statistical methods to sophisticated deep learning architectures. The initial research combined statistical analysis using Seasonal Trend Decomposition (STL) with CNN-LSTM networks, achieving 84.38% accuracy for close-range jammers with the statistical approach and up to 99.99% accuracy with deep learning for higher-power jamming scenarios. This established the superiority of ML approaches while highlighting the continued value of simpler statistical methods for resource-constrained situations.

A significant advancement came with the development of the DAtR architecture specifically designed for 5G UAV networks. This approach innovatively combined RSSI and SINR measurements to detect attacks under both LoS and NLoS conditions. The research demonstrated robust performance in various scenarios, including multiple terrestrial users and static and moving attackers, although performance degraded somewhat in NLoS conditions and with lower-power jamming signals.

The most recent research introduced a hybrid approach combining DNNs with conventional ML algorithms, particularly focusing on the reliability of detection decisions. This work contributed to important innovations in pre-processing through TSA and post-processing techniques to enhance accuracy and reduce false alarms. The hybrid approach proved particularly effective in handling complex urban environments where multiple sources of interference exist.

The articles collectively highlight several critical findings about jamming in UAV networks. First, the power and distance of the jammers significantly affect the detection accuracy, with close-range high-power jammers being easier to detect. Second, channel conditions play a crucial role, with NLoS scenarios presenting significantly greater challenges than LoS conditions. Third, the presence of legitimate terrestrial users complicates detection, requiring more sophisticated algorithms to distinguish between intentional jamming and normal interference.

Implementation considerations emerged as a key theme in all papers. Research shows that while deep learning approaches generally outperform simpler methods, they must be carefully optimized for UAV deployment. Window size selection proves critical, with larger windows improving accuracy but increasing latency and computational overhead. The papers also demonstrate that hybrid approaches that combine multiple techniques can provide better practical performance than single-method solutions.

Looking specifically at jamming patterns, the research reveals that attackers with power levels above 5 dBm are more easily detected, while low-power jammers (below 2 dBm) present significant detection challenges. Moving jammers introduce additional complexity, though the DAtR architecture showed promising results in tracking and identifying mobile threats. The articles also highlight the importance of considering multiple attack scenarios, as jamming patterns and effectiveness vary significantly depending on environmental conditions and network configurations.

These research papers collectively demonstrate significant progress in developing robust jamming detection solutions for UAV networks, while also highlighting remaining challenges for future work. The evolution from statistical methods to hybrid Deep Learning architectures has enabled more accurate detection in diverse operating conditions, with newer approaches achieving up to 99.99% accuracy in favorable scenarios. However, challenges persist in detecting low-power jammers, handling NLoS conditions, and maintaining performance with limited computational resources. Future research directions should focus on further optimizing detection algorithms for resource-constrained UAV platforms, improving performance in challenging urban environments, and developing standardized evaluation frameworks. As UAV applications continue to expand in both commercial and emergency response scenarios, the importance of reliable jamming detection will only increase. The findings suggest that hybrid approaches combining multiple detection techniques, along with sophisticated pre-processing and post-processing methods, currently offer the most promising path forward for practical implementation.

References

[1] T. T. Nguyen and G. Armitage, "A survey of techniques for internet traffic classification using machine learning," *IEEE Communications Surveys and Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

[2] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 343–356, 2010.

[3] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[4] S. Naseer and Y. Saleem, "Enhanced network intrusion detection using deep convolutional neural networks," *KSII Transactions on Internet and Information Systems*, vol. 12, no. 10, pp. 5159–5178, 2018.

[5] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, no. 1, pp. 18–28, 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404808000692

[6] F. Elmas, "Artificial intelligence." [Online]. Available: https://www.din.de/en/innovation-and-research/artificial-intelligence

[7] A.Géron, *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems.* O'Reilly Media, Inc., 2019. [Online]. Available: https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/

[8] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer-Verlag, 2006.

[9] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018.

[10] H. Lee, S. Eom, J. Park, and I. Lee, "Uav-aided secure communications with cooperative jamming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9385–9392, 2018.

[11] A. Marttinen, A. M. Wyglinski, and R. Jäntti, "Statistics-based jamming detection algorithm for jamming attacks against tactical manets," in *2014 IEEE Military Communications Conference*, 2014, pp. 501–506.

[12] L. Xiao, C. Xie, M. Min, and W. Zhuang, "User-centric view of unmanned aerial vehicle transmission against smart attacks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 4, pp. 3420–3430, 2018.

[13] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[14] Y. Li, J. Pawlak, J. Price, K. Al Shamaileh, Q. Niyaz, S. Paheding, and V. Devabhaktuni, "Jamming detection and classification in ofdm-based uavs via feature- and spectrogram-tailored machine learning," *IEEE Access*, vol. 10, pp. 16 859–16 870, 2022.

[15] Z. Lu, W. Wang, and C. Wang, "Modeling, evaluation and detection of jamming attacks in time-critical wireless applications," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1746–1759, 2014.

[16] M. Greenacre, P. J. Groenen, T. Hastie *et al.*, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, p. 100, 2022. [Online]. Available: https://doi.org/10.1038/s43586-022-00184-w

[17] F. Qi, X. Zhu, G. Mang, M. Kadoch, and W. Li, "Uav network and iot in the sky for future smart cities," *IEEE Network*, vol. 33, no. 2, pp. 96–101, 2019.

[18] M. M. Azari, F. Rosas, and S. Pollin, "Cellular connectivity for uavs: Network modeling, performance analysis, and design guidelines," *IEEE Transactions on Wireless Communications*, vol. 18, no. 7, pp. 3366–3381, 2019.

[19] W. Jin, J. Yang, Y. Fang, and W. Feng, "Research on Application and Deployment of UAV in Emergency Response," *ICEIEC 2020 - Proceedings of 2020 IEEE 10th International Conference on Electronics Information and Emergency Communication*, pp. 277–280, 2020.

[20] G. Geraci, A. Garcia-Rodriguez, M. M. Azari, A. Lozano, M. Mezzavilla, S. Chatzinotas, Y. Chen, S. Rangan, and M. D. Renzo, "What will the future of uav cellular communications be? a flight from 5g to 6g," *IEEE Communications Surveys and Tutorials*, vol. 24, no. 3, pp. 1304–1335, 2022.

[21] Z. Kaleem, M. Yousaf, A. Qamar, A. Ahmad, T. Q. Duong, W. Choi, and A. Jamalipour, "UAV-empowered disaster-resilient edge architecture for delay-sensitive communication," *arXiv*, no. December, pp. 124–132, 2018.

[22] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5g and resilient public safety networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 36–42, 2018.

[23] P. Skokowski, K. Malon, M. Kryk, K. Maślanka, J. M. Kelner, P. Rajchowski, and J. Magiera, "Practical trial for low-energy effective jamming on private networks with 5g-nr and nb-iot radio interfaces," *IEEE Access*, vol. 12, pp. 51 523–51 535, 2024.

[24] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," in *2020 International Conference on Information Networking (ICOIN)*, 2020, pp. 459–464.

[25] Y. Shi, X. Lu, Y. Niu, and Y. Li, "Efficient jamming identification in wireless communication: Using small sample data driven naive bayes classifier," *IEEE Wireless Communications Letters*, vol. 10, no. 7, pp. 1375–1379, 2021.

[26] J.-C. Li, J. Liu, B.-G. Cai, and J. Wang, "Jamming identification for gnss-based train localization based on singular value decomposition," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, 2021, pp. 905–912.

[27] M. Cheng, Y. Ling, and W. B. Wu, "Time series analysis for jamming attack detection in wireless networks," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–7.

[28] N. Liu, X. Tang, R. Zhang, D. Wang, and D. Zhai, "A dnn framework for secure transmissions in uav-relaying networks with a jamming receiver," in *2020 IEEE 20th International Conference on Communication Technology (ICCT)*, 2020, pp. 703–708.

[29] J. Gao, M. Wang, L. Chen, B. Hui, C. Wang, and H. Fan, "Drfm jamming mode identification leveraging deep neural networks," in *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2021, pp. 444–449.

[30] M. P. Arthur, "Detecting signal spoofing and jamming attacks in uav networks using a lightweight ids," in *2019 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2019, pp. 1–5.

[31] M. Hachimi, G. Kaddoum, G. Gagnon, and P. Illy, "Multi-stage jamming attacks detection using deep learning combined with kernelized support vector machine in 5g cloud radio access networks," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, 2020, pp. 1–5.

[32] D. Su and M. Gao, "Research on jamming recognition technology based on characteristic parameters," in *2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP)*, 2020, pp. 303–307.

[33] F. Ruo-Ran, "Compound jamming signal recognition based on neural networks," in *2016 Sixth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC)*, 2016, pp. 737–740.

[34] O. Sharifi-Tehrani, M. F. Sabahi, and M. Danaee, "Gnss jamming detection of uav ground control station using random matrix theory," *ICT Express*, vol. 7, no. 2, pp. 239–243, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405959520303040

[35] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Afrl: Adaptive federated reinforcement learning for intelligent jamming defense in fanet," *Journal of Communications and Networks*, vol. 22, no. 3, pp. 244–258, 2020.

[36] M. Liyanage, I. Ahmad, A. B. Abro, A. Gurtov, and M. Ylianttila, *A Comprehensive Guide to 5G Security*. Hoboken, NJ, USA: Wiley, 2018, ch. 6, pp. 117–141.

[37] P. Skokowski, J. M. Kelner, K. Malon, K. Maślanka, A. Birutis, M. A. Vazquez, S. Saha, W. Low, A. Czapiewska, J. Magiera, P. Rajchowski, and S. Ambroziak, "Jamming and jamming mitigation for selected 5g military scenarios," *Procedia Computer Science*, vol. 205, pp. 258–267, 2022, 2022 International Conference on Military Communication and Information Systems (ICMCIS). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050922008924

[38] G. Geraci, A. Garcia-Rodriguez, L. Galati Giordano, D. López-Pérez, and E. Björnson, "Understanding uav cellular communications: From existing networks to massive mimo," *IEEE Access*, vol. 6, pp. 67 853–67 865, 2018.

[39] J. F. Harvey, M. B. Steer, and T. S. Rappaport, "Exploiting high millimeter wave bands for military communications, applications, and design," *IEEE Access*, vol. 7, pp. 52 350–52 359, 2019.

[40] P. Schneider and G. Horn, "Towards 5g security," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, 2015, pp. 1165–1170.

[41] L. Bastos, G. Capela, A. Koprulu, and G. Elzinga, "Potential of 5g technologies for military application," in *2021 International Conference on Military Communication and Information Systems (ICMCIS)*, 2021, pp. 1–8.