# 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 ¶, 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, Hamed_Farkhari}@iscte-iul.pt

***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*—Security, Convolutional Neural Networks, Deep Learning, Jamming Detection, Jamming Identification, UAV, Unmanned Aerial Vehicles, 4G, 5G;**

## 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 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

[§] Collaborative authors with equal contribution.

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 |

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] also identify jamming samples using signal-extracted features, but the authors add another way to detect attacks that employs 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 paper, 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 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 paper 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. In this section, 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 paper.

## 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 × 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

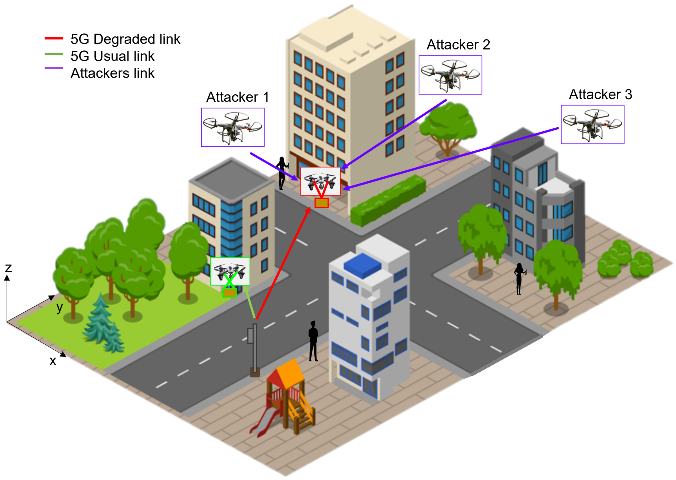


Fig. 1: Simulation scenario.

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 |

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 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), $\eta^{\alpha}$ is the shadowing (in dB), and $\alpha$ 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 Eq. (1) depends on the high/low altitude configurations and the LoS/NLoS conditions. We compute it as follows:

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

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

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

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:

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

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 ($\eta^{\alpha}$) in Eq. (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 [34, 33].

| | 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 ($p_{\text{LoS}}$) is given by:

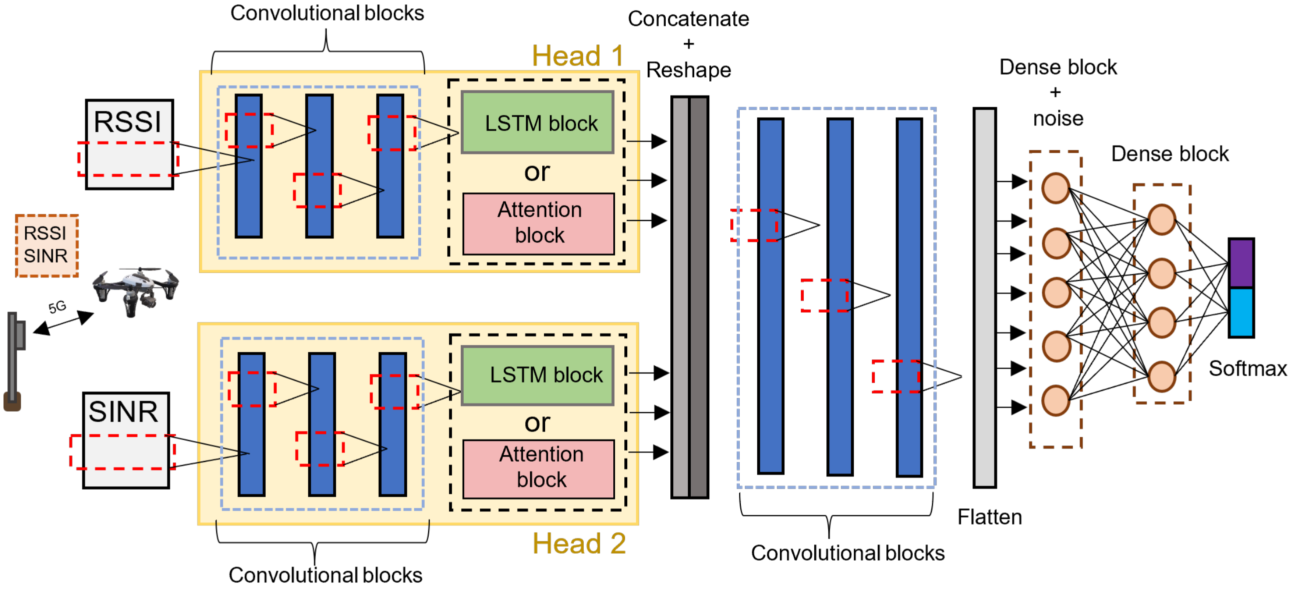


Fig. 2: Multi-Headed Deep Neural Network (MH-DNN) architecture. Note the switch from LSTM to Attention layers.

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

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 [34, 33]. 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, $\Gamma_{uav}$, 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 $P^{i}_{\text{user}}$ and $P^{j}_{\text{jammer}}$ 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, $\zeta^2$ 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 100k ($T < 10^5$), 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 $p_{\mathrm{LoS}}$ and $p_{\mathrm{NLoS}} = 1 - p_{\mathrm{LoS}}$ (according to Eq. (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 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.

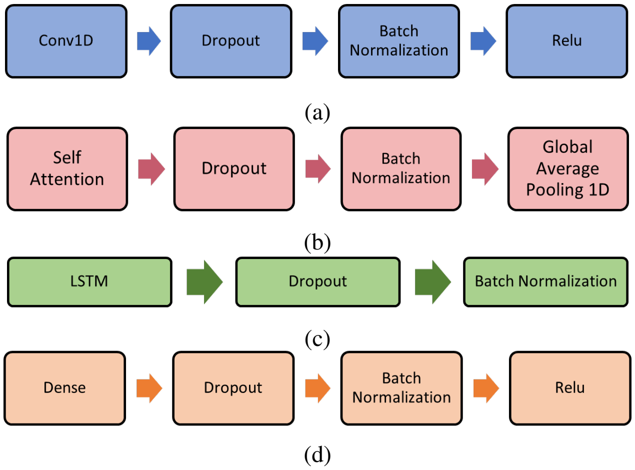


Fig. 3: Detailed block components in the proposed MH-DNN model (a) Convolutional block, (b) Attention block, (c) LSTM block, (d) Dense block.

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 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 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.

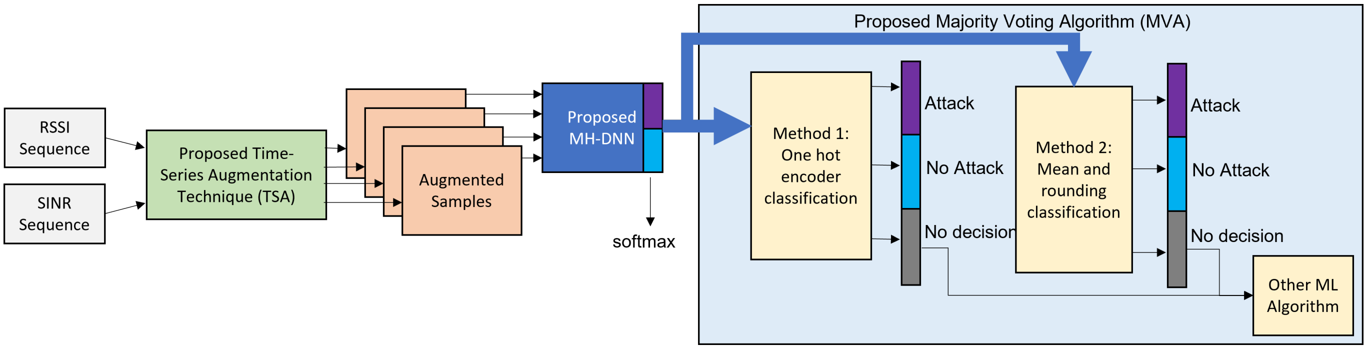

Proposed Majority Voting Algorithm (MVA)
RSSI Sequence
SINR Sequence
Proposed Time-Series Augmentation Technique (TSA)
Augmented Samples
Proposed MH-DNN
softmax
Method 1: One hot encoder classification
Attack
No Attack
No decision
Method 2: Mean and rounding classification
Attack
No Attack
No decision
Other ML Algorithm

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

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., $32 \times 16$). 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., $32 \times 16$). 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 $(4 \times 4)$ and $(8 \times 8)$ 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 $(16 \times 16)$ heads and keys. Therefore, based on the knee (elbow) rule, we choose the Attention configuration $(8 \times 8)$, 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 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** |

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 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

TABLE V: MH-DNN Configuration Parameters.

| Deep network Parameters | Values |
|---|---|
| Number of input heads | 2 |
| Base learning rate | $2.5 \times 10^{-2}$ |
| 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 | $1 \times 10^{-6}$ |
| L2 regularization for Dense and Attention layers | $1 \times 10^{-5}$ |

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.

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 |

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.

**Algorithm 1** Majority Voting Algorithm.

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

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 $\tau$ is the primary sample, and $Aug$ represents the four augmented samples for the $\tau$ 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 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

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 |

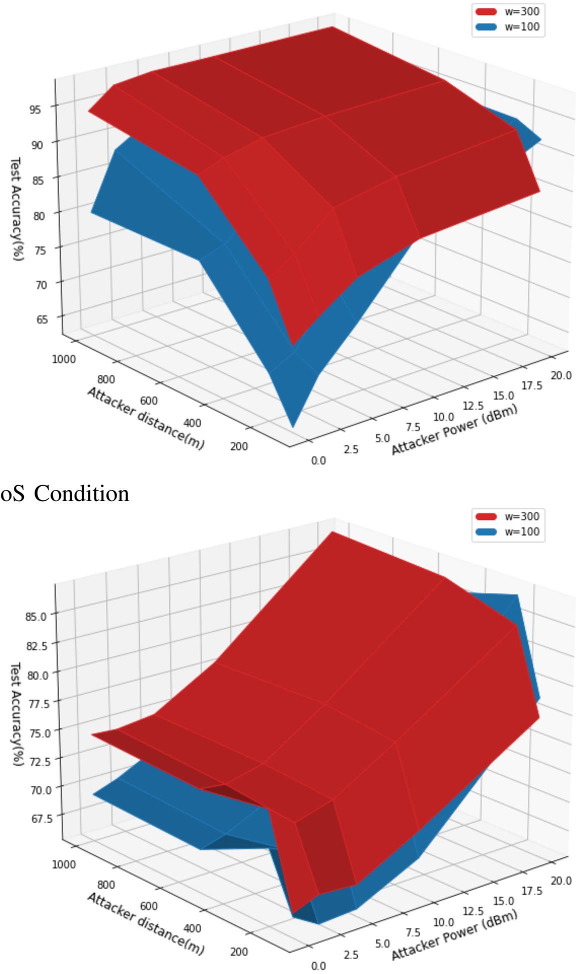
*URD - Uniformly Random Distributed.

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 paper: 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 (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.



(a) LoS Condition

(b) NLoS Condition

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

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

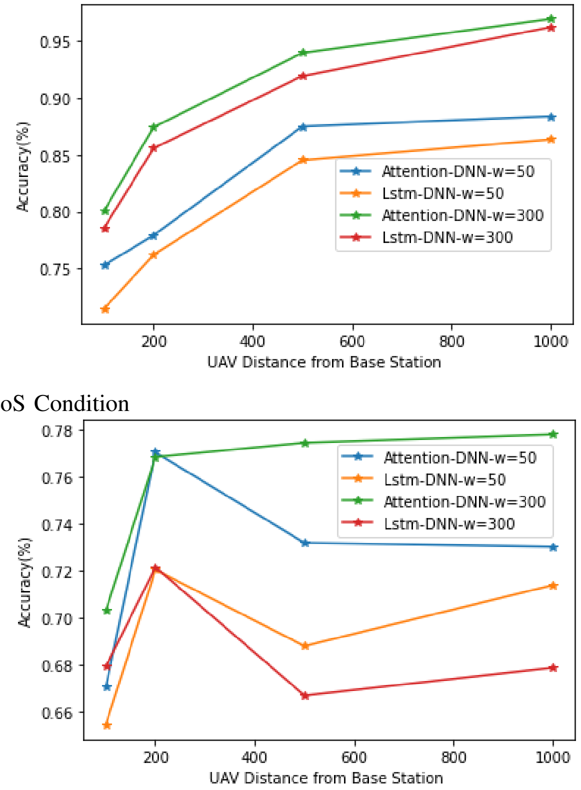
The trainable parameters do not change between the different window sizes or conditions. In our example, the MH-DNN configured with LSTM has 59,984 trainable parameters compared to 64,368 in the one with the Attention. However, most well-known deep neural networks, such as VGG [41] and ResNet [42], employ 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

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 |



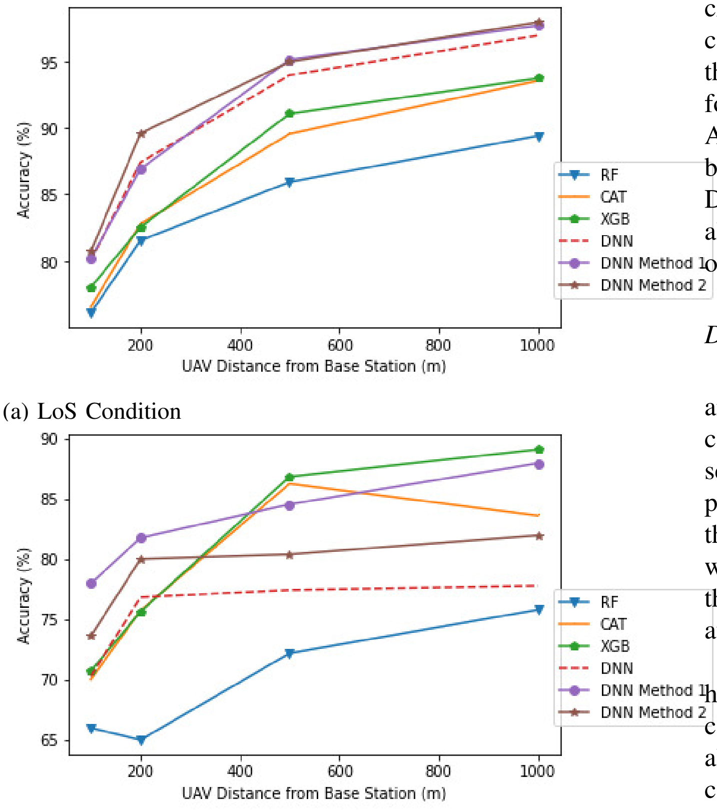
(a) LoS Condition

(b) NLoS Condition

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, (b) In NLoS.

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.



(a) LoS Condition

(b) 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, (b) In NLoS.

### 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 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

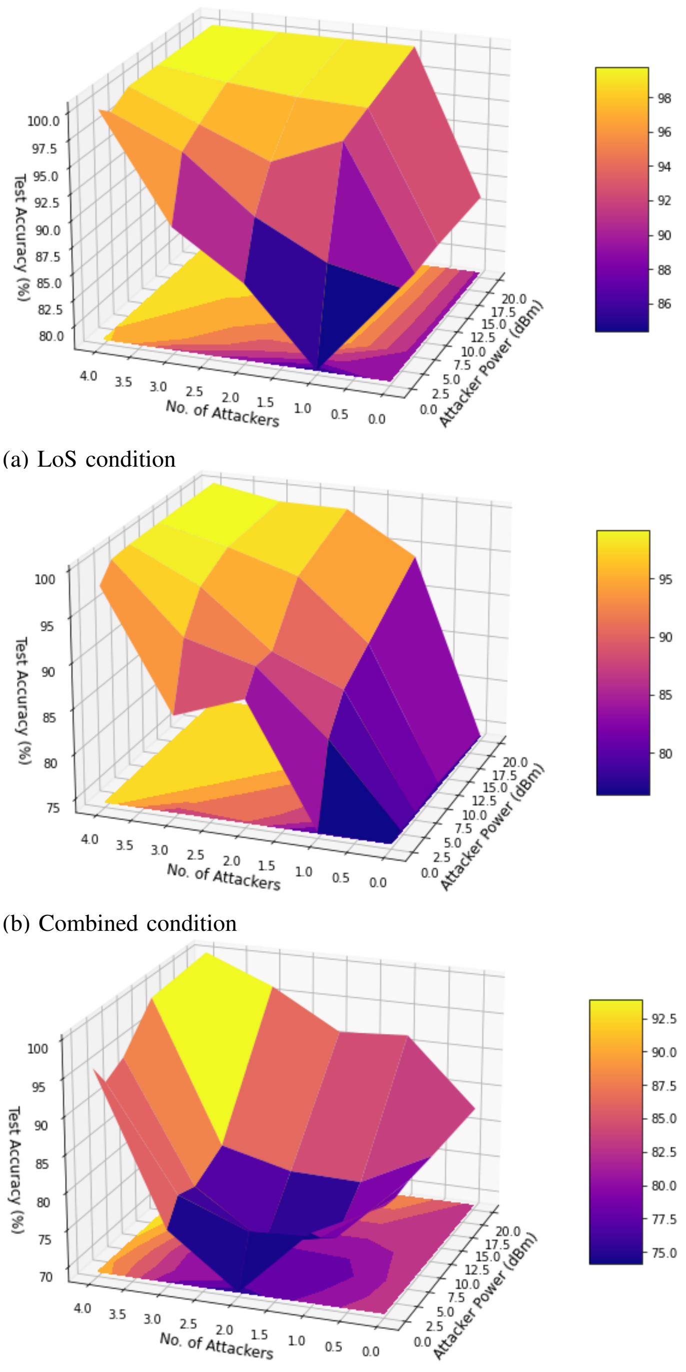
Fig. 8 presents the accuracy over the number of attackers and their power in (a) LoS, (b) Combined, and (c) NLoS conditions. If we look closely at the individual charts, we see that the accuracy increases with more attackers and more power for LoS and combined conditions. In the NLoS case, the low accuracy 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 for NLoS settings. For example, Fig. 9a 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

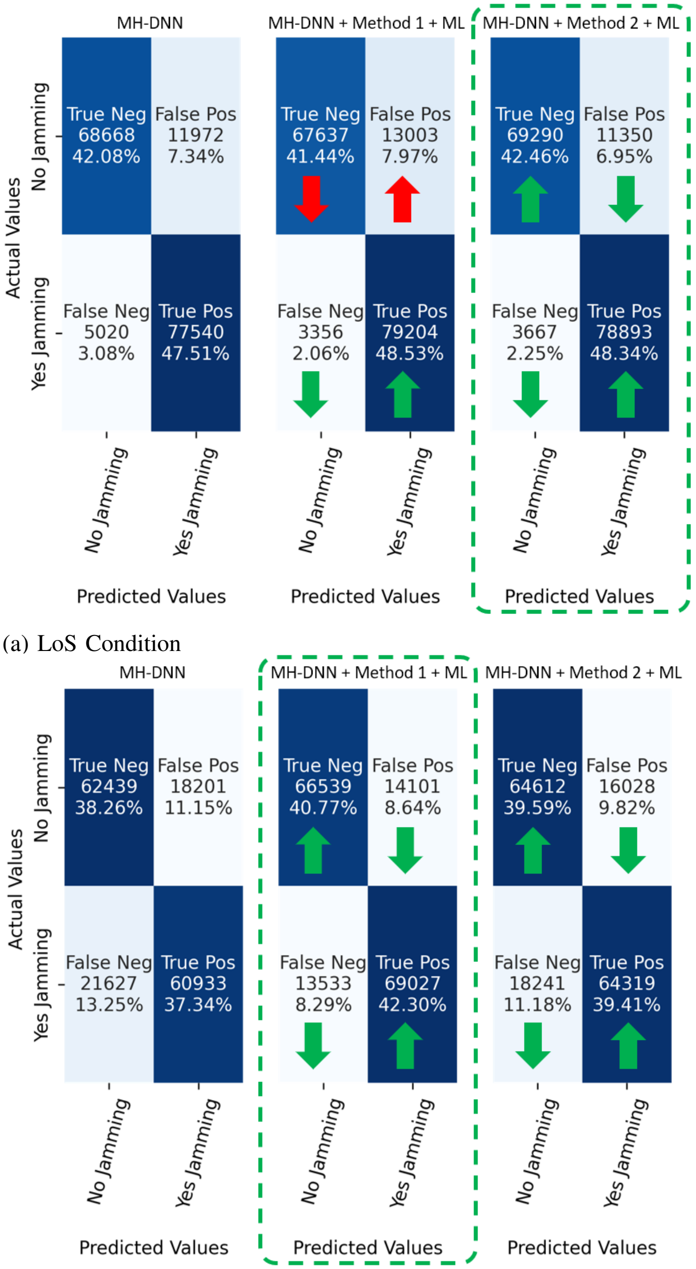


(a) LoS condition

(b) Combined condition

(c) NLoS condition

Fig. 8: Accuracy vs. Attackers Number and Attacker Power data during the test, $N_u = 20$, $d = 100$ m, $w = 300$, (a) LoS only, (b) LoS and NLoS, (c) NLoS only.

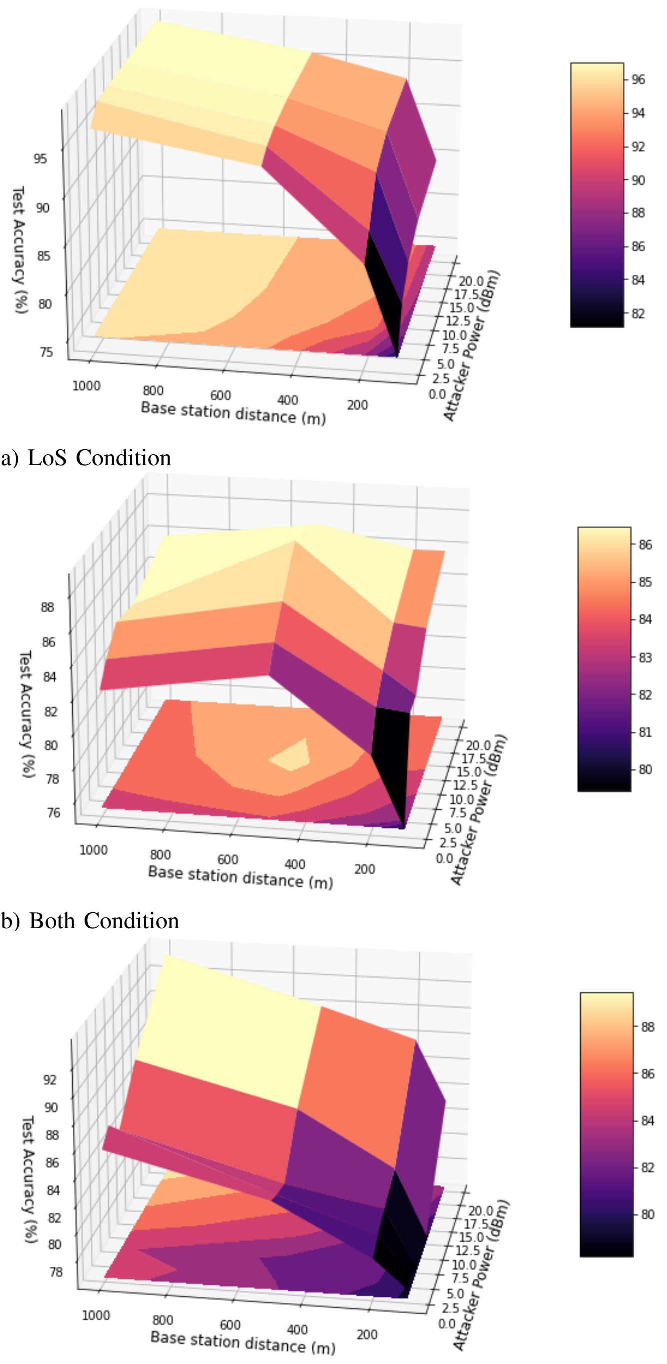


(a) LoS Condition

(b) NLoS Condition

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, (b) In NLoS. Green arrows indicate enhancement, while the red ones refer to reduction.

of `False Pos` by -0.39%. We see the opposite effect in Fig. 9b. 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 matrix is the average accuracy from all the scenarios in the dataset. It significantly impacts the specific cases, as shown in the following sections.



(a) LoS Condition

(b) Both Condition

(c) NLoS Condition

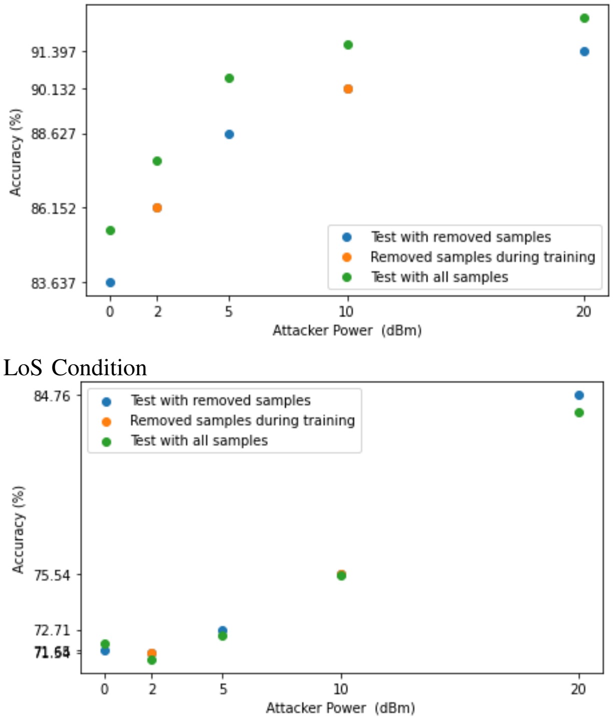
Fig. 10: Accuracy vs. Attackers Power and Attacker Distance test data, window size $w = 300$, $N_{att} = 2$, $N_u = 20$ (a) LOS only, (b) LOS and NLoS, (c) NLoS only.

### *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.



(a) LoS Condition

(b) NLoS Condition

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

Fig. 11a 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. 11b, 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.

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 $\mu$s | 30.9 ms ± 335$\mu$s | 31.9 ms ± 656$\mu$s | 30.8 ms ± 391$\mu$s |
| DNN-LSTM | 31.3 ms ± 1.03 ms | 31 ms ± 351 $\mu$s | 31.2 ms ± 311 $\mu$s | 30.5 ms ± 393 $\mu$s |
| CAT | **0.52** ms ± 561 ns | 0.82 ms ± 744 ns | 1.49 ms ± 939 ns | 2.19 ms ± 2.02 $\mu$s |
| 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 $\mu$s | **0.67** ms ± 22.5 $\mu$s | **0.68** ms ± 23.9 $\mu$s | **0.74** ms ± 21.6 $\mu$s |

### 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.

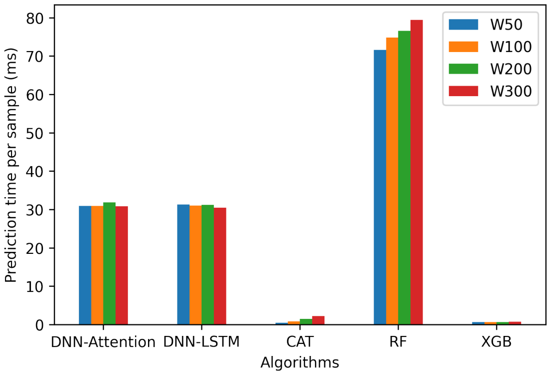


Fig. 12: Average processing time for each classifier.

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 paper 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.

## Acknowledgment

This research received funding from the European Union’s Horizon 2020 research and innovation program under the Marie Sklodowska-Curie Project Number 813391 and Fundação para a Ciência e a Tecnologia and Instituto de Telecomunicações under Project UIDB/50008/2020. In addition, CTTC authors have received funding from Grant PID2021-126431OB-I00 funded 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, and Generalitat de Catalunya grant 2021 SGR 00770.

## References

[1] Brena Kelly S. Lima et al. “Aerial Intelligent Reflecting Surfaces in MIMO-NOMA Networks: Fundamentals, Potential Achievements, and Challenges”. In: *IEEE Open Journal of the Communications Society* 3 (2022), pp. 1007–1024. DOI: 10.1109/OJCOMS.2022.3182223.

[2] 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.

[3] 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.

[4] Xiucheng Wang et al. “Joint Flying Relay Location and Routing Optimization for 6G UAVndash;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.

[5] 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.

[6] 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.

[7] 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.

[8] Mojtaba Vaezi et al. “Cellular, Wide-Area, and Non-Terrestrial IoT: A Survey on 5G Advances and the Road Toward 6G”. In: *IEEE Communications Surveys and Tutorials* 24.2 (2022), pp. 1117–1174. DOI: 10.1109/COMST.2022.3151028.

[9] Bin Li et al. “3D Trajectory Optimization for Energy-Efficient UAV Communication: A Control Design Perspective”. In: *IEEE Transactions on Wireless Communications* 21.6 (2022), pp. 4579–4593. DOI: 10.1109/TWC.2021.3131384.

[10] Bin Li et al. “A Hybrid Offline Optimization Method for Reconfiguration of Multi-UAV Formations”. In: *IEEE Transactions on Aerospace and Electronic Systems* 57.1 (2021), pp. 506–520. DOI: 10.1109/TAES.2020.3024427.

[11] Kok Lay Teo et al. *Applied and Computational Optimal Control*. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-69913-0. URL: https://doi.org/10.1007/978-3-030-69913-0.

[12] 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.

[13] 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.

[14] Nicolas Souli, Panayiotis Kolios, and Georgios Ellinas. “An Autonomous Counter-Drone System with Jamming and Relative Positioning Capabilities”. In: *ICC 2022 - IEEE International Conference on Communications*. 2022, pp. 5110–5115. DOI: 10.1109/ICC45855.2022.9838783.

[15] 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.

[16] Omid Sharifi-Tehrani, Mohamad F. Sabahi, and M.R. Danaee. “GNSS jamming detection of UAV ground control station using random matrix theory”. In: *ICT Express* 7.2 (2021), pp. 239–243. ISSN: 2405-9595. DOI: https://doi.org/10.1016/j.icte.2020.10.001. URL: https://www.sciencedirect.com/science/article/pii/S2405959520303040.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] Ali Krayani et al. “Automatic Jamming Signal Classification in Cognitive UAV Radios”. In: *IEEE Transactions on Vehicular Technology* (2022), pp. 1–17. DOI: 10.1109/TVT.2022.3199038.

[23] 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.

[24] Menaka Pushpa 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. DOI: 10.1109/CITS.2019.8862148.

[25] 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.

[26] 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. DOI: 10.1007/s10618-019-00619-1.

[27] Ashish Vaswani et al. “Attention is All you Need”. In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 6000–6010. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.

[28] Bendong Zhao et al. “Convolutional neural networks for time series classification”. In: *Journal of Systems Engineering and Electronics* 28.1 (2017), pp. 162–169. DOI: 10.21629/JSEE.2017.01.18.

[29] 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.

[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] Jiannan Gao et al. “DRFM Jamming Mode Identification Leveraging Deep Neural Networks”. In: *2021 International Conference on Control, Automation and Information Sciences (ICCAIS)*. 2021, pp. 444–449. DOI: 10.1109/ICCAIS52680.2021.9624526.

[32] 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.

[33] *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.

[34] *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.

[35] L. F. HENDERSON. “The Statistics of Crowd Fluids”. In: *Nature* 229.5284 (1971), pp. 381–383. DOI: 10.1038/229381a0. URL: https://doi.org/10.1038%2F229381a0.

[36] Joseanne Viana 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)*. 2022, pp. 1–5. DOI: 10.1109/VTC2022-Fall57202.2022.10012726.

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

[38] Sebastian Ruder. *Deep Learning for NLP Best Practices*. http://ruder.io/deep-learning-nlp-best-practices/. 2017.

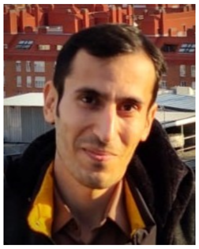
[39] Noam Levi et al. *Noise Injection Node Regularization for Robust Learning*. 2022. DOI: 10.48550/ARXIV.2210.15764.

[40] Tong He et al. “Bag of Tricks for Image Classification with Convolutional Neural Networks”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 558–567. DOI: 10.1109/CVPR.2019.00065.

[41] Keras. *VGG16 and VGG19*. 2022. URL: https://keras.io/api/applications/vgg/ (visited on 09/30/2022).

[42] Keras. *ResNet and ResNetV2*. 2022. URL: https://keras.io/api/applications/resnet/ (visited on 09/30/2022).

**Joseanne Viana** is a Ph.D. candidate at ISCTE - Lisbon University Institute in the Radio Systems Department. 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 in the frame of (MSCA ITN) of the European Commission's Horizon 2020. Her research interests are wireless communications applied to interconnected systems such as UAVs, aerial vehicles, and non-terrestrial devices.



**Hamed Farkhari** is a Ph.D. candidate at ISCTE - Lisbon University Institute. He is an Early-Stage Researcher in the TeamUp5G group, a European Training Network in the frame of (MSCA ITN) of the European Commission's Horizon 2020. Also, he is a researcher and developer at PDMFC Co. He is interested and working on cybersecurity, machine learning, deep learning, data science, meta-heuristic, and optimization algorithms.

**Pedro Sebastião** received a 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. He is the author or co-author of more than 200 scientific articles, and he has been responsible for several national and international Research and Development projects. He has been an expert and evaluator of more than one hundred national and international Civil and Defense Research and Development projects. 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 the creation of startups and spinoffs of value to society and the 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 research interests are in monitoring, control, and communications of drones, unmanned vehicles, planning tools, stochastic processes (modeling and efficient simulations), the Internet of Things, and efficient communication systems.

**Katerina (Aikaterini) Koutlia** received her B.Sc. in Electronics Engineering (2009) from the Technological Institution of Thessaloniki (Greece) and her M.Sc. with distinction (2011) in Wireless Communication Systems from the Brunel University (Uxbridge, UK). In 2016 she obtained her Ph.D. with honors (supported by a grant from the Spanish Ministry of Education, Culture, and Sport) from the Polytechnic University of Catalonia (UPC). She has worked as a Post-Doctoral Researcher in the Mobile Communication Research Group (GRCM) at UPC, where she has been involved in several European and National Projects. In 2018 she joined CTTC, where she is currently employed as 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.

**Luis Miguel Campos** received the B.Tech. degree from the Instituto Superior Técnico (IST), Lisbon, in 1992, the M.S. degree in information and computer science from the University of California at Irvine, Irvine, in 1995, and the Ph.D. degree in information and computer science in 1999. He worked as a Faculty Member at the University of California at Irvine, an Intern with NASA, a Researcher with INESC, and a Teaching Assistant with the Instituto Superior Técnico. 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 serves as an Expert Evaluator for the European Commission, leads the Business Angel Fund SMENT Digital, and serves on the board of over ten companies. 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. He is also the Managing Director of Koala Tech and responsible for worldwide investment funding. He is involved in 12 European-funded research projects (Horizon2020) and five national research projects (Portugal2020). He has published dozens of papers at international conferences in areas as diverse as parallel computing, agent-based computing, resource management in distributed systems, simulation theory, cluster computing and grid computing, computer vision, information systems, and e-Government. Some companies have received the Prestigious Award Deloitte Technology Fast 500, namely Go4Mobility, and some have been selected by the Portuguese state as one of the most innovative companies in the country, namely PDMFC.

**Biljana Bojović** received her MSc in Electrical and Computer Engineering from the Faculty of Technical Sciences, Novi Sad, Serbia, in 2008 and her Ph.D. 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 at CONFTELE in 2021. In addition, she was a mentor of ns-3 GSoC on several occasions. In 2020 she received ACM SIGCOMM Networking System Award. She worked on many research projects for industrial clients, such as Wi-Fi Alliance, SpiderCloud, Interdigital, the US Department of Defense, NIST, Facebook, etc. She is a 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 her Telecommunications Engineering, M.S., and Ph.D. degrees from Universitat Politècnica de Catalunya (UPC), Spain, in 2011, 2013, and 2016, respectively. COIT awarded her dissertation the best national Ph.D. thesis on high-speed broadband mobile communications (2017) and received a Special Doctoral Award from UPC (2019). In 2017, she joined CTTC, 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. She is a 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. Her research interests include wireless communications, spectrum and interference management, and optimization theory.

**Rui Dinis** (Senior Member, IEEE) received a Ph.D. degree from the Instituto Superior Técnico (IST), Technical University of Lisbon, Portugal, in 2001, and a 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 Análisee 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 was a Researcher 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 an Associate Professor with FCT, Universidade Nova de Lisboa (UNL). He has been actively involved in several national and international research projects in the broadband wireless communications area. His research interests include transmission, estimation, and detection techniques. Prof. Dinis is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE OPEN JOURNAL ON COMMUNICATIONS, and Physical Communication (Elsevier). He was also a Guest Editor of Physical Communication (Elsevier) (Special Issue on Broadband Single-Carrier Transmission Techniques). He is a VTS Distinguished Lecturer.