



Performance analysis of capsule networks for detecting GPS spoofing attacks on unmanned aerial vehicles

Tala Talaei Khoei¹ · Khair Al Shamaileh² · Vijaya Kumar Devabhaktuni³ · Naima Kaabouch⁴

© The Author(s) 2025

Abstract

Unmanned aerial vehicles (UAVs) are prone to several cyber-attacks, including global positioning system (GPS) spoofing. The use of machine learning and deep learning are becoming increasingly common for UAV GPS spoofing attack detection; however, these approaches have some limitations, such as a high rate of false alarm and misdetection. We propose using capsule networks to detect and classify UAV-focused GPS spoofing attacks. This paper compares simple capsule networks, efficient capsule networks, dual attention capsule networks, and convolutional neural network in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, prediction time, training time per sample, and memory size. The results indicate that the Efficient-capsule network outperforms the other models, as demonstrated by an accuracy of 99.1%, a probability of detection of 99.9%, a probability of misdetection of 0.1%, a probability of false alarm of 0.37%, a prediction time of 0.5 seconds, a training time per sample of 0.2 seconds, and a memory size of 123 mebibytes for binary classification.

Keywords Capsule network · Convolutional neural network (CNN) · Deep learning · Global positioning system (GPS) · GPS spoofing attacks · Machine learning · Unmanned aerial vehicle (UAV)

1 Introduction

Unmanned aerial vehicles (UAVs) are military and civilian aircraft designed to conduct many operations, including inspections, surveillance, reconnaissance, shooting, agricultural inspections, and rescue operations. UAVs use global positioning systems (GPS) for positioning and navigation. Civilian GPS signals are not encrypted, which creates several security risks. GPS systems are particularly prone to multiple cyber-attacks and threats, including spoofing, mimicking, and jamming. Attackers send fake GPS signals containing incorrect times, positions, and navigation information, result-

ing in serious navigation changes, and can be used to hijack a UAV, steal the onboard data, and crash it into populated areas [1, 2]. Recent security incidents related to attacks on UAV-based GPS devices have threatened human lives during wars in the Ukraine, Russia, Iran, and Iraq [3].

Numerous techniques have been proposed to classify, detect, and mitigate UAV-focused GPS spoofing. These techniques are divided into three categories: (1) hardware-based, (2) signal processing-based, and (3) machine learning (ML) and deep learning (DL)-based techniques [4]. The first category includes spatial and geometrical UAV characteristics that impact onboard sensors such as compasses, barometers, and inertial measurement units. The accuracy of these techniques requires additional hardware, such as high-quality compact sensors and continuous sensor calibration. These costly sensors are not acceptable solutions for GPS spoofing detection in small UAVs [5–7]. The second category uses vision-based methods requiring intensive signal processing, potentially impacting real-time system performance and additional communication overhead. These methods are not effective if an attacker introduces a time delay or drifts into the spoofing signals and if an attacker does not know the receiver's actual location.

✉ Tala Talaei Khoei
t.talaeikhoei@northeastern.edu

¹ Khoury College of Computer Sciences, Roux Institute at Northeastern University, Portland, ME 04101, USA

² Electrical and Computer Engineering Department, Purdue University Northwest, Hammond, IN 46323, USA

³ Electrical and Computer Engineering Department, Illinois State University, Normal, IL 61761, USA

⁴ School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA

The third category utilizes machine learning and deep learning models to detect UAV-focused GPS spoofing. Existing studies used these models to detect GPS spoofing attacks; however, machine learning models suffer from several limitations, such as overfitting, high rates of false alarms, and misdetection [8]. Some studies have used deep learning models, such as long short-term memory, residual neural networks, and artificial neural networks, as a result. Convolutional neural networks (CNNs) are commonly used deep learning models that have been investigated. CNN models can learn information through a supervised approach by using convolution operations, pooling layers, and SoftMax functions. These models can automatically extract numerous invariant and discriminative features for one- and two-dimensional data. The key characteristic of these models is their ability to replicate the same knowledge at all points in an input dataset's spatial dimension; therefore, the features at one spatial location using replicas of feature detectors are available at other locations. The CNN models have local shared connectivity that is connected to layers of spatial reduction, such as max-pooling, and local translation-invariant features, which result in routing low-level features between layers using max-pooling.

CNN algorithms have several shortcomings. For example, these algorithms are significantly slower than other DL models due to the max-pooling layers' performance. These algorithms also have a longer training process and require large datasets for processing, training, testing, and validation. A CNN-based algorithm, capsule network (CapsNet), was recently proposed to address these issues. This model consists of a group of capsules with each neuron's output representing a different feature property. The capsules process the given features at their inputs and encapsulate the results into a vector of highly informative outputs. A capsule is a replacement for artificial neurons. One difference is that artificial neurons handle scalars while the capsules handle vectors. The length of an activity vector is defined as the probability of an entity, and its orientation is an instantiation parameter. Active capsules make predictions through transformation metrics for instantiation parameters of higher-level capsules. A higher-level capsule change to an active capsule once several predictions are completed [9, 10].

We investigated the performance of three new CapsNet algorithms: conventional CapsNet, efficient-CapsNet, and dual attention-CapsNet (DA-CapsNet). We then compared these results with those of a conventional CNN models for detecting UAV-focused GPS spoofing attacks. The evaluation was performed in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, prediction time, training time per sample, and memory size. To summarize, the main contributions of this study are listed, as follows:

- Introducing three CNN-based algorithms, part of the Capsule Family, addressing the issues of CNN models,
- Developing conventional CapsNet, efficient-CapsNet, and dual attention-CapsNet to detect and classify GPS spoofing attacks on UAVs,
- Evaluating these models in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, prediction time, training time per sample, and memory size,
- Providing a comprehensive comparison between these models and other proposed models in the literature with respect to the used data.

This paper is organized as follows: Section 2 presents the related work, whereas Sect. 3 discusses the corresponding dataset, data pre-processing techniques, training process, and evaluation metrics. The results of the study are presented and analyzed in Sect. 4. The conclusion is outlined in Sect. 5.

2 Related work

Numerous techniques have been proposed to classify, detect, and mitigate GPS spoofing attacks. These techniques are divided into three categories: (1) hardware-based, (2) signal processing-based, and (3) machine learning (ML) and deep learning (DL)-based techniques [4]. The first category includes techniques based on spatial and geometrical UAV characteristics, which impact onboard sensors such as compasses, barometers, and inertial measurement units. To be efficient, these techniques require additional hardware, such as high-quality compact sensors and continuous sensor calibration. These costly sensors are not acceptable solutions for GPS spoofing detection in small UAVs [5–7]. The second category uses vision-based methods requiring intensive signal processing, potentially impacting real-time system performance and additional communication overhead. These methods are not effective if an attacker introduces a time delay or drifts into the spoofing signals and if an attacker does not know the receiver's actual location.

The third category utilizes machine learning and deep learning models to detect UAV-focused GPS spoofing. Existing studies used these models to detect GPS spoofing attacks; however, machine learning models suffer from several limitations, such as overfitting, high rates of false alarms, and misdetection [8]. Some studies have used deep learning models, such as long short-term memory, residual neural networks, and artificial neural networks, as a result. Convolutional neural networks (CNNs) have also been investigated. These models can learn information through a supervised approach by using convolution operations, pooling layers, and SoftMax functions. They can automatically extract numerous invariant and discriminative features for one- and

two-dimensional data. The key characteristic of these models is their ability to replicate the same knowledge at all points in an input dataset's spatial dimension; therefore, the features at one spatial location using replicas of feature detectors are available at other locations. These models have local shared connectivity that is connected to layers of spatial reduction, such as max-pooling and local translation-invariant features, which result in routing low-level features between layers using max-pooling.

Table 1 provides a short list of techniques that have been proposed to detect and classify UAV-focused GPS spoofing attacks. These studies are divided according to these categories. For instance, the authors of [11] used a hardware-based technique to detect GPS spoofing using an inertial measurement unit (IMU). This technique depends heavily on acceleration error, which is calculated by comparing the GPS data to the IMU data. The authors of [12] introduced a vision-based approach for detecting GPS spoofing attacks using monocular and IMU sensors. The UAV's velocity was computed using the onboard sensors. This velocity was compared to the velocity calculated using the Lucas Kanade approach. The UAV was spoofed if the value of the root means square errors from these two approaches were different. Another vision-based approach was proposed in [13]. A UAV trajectory was determined using visual odometry since fake GPS signals do not alter images. The GPS flight trajectory was compared to the UAV's trajectory to detect fake signals. This approach was effective for long-distance UAV flight scenarios when the direction was changed by more than 3°.

Many studies have used ML and DL models to detect and classify UAV-focused GPS spoofing attacks. For instance, the authors of [2] proposed using artificial neural networks. Authentic GPS signals and simulated attacks were collected to implement the training and testing dataset. Another study introduced a genetic algorithm with an extreme boosting model [14]. The model was pre-trained offboard using the flight logs to decrease power consumption and hardware resources during long-term operations, then the genetic algorithm was used to optimize the developed model. The authors of [15] compared the performance of several tree-based ML models for detecting UAV-focused GPS spoofing, including gradient boosting, light gradient boosting, and extreme gradient boosting. A dataset with 13 features that included spoofed and non-spoofed signals was used for training. The authors of [16, 17] developed some common machine learning models, such as support vector machine, to detect UAV-focused GPS spoofing attacks.

The authors of [18] compared various conventional instance-based ML models to detect and classify UAV-focused GPS spoofing attacks, including linear support vector machine, numerical support vector machine, C-support vector machine, K nearest neighbor, and radius neighbor. The authors used a correlation-based technique, Spearman

Correlation Coefficient, to reduce dataset dimensionality, training complexity, and time. The authors of [19] evaluated three different ensemble models: stacking, bagging, and boosting. The results indicated that these models outperformed other traditional machine learning models, such as support vector machine, when detecting UAV-focused GPS spoofing attacks. Other papers have proposed using DL-based techniques to detect UAV-focused GPS spoofing attacks. For instance, the authors of [20] proposed an approach, long short-term memory, that can effectively detect GPS spoofing attacks within five seconds. The authors of [21] proposed a residual neural network to detect GPS spoofed signals using images. The historical satellite images were compared to real-time camera images, and the detection was performed using a threshold of the similarity between satellite imagery and aerial photography.

3 GPS spoofing detection

The proposed detection architecture consists of several phases (Fig. 1), including data acquisition, pre-processing, training, detection, and classification. Real-time experiments and simulations were performed during data acquisition to collect authentic signals and GPS spoofing attacks [15]. The corresponding dataset was then pre-processed by applying several techniques, including data transformation, class balancing, and data encoding. The models were trained with the dataset, and their performance was validated with the testing dataset.

3.1 Dataset

The dataset used in this work was created during a previous work [18]. This dataset consists of authentic and spoofed signals. The corresponding data consists of three types of GPS spoofing attacks: simplistic, intermediate, and sophisticated. Thirteen features were identified and extracted from the raw signals. In simplistic attacks, the spoofer generates fake GPS signals, unsynchronized with normal signals. In this attack, the attacker does not have any knowledge about the receiver's position, resulting in a high Doppler Shift. Therefore, a huge deviation can occur in the pseudo-range measurement. In addition, the attacker transmits the spoofed signals at high-power levels, leading to higher carrier to noise levels.

In intermediate spoofing, the attacker can control the UAV by precisely handling the GPS-generated signals. Here, the attacker knows the target position, resulting in code phase alignment between the real and spoofed transmissions. In contrast to simplistic spoofing, the Doppler shift and pseudo range resulting from the intermediate spoofing are usually within normal ranges. In addition, other features, such as time of the week, carrier phase shift, and correlator amplitude,

Table 1 Related works on detecting UAV GPS spoofing attacks

Category	Method	Used Data Set	Use Metrics	Study highlights and Evaluation
Hardware-Based Techniques	IMU-Based [11]	GPS receiver and Accelerator Data	Detection, Minimum Acceleration,	Provides magnitude acceleration error for high-performance results. Achieves the best detection when both moving acceleration and spoofing acceleration are heading within roughly 25° from the east or west.
Signal Processing	Vision-Based [12]	monocular camera and Inertial Measurement Unit sensor of UAV	Root Mean Squared Error values	Detects GPS spoofing attacks based on multiple model characteristics with an average of 5 seconds. Detects GPS spoofing at 2021ms (about 2 seconds) on x-axis and at 1702ms (about 17 seconds) on y-axis.
	Vision-Based [13]	Images of UAV photograph Project and spoofing simulations	Measures of Dissimilarity at different Window Size.	Detects GPS spoofing attacks for long-distance UAV flights when the changes in flight direction are greater than 3° . Provides good results when no redirection and velocity of the UAV was changed by Sum of Euclidian Distances between Corresponding Points.
ML and DL Models	Artificial Neural Networks [2]	Self-collected signals, including 5 features	Accuracy, Detection, Misdetction, False Alarm.	Provides good performance results. Compares the efficiency of one- and two hidden-layer neural networks with various numbers of hidden neurons.

Table 1 continued

Category	Method	Used Data Set	Use Metrics	Study highlights and Evaluation
	Genetic Algorithm and Extreme Boosting [14]	GPS and Inertial Measurement Unit data	Detection Rate, Correctness Ratio,	Provides a detection correctness of 96.3% in hijacked scenarios, Obtains the detection correctness in non-hijacked scenarios.
	Tree-Based Models [15]	Self-collected signals, including 13 features	Detection Delay Accuracy,	Extreme Gradient Boosting gives the best results compared to tree-based models, such as Gradient Boosting, Light Gradient Boosting, and Random Forest,
	Support Vector Machine [16]	Self-collected signals, including 11 features	Detection, Misdetection False Alarm, Processing Time in training and testing. Memory Size in training and testing. Accuracy	Provides an accuracy of 98.72% to detect potential GNSS signal manipulation attempts using C-Support Vector Machine.
	Support Vector Machine [17]	UAV autopilot Data	Gives acceptable results when using manometer sensors.	
	False positive,	Due to no knowledge of the real trajectory, any GPS spoofing attack is detected.	True positive, Accuracy	

Table 1 continued

Category	Method	Used Data Set	Use Metrics	Study highlights and Evaluation
	Instance Models [18]	Self-collected signals, including 13 features	Accuracy, Detection, Misdetction, False Alarm, Processing Time Memory Size in training and testing.	Nu-Support Vector Machine outperforms tother learning Instance-based models.
	Ensemble Models [19]	Self-collected signals, including 13 features	Accuracy, Detection, Misdetction, False Alarm, Memory Size, Processing Time, Average prediction Time Per Sample.	Stacking model provides the best performance compared to the Bagging and Boosting models.
	Long Short-Term Memory [20]	Self-collected signals	Detection Rate	Detects GPS spoofing attacks with no need for upgrading existing equipment with a detection rate of 78% in Time Cost of 3 seconds.
	Residual Neural Network [21]	Self-Collected Images	Time Cost Accuracy Precision, Recall, Error, F1 -Score	Proposes four visual image anti-spoofing models with good results.

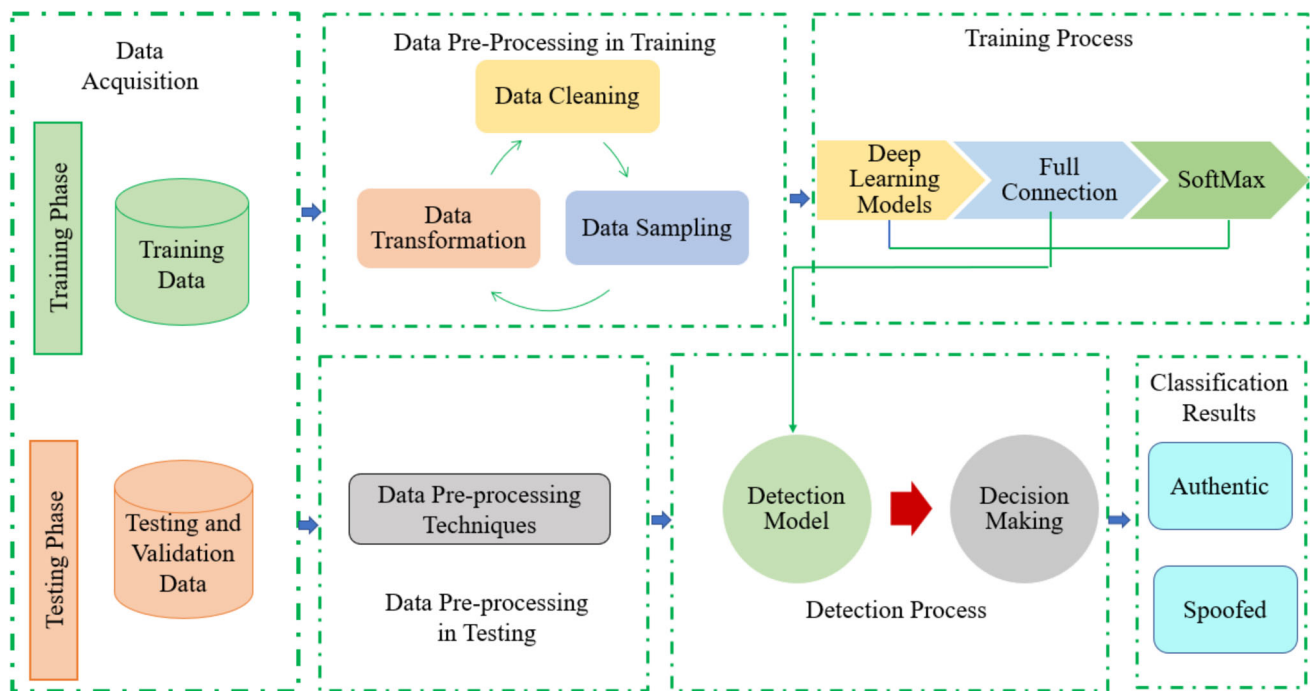


Fig. 1 Architecture of the Proposed System

Table 2 List of features

Used features	Abbreviations
Pseudorandom number	PRN
Carrier doppler	DO
Pseudo range	PD
Receiver time	RX
Time of week	TOW
Prompt In-Phase component	PIP
Prompt quadrature component	PQP
Tracking carrier doppler	TCD
Carrier to noise	C/N0
Carrier phase shift	CP
Prompt correlator	PC
Early correlator	EC
Late correlator	LC

vary abruptly, potentially indicating a presence of the attack. Finally, a sophisticated attack applies several synchronized antennas to emulate the GPS constellation. Thus, the attacker can spoof various channels simultaneously and can take full control of the UAV. The list of features in the given dataset is depicted in Table I. This dataset contains 36,459 simplistic attacks, 32,013 intermediate attacks, 44,232 sophisticated attacks, and 397,826 legitimate samples.

Table 3 Number of cases per class

	Simplistic	Intermediate	Sophisticated	Normal
Training	25,611	25,611	25,611	76,833
Testing	3,201	3,201	3,201	9,603
Validation	3,201	3,201	3,201	9,603

3.2 Data pre-processing

Several techniques, including class balancing, data transformation, and data encoding, were used to pre-process the dataset. The first step was to balance the classes in the given dataset. The number of attacks and authentic signals were not equally distributed. We included 32,013 samples from each set of attack and authentic signals to ensure class balance in the dataset, resulting in a total of 192,078 samples. A DL-based approach tends to be biased toward the majority classes, or the scenario with the largest samples, leading to inaccurate results. The classes were balanced using 10% of the data for validation and 10% for testing to address these issues, whereas the remaining data were used for training purposes. Table 3 provides an overview of the number of different authentic and spoofed signals used for training, testing, and validation.

We used a simple standardization technique for data transformation. This technique converts the mean value of the numerical data to zero and standard deviation to one. The authentic signals are encoded to 0, simplistic attacks to 1,

intermediate attacks to 2, and sophisticated attacks to 3 during data encoding.

3.3 DL models

CNN models have become increasingly popular in many fields, including security. The basic CNN architecture is illustrated in Fig. 2a. This architecture typically consists of several convolutional layers, pooling layers, and fully connected layers. A primary advantage of these networks is the low number of parameters compared to other types of neural networks, such as ANNs. Another important characteristic of CNNs is that they can extract abstract information when their input data grows into deeper layers. The CNN models provide several benefits; however, they have some significant limitations, such as slow performance and long training time [21–23].

A new CNN-based model, CapsNet, was proposed to solve these limitations. This architecture is a novel type of neural network that uses a vector-in and vector-out to transmit information. The typical information unit in a capsule network is a vectorized capsule that consists of multiple scalars, unlike traditional neural networks that are embedded with scalar neurons. Each capsule has different features with characteristics. The module length of a capsule also provides special meaning, such as probability of feature existence. CapsNet has three important layers: convolutional layer, primary capsule layer, and digit capsule layer (Fig. 2b). This model also consists of a function, attention, primarily used to find the data with the most important and relevant information, helping the network focus on specific parts of the data rather than the entire dataset. The attention function can help distinguish the best features corresponding to the target variables. CapsNet can also transfer the features using vectorized methods. These methods may lead to computational overhead, allowing the CapsNet architecture to work effectively with sufficient features, which highly depends on the model's first few neural layers [24, 25].

Little attention has been focused on CapsNet's efficiency and its ability to present knowledge transformations, despite the benefits of CapsNet models over CNN models. The existing solutions for classification problems using CapsNet consist of many parameters, which automatically hide the generalization ability of the capsules. Efficient-CapsNet has been proposed to reduce the number of CapsNet parameters and improve capsule efficiency. The Efficient CapsNet architecture is divided into three Sections. Sections 1 and 2 are the necessary parts of the capsule layers that interact with the input space (Fig. 2c). The self-attention function permits the capsules to communicate with other capsules, known as self-capsules, and discover the capsules that need more attention. The output is the aggregation of these communication and attention scores. The capsule's final layer does not provide the probability of a particular class; however, it returns

the probabilities extracted from its individual Sections [26]. Another CapsNet model, DA-CapsNet (Fig. 2d) was proposed to investigate the importance of the attention function in the performance of the CapsNet family. DA-CapsNet is significantly different than the other capsule families, such as CapsNet and Efficient-CapsNet. DA-CapsNet has two layers of attention mechanisms. These two layers are known as Convolutional Attention in ReLU Convolution to PrimaryCaps and Caps Attention in PrimaryCaps to Digit-Caps. The main reason for adding two layers of attention in DA-CapsNet is to improve the extraction of essential information in the capsules, decrease the transforming of the non-essential information, improve the contribution of the important information into capsules, and improve the hierarchy of the capsules [27].

3.4 Evaluation metrics

To evaluate and compare the performance of the proposed model to existing models, the following metrics are used:

- Accuracy (ACC): represents the probability of correct predictions to all predictions, and is given by:

$$ACC = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \times 100 \quad (1)$$

- Probability of Detection (p_d): denotes the probability of correctly classified spoofed signals over the total number of spoofed signals, and is given by:

$$p_d = \frac{T_p}{T_p + F_N} \times 100 \quad (2)$$

- Probability of Misdetection (p_{md}): features the probability of the spoofed signals incorrectly classified as authentic signals over the total number of spoofed signals, and is given by:

$$p_{md} = \frac{F_N}{T_P + F_N} \times 100 \quad (3)$$

- Probability of False Alarm (p_{fa}): denotes the probability of the authentic signals classified incorrectly as spoofed signals over the total number of authentic signals, and is given by:

$$p_{fa} = \frac{F_P}{T_F + F_N} \times 100 \quad (4)$$

where T_P stands for the true positive, T_N defines as the true positive, F_P represents the false positive, and F_N denotes the false negative. In addition, the efficiency of the models is

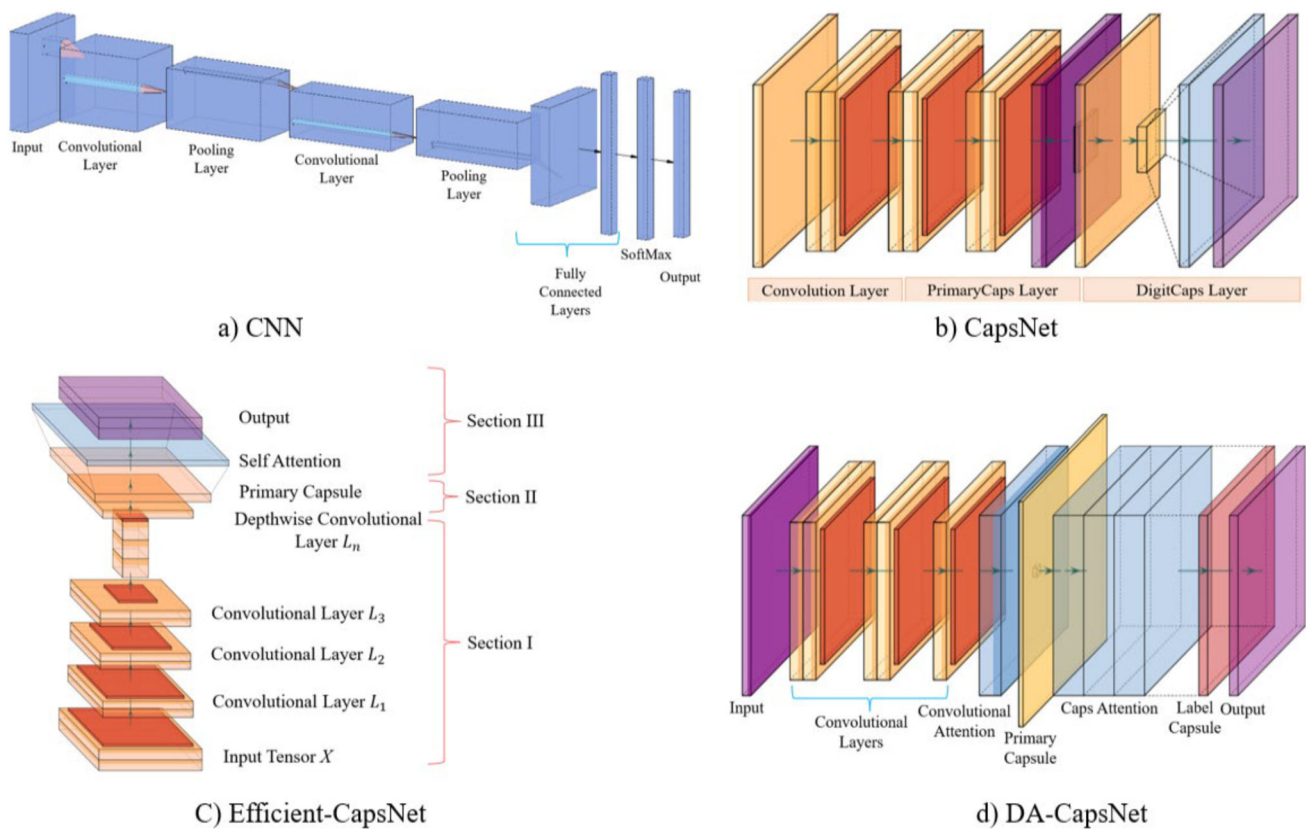


Fig. 2 Architecture of the selected models

evaluated in terms of prediction time (PT), training per sample (TPS), and memory size (M). These metrics are defined as follows:

- PT: time that a model uses to predict malicious signals.
- TPS: time required to process a sample during model training.
- M: size of the memory that a model uses during the whole deep learning process.

In this study, the confusion matrix is used to visualize the results of binary and multi-classification scenarios. Figure 3 shows the structures of such matrices for these scenarios. The values in these matrices are used to calculate the ACC, p_d , p_{md} , p_{fa} , based on (1)–(4). As one can observe, Fig. 3a has labels of positive and negative, and the result of predicted class is indicated as T_P , F_P , T_N , and F_N . However, in Fig. 3b, confusion matrix in multiclassification scenario consists of N different classes. In this matrix, the characterization of T_P , F_P , T_N , and F_N samples is not applicable. In this case, it is practical to perform an analysis, focusing on particular class based on their provided labels [28].

4 Results

Four neural network models were implemented: CapsNet, Efficient-CapsNet, DA-CapsNet, and CNN. These models were trained using 10-fold cross validation for 50 epochs per fold with a batch size of 50 and tested using the adaptive moment estimation (ADAM) optimizer, with a learning rate of 0.01. The results were obtained using a $\beta_{1,2} = 0.96$. Here, β_1 is the decay rate of the first moment and sum of the gradient, while β_2 is the decay rate for the second moment and sum of the gradient squared. It is noteworthy to point out that a 1×10^{-5} decay is used during training and testing. The dataset is shuffled and split into 60% and 20% in training and validation, respectively. The remaining 20% is used for testing. Training is performed with an Intel Xeon CPU E5-1620 v4@3.50 GHz CPU with 16 GB of memory, TensorFlow 2.0, and Python 3.8.

Fig. 4 presents the confusion matrices of the selected models for binary classification (Fig. 4A) and multiclassification (Fig. 4B). For example, the CapsNet confusion matrix in binary classification, as shown in Fig. 4aA, indicates that 359 out of 1600 spoofed samples are correctly classified, while only 18 out of 1601 samples are misclassified as authentic signals. The Efficient-CapsNet confusion matrix, as illustrated in Fig. 4aB, shows that 713 out of 1600 spoofed

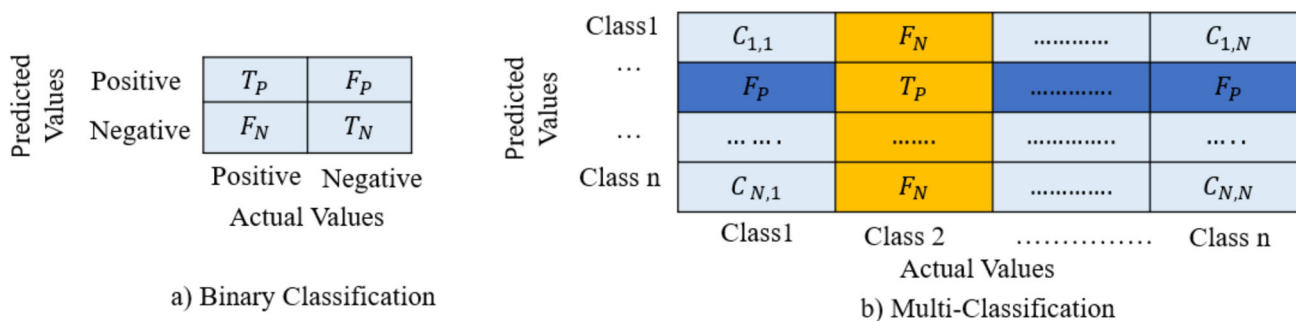


Fig. 3 Structure of confusion matrix for the binary and multi-classifications [28]

samples are classified correctly and only 1 sample out of 1601 is misclassified as authentic signal for binary classification. Moreover, the Efficient-CapsNet confusion matrix in multiclassification, as presented in Fig. 4bB, depicts 669 simplistic samples out of 1600, 268 intermediate samples out of 1600, 690 sophisticated samples out of 1600 samples, and 262 authentic out of 1601 samples, which are correctly classified. Also, only 1 intermediate sample out of 1600 samples is misclassified as sophisticated sample. The contents of these matrices were used to calculate the previously mentioned evaluation metrics, and the results are provided in Figs. 5, 6, 7, 8 and 9 and Tables 4, 5, and 6.

Figure 5 illustrates the accuracy of the proposed models. The accuracy of the Efficient-CapsNet model was higher compared to the other models. This model outperformed the other models with an accuracy of 98.29% for binary classification. The CNN model had the worst accuracy among all models, with an accuracy of 87.8% for the binary classes. The other models, DA-CapsNet and CapsNet, yielded acceptable results in terms of accuracy. Sophisticated attacks were detected in multiclassification, with the highest accuracy of 99.57% among all attacks using this Efficient-capsNet model; however, the simplistic and intermediate attacks were detected with lower accuracy than the sophisticated attacks using Efficient-CapsNet. These attacks could be detected and classified using CNN with the worst accuracy.

Figure 6 represents the probability of detection for the four models. The probability of detection of Efficient-CapsNet was higher than the other models for binary classification. This model had a probability of detection of 99.9% for binary classification; however, the CNN model had the lowest of detection among all models, with a probability of detection of 85.58% for binary classification. The sophisticated attacks were detected during multiclassification with a probability of detection of 99.85% using the Efficient-CapsNet model. Simplistic and intermediate attacks were detected with a lower probability of detection than that of the sophisticated attacks using Efficient-CapsNet. The other DL models, DA-CapsNet and CapsNet, yielded satisfactory results in terms of prob-

ability of detection. The attacks could be detected with the lowest probability of detection using the CNN model.

Figure 7 presents the results of the highlighted models in terms of probability of misdetection. The Efficient-CapsNet model yielded the lowest probability of misdetection of 0.1% among other models for binary classification, followed by DA-CapsNet, CapsNet, and CNN. All GPS spoofing attacks were detected during multiclassification with a low probability of misdetection using the Efficient-CapsNet model, while the same attacks were detected with the worst probability of misdetection using CNN. This figure also indicates that Efficient-CapsNet detects sophisticated attacks with a lower probability of misdetection than the other attacks. The intermediate attacks could be detected and classified with a slightly higher probability of misdetection using Efficient-CapsNet compared to the same attacks using DA-CapsNet and CapsNet. DA-CapsNet and CapsNet yielded a satisfactory probability of misdetection. The CNN model yielded the highest probability of misdetection for detecting and classifying GPS spoofing attacks.

Figure 8 presents the results of the selected models in terms of probability of false alarm. The Efficient-CapsNet model had the lowest result and the best probability of false alarm among all models for binary classification. The CNN model yielded the highest result and the worst probability of false alarm for binary classification. Efficient-CapsNet could detect any types of GPS spoofing for multiclassification, with a low probability of false alarm, ranging between 0.37% to 4.23%, while the other models had a considerably higher probability of false alarm. The intermediate attacks could be detected and classified using Efficient-CapsNet with a probability of false alarm of 0.05%, the lowest probability of false alarm compared to the other GPS spoofing attacks.

Table 4 lists the results of the other metrics, prediction time, training time per sample, and memory size. The Efficient-CapsNet model yielded the lowest prediction time, training time per sample, and memory size, while CNN performed the worst. The GPS spoofing attacks could be detected using Efficient-CapsNet, yielding the best results compared to the other models. The Efficient-CapsNet mod-

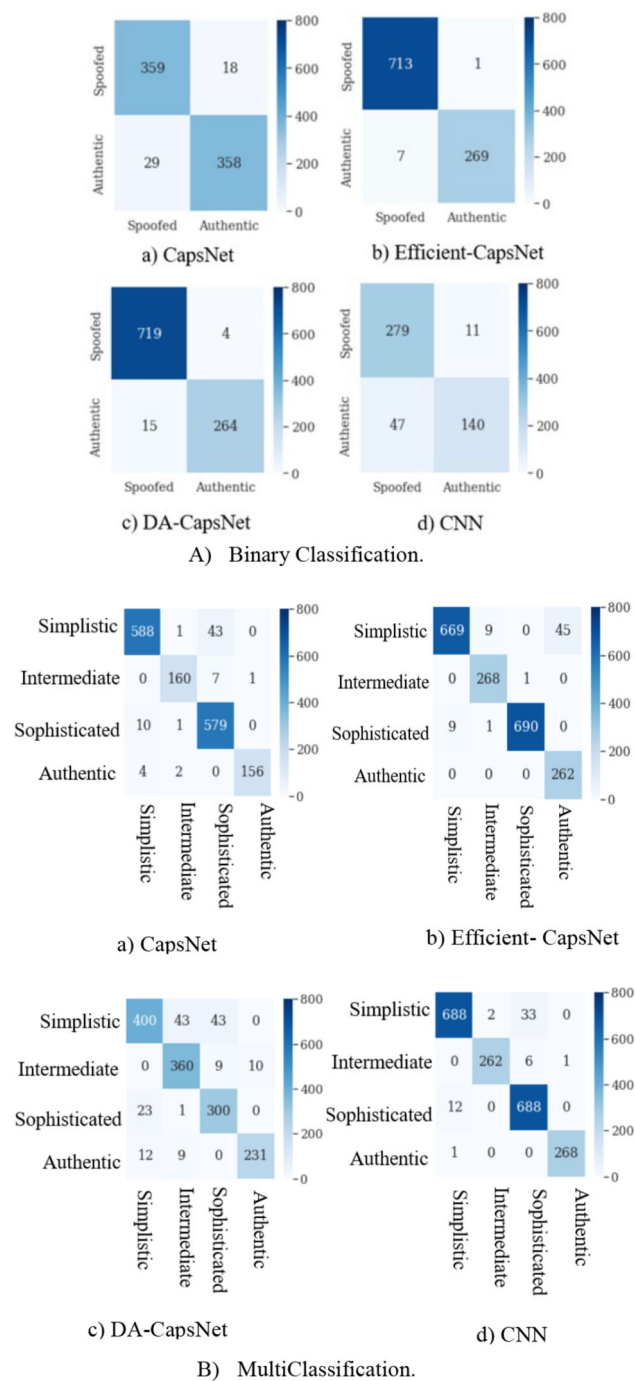


Fig. 4 Confusion matrices of the binary and four-class classification of the selected models

els outperformed the other models in terms of highlighted metrics due to its several benefits, such as a limited number of parameters, self-attention routing algorithm, and highly efficient pooling layers, resulting in better detection techniques and UAV-focused GPS spoofing attack classification.

It is critical to select efficient models for detecting and classifying UAV-focused GPS spoofing attacks due to the

size, weight, and power constraints; therefore, we selected time metrics, prediction time and training time per sample, as functions of Epoch. These metrics can present model effectiveness with respect to the amount of time a learning algorithm requires to complete the entire training benchmark. Understanding the training time per sample and prediction time in any deep learning model is pivotal for optimizing model performance and resource utilization. Knowing the time it takes to train the model on each sample enables the identification and rectification of bottlenecks in the training process, resulting in a faster convergence. This information aids in proper resource allocation, ensuring that computational power is utilized efficiently, particularly in large-scale or distributed training scenarios, such as UAV-based networks. Prediction time is equally crucial, especially in security applications. Faster prediction times lead to more responsive applications, contributing to a user-friendly interface. Additionally, awareness of time requirements helps in selecting appropriate hardware and managing associated costs. It also plays a role in model selection, allowing practitioners to balance model accuracy with computational efficiency. In essence, these time metrics serve as essential parameters for making informed decisions and enhancing the overall efficiency of DL-based workflows.

Tables 5 and 6 present the results of the existing studies for detecting and classifying UAV-focused GPS spoofing attacks and compare the results to our proposed models. Table 4 lists the results of the existing studies with respect to different datasets. The proposed models in this study were evaluated based on several evaluation metrics for binary and multi-classifications, although existing techniques were evaluated based on a limited number of metrics and binary classifications. For example, the authors [2] used artificial neural networks to detect and classify GPS spoofing attacks, using a dataset with 5 features and limited numbers of samples. The proposed model provides good performance with an accuracy of 98.3%, a probability of detection of 99.2%, a probability of misdetection of 0.8%, and false alarm of 2.6%. The authors [16] used an SVM-based technique to detect GPS spoofing attacks. The proposed approach provided a high accuracy of 98.77% for binary classification, using a dataset with 11 features. Although the results of these existing studies are satisfactory, the proposed models in this study provided higher performance, particularly the Efficient-CapsNet model which outperformed the other techniques from the literature.

Table 6 illustrates the results of the existing studies using a similar dataset and compares the results of this study to studies from the literature. A limited number of studies have detected and classified UAV-focused GPS spoofing attacks using the same dataset we used in this study. No existing works provide results for detecting and classifying different types of UAV-focused GPS spoofing attacks to the best

Fig. 5 Test accuracy of the proposed models

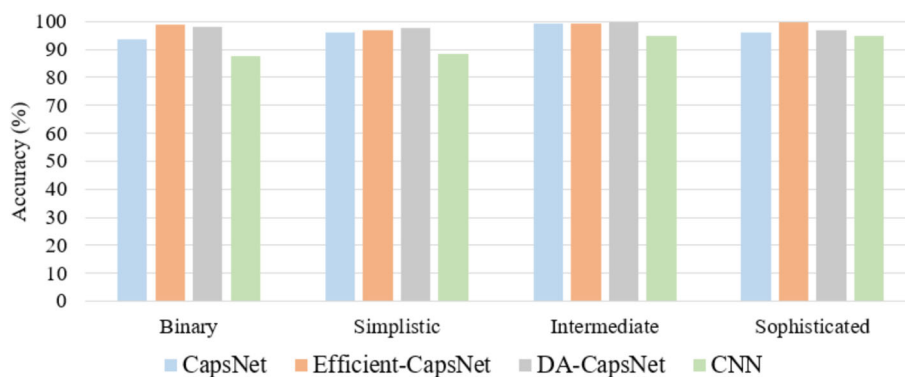


Fig. 6 Test probability of detection for the proposed models

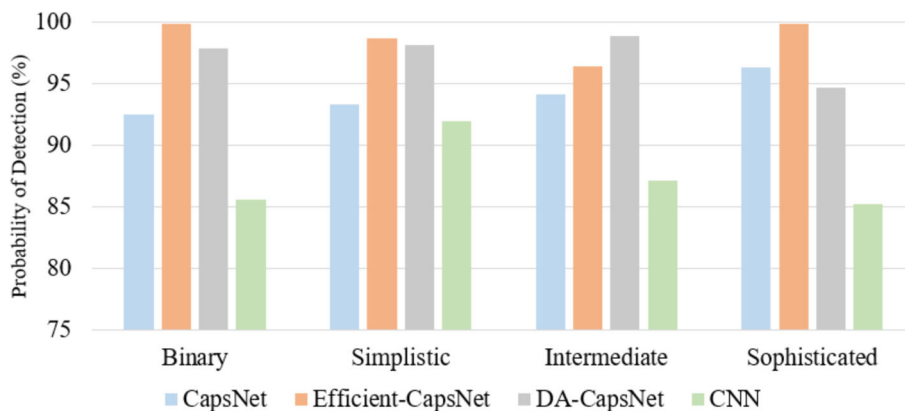


Fig. 7 Test Probability of misdetection for the proposed models

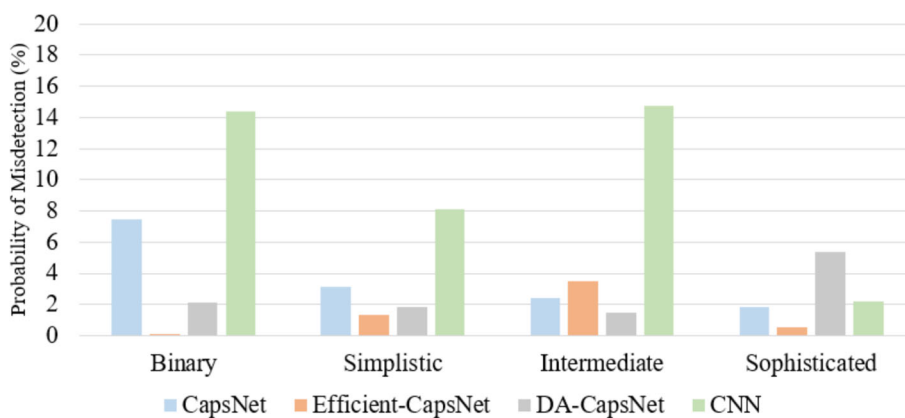


Fig. 8 Test probability of false alarm for the proposed models

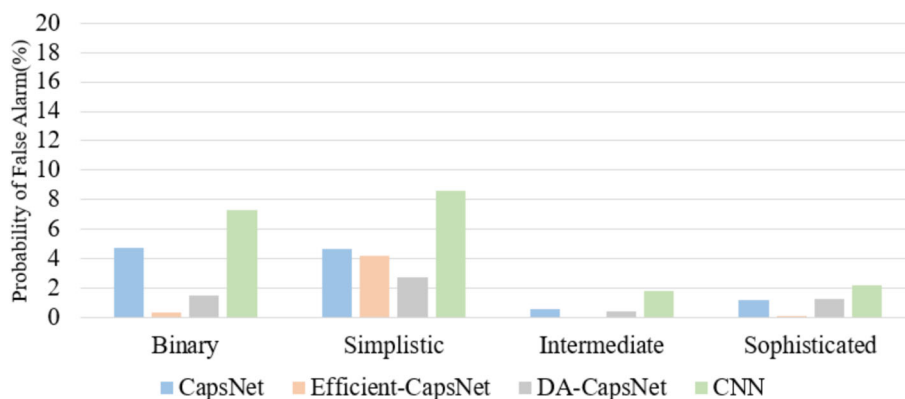


Table 4 Test results of the proposed models in terms of prediction time, training time per sample, and memory size

Model	Attack category	Prediction time (Seconds)	Training time per sample (Seconds)	Memory size (Mebibytes)
CapsNet	Simplistic	0.88	0.49	187
	Intermediate	0.82	0.34	172
	Sophisticated	0.79	0.3	166
	Binary	0.76	0.4	170
Efficient-CapsNet	Simplistic	0.54	0.32	127
	Intermediate	0.53	0.31	133
	Sophisticated	0.47	0.19	121
	Binary	0.5	0.2	123
DA-CapsNet	Simplistic	0.77	0.45	187
	Intermediate	0.67	0.33	149
	Sophisticated	0.69	0.21	129
	Binary	0.54	0.28	150
CNN	Simplistic	1.78	1.23	445
	Intermediate	1.76	0.99	490
	Sophisticated	1.12	0.78	421
	Binary	1.85	0.86	273.9

Bold indicates the best performance results

Table 5 Comparison of our study to other related works using different datasets

Models used	Predicted classes	ACC (%)	P_d (%)	P_{md} (%)	P_{fa} (%)	PT (milli-Seconds)	TPS (Seconds)	M (Mebibytes)
Artificial Neural Network [2]	Spoofed and Non-Spoofed	98.3	99.2	0.8	2.6	N/A	N/A	N/A
Genetic Algorithm and Extreme Boosting [14]	Hijacked and Non-Hijacked	N/A	96.3, and 100	N/A	N/A	N/A	N/A	N/A
Support Vector Machine [16]	Spoofed and Non-Spoofed	98.77	N/A	N/A	N/A	N/A	N/A	N/A
Long Short-Term Memory [20]	Spoofed and Non-Spoofed	N/A	78	N/A	N/A	N/A	N/A	N/A
DeepSIM [21]	On Ground	89.5	N/A	N/A	N/A	N/A	N/A	4291.53
	On Board	82.6	N/A	N/A	N/A	N/A	N/A	271.797
Our Models	Binary	93.84	92.52	7.5	4.78	0.76	0.4	170
	Simplistic	95.95	93.3	3.14	4.63	0.88	0.49	187
	Intermediate	99.29	94.1	2.44	0.57	0.82	0.34	172
	Sophisticated	96.10	96.3	1.87	1.17	0.79	0.30	166
	Binary	99.1	99.9	0.1	0.37	0.5	0.2	123
	Simplistic	96.77	98.67	1.32	4.23	0.54	0.32	127
	Intermediate	99.43	96.40	3.5	0.059	0.53	0.31	133
	Sophisticated	99.57	99.85	0.57	0.14	0.47	0.19	121
DA-CapsNet	Binary	98.10	97.95	2.1	1.49	0.54	0.28	150
	Simplistic	97.55	98.14	1.85	2.77	0.77	0.45	187
	Intermediate	99.54	98.88	1.46	0.41	0.67	0.33	149
	Sophisticated	96.98	94.63	5.36	1.24	0.69	0.21	129
CNN	Binary	87.84	85.58	14.42	7.28	1.85	0.86	573.9
	Simplistic	88.37	91.95	8.15	8.54	1.78	1.23	445
	Intermediate	95	87.16	14.77	1.84	1.76	0.99	490
	Sophisticated	94.72	85.22	2.2	2.2	1.12	0.78	421

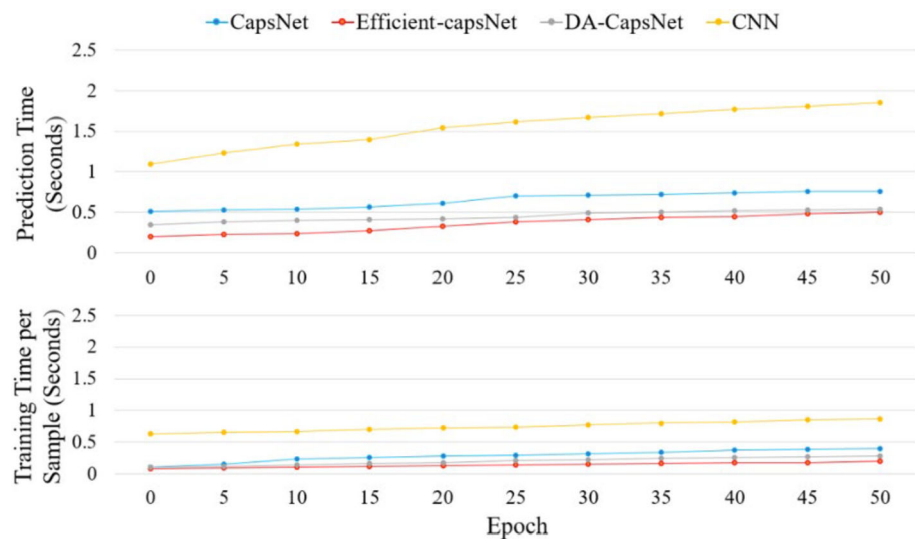
Bold indicates the best performance results

Table 6 Comparison of our study to other related works using same dataset

Predicted Classes	Predicted Classes	ACC (%)	P_d (%)	P_{md} (%)	P_{fa} (%)	PT (milli-Seconds)	TPS (Seconds)	M (Mebibytes)
Spoofed and non-spoofed [15]	Random Forest	94.07	96.23	3.77	8.53	21.01	N/A	N/A
	Gradient Boosting	91.45	92.52	7.48	9.84	6.99	N/A	N/A
	Light Gradient Boosting	95.23	95.38	4.62	4.96	8.99	N/A	N/A
	Extreme Gradient Boosting	95.52	95.38	4.62	4.3	2	N/A	N/A
Spoofed and non-spoofed[18]	Radius Neighbor	73.81	71.97	58.3	41.7	2.7	N/A	N/A
	K Nearest Neighbor	79.08	70.75	29.2	13.7	0.22	N/A	N/A
	Linear Support Vector Machine	64.56	62.08	37.9	17.6	0.04	N/A	N/A
	Nu- Support Vector Machine	92.78	91.26	8.73	6.02	0.12	N/A	N/A
Spoofed and non-spoofed [19]	C- Support Vector Machine	91.85	88.86	11.1	7.7	0.16	N/A	N/A
	Bagging	95.28	99.24	0.64	1.07	N/A	N/A	190.4
	Stacking	95.43	99.56	0.36	0.03	N/A	N/A	191.3
	Boosting	94.61	96.55	2.95	5.08	N/A	N/A	19 0.5
Our Models	Binary CapsNet	93.84	92.52	7.5	4.78	0.76	0.4	170
	Simplistic	95.95	93.3	3.14	4.63	0.88	0.49	187
	Intermediate	99.29	94.1	2.44	0.57	0.82	0.34	172
	Sophisticated	96.10	96.3	1.87	1.17	0.79	0.30	166
	Efficient-CapsNet	99.1	99.9	0.1	0.37	0.5	0.2	123
	Simplistic	96.77	98.67	1.32	4.23	0.54	0.32	127
	Intermediate	99.43	96.40	3.5	0.059	0.53	0.31	133
	Sophisticated	99.57	99.85	0.57	0.14	0.47	0.19	121
	DA-CapsNet	98.10	97.95	2.1	1.49	0.54	0.28	150
	Simplistic	97.55	98.14	1.85	2.77	0.77	0.45	187
	Intermediate	99.54	98.88	1.46	0.41	0.67	0.33	149
	Sophisticated	96.98	94.63	5.36	1.24	0.69	0.21	129
CNN	Binary	87.84	85.58	14.42	7.28	1.85	0.86	573.9
	Simplistic	88.37	91.95	8.15	8.54	1.78	1.23	445
	Intermediate	95	87.16	14.77	1.84	1.76	0.99	490
	Sophisticated	94.72	85.22	2.2	2.2	1.12	0.78	421

Bold indicates the best performance results

Fig. 9 Test results for prediction time and training time per sample with respect to epoch



of our knowledge. Existing studies in the literature used a limited number of evaluation metrics, resulting in inaccurate evaluations; therefore, the DL models proposed in this study exhibited higher performance when detecting and classifying attacks on UAVs compared to other studies in the literature. Efficient-CapsNet yielded the best results for detecting and classifying binary and multiclass. The CNN model had the lowest performance for detecting UAV-focused GPS spoofing attacks. The other models, CapsNet and DA-CapsNet, had moderately lower accuracy, probability of detection, higher probability of misdetection, probability of false alarm, prediction time, training time per sample, and memory size compared to Efficient-CapsNet. Sophisticated attacks could be classified and detected with higher-performance results than other GPS spoofing attacks. Efficient-CapsNet could detect these attacks with high accuracy, probability of detection, low probability of misdetection, probability of false alarm, prediction time, training time per sample, and memory size. Efficient-CapsNet could also yield lower prediction times and training time per sample than other proposed DL models with respect to Epoch. The results can be summarized as follows:

- The capsule family outperforms other existing techniques for binary and multi-class problems,
- Efficient-CapsNet yielded the best results for binary and multi-class problems in terms of selected metrics,
- CNN had the worst results among Efficient-CapsNet, CapsNet, and Da-CapsNet for binary and multi-class problems.

5 Conclusion

Many approaches have been proposed in the literature to detect and classify GPS spoofing attacks; however, these techniques have several limitations, including high false alarm and misdetection rates. We evaluated and compared the performance of new deep learning model types, Capsule Networks (Simple Capsule Network, Efficient Capsule Network, and Dual Attention Capsule Network), to Convolutional Neural Networks. The evaluation was performed in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, prediction time, training time per sample, memory size, and learning curves. The results indicate that the Efficient-Capsule Network yielded better results than the Capsule Network, Dual Attention Capsule Network, and Convolutional Neural Network with an accuracy of 99.1%, a probability of detection of 99.9%, a probability of misdetection of 0.1%, a probability of false alarm of 0.37%, a prediction time of 0.5 seconds, and a training time per sample of 0.2 seconds. The Convolutional Neural Network model yielded the worst results with an accuracy of 87.84%, a probability of detection of 85.57%, a probability of misdetection of 14.42%, a probability of false alarm of 7.28%, a prediction time of 1.85 seconds, a training time per sample of 0.86 seconds, and a memory size of 573.9 mebibytes.

Acknowledgements The authors acknowledge the support of the National Science Foundation (NSF), Award Number: 2006674.

Author Contributions Conceptualization, T.T.K.; software, T.T.K.; formal analysis, T.T.K.; investigation, T.T.K.; methodology, T.T.K.; visualization, T.T.K.; writing—original draft, T.T.K., K.A.S., and N.K.; writing—review and editing, T.T.K., K.A.S., and N.K.; project administration, K.A.S. and V.K.D. N.K.; validation, K.A.S. and V.K.D.; supervision, N.K.; funding acquisition, N.K. All authors have read and agreed to the published version of the manuscript.

Funding Open access funding provided by Northeastern University Library This work has the support of the National Science Foundation (NSF), Award Number: 2006674.

Data Availability Dataset is available at <https://ieee-dataport.org/documents/dataset-gps-spoofing-detection-autonomous-vehicles>.

Declarations

Conflicts of interest The authors declare no conflicts of interest relevant to this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Kerns, A.J., Shepard, D.P., Bhatti, J.A., Humphreys, T.E.: Unmanned aircraft capture and control via GPS spoofing. *J. Field Robotics* **31**(4), 617 (2014)
- Manesh, M.R., Kaabouch, N.: Cyber-attacks on unmanned aerial system networks: Detection, countermeasure, and future research directions. *Comput. Security* **85**, 386 (2019)
- Talaei Khoei, T., Ismail, S., Shamaileh, K.A., Devabhaktuni, V.K., Kaabouch, N.: Impact of dataset and model parameters on machine learning performance for the detection of gps spoofing attacks on unmanned aerial vehicles. *Appl. Sci.* **13**(1), 383 (2022)
- Talaei Khoei, T., Ismail, S., Kaabouch, N.: Dynamic selection techniques for detecting GPS spoofing attacks on UAVs. *Sensors* **22**(2), 662 (2022)
- Manickam, S.: Gps signal authentication using ins-a comparative study and analysis. Ph.D. thesis, University of Calgary (2016)
- Jiang, P., Wu, H., Xin, C.: DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network. *Digital Commun. Networks* **8**(5), 791 (2022)
- Narain, S., Ranganathan, A., Noubir, G.: Security of GPS/INS based on-road location tracking systems. In: 2019 IEEE Symposium on Security and Privacy (SP) (IEEE, 2019), pp. 587–601
- Manesh, M.R., Kenney, J., Hu, W.C., Devabhaktuni, V.K., Kaabouch, N.: Detection of GPS spoofing attacks on unmanned aerial systems. In: 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC) (IEEE, 2019), pp. 1–6
- Qiu, W., Tang, Q., Wang, Y., Zhan, L., Liu, Y., Yao, W.: Multi-view convolutional neural network for data spoofing cyber-attack detection in distribution synchrophasors. *IEEE Trans. Smart Grid* **11**(4), 3457 (2020)
- Patrick, M.K., Adekoya, A.F., Mighty, A.A., Edward, B.Y.: Capsule networks-a survey. *J. King Saud Univ.-Comput. Information Sci.* **34**(1), 1295 (2022)
- Kwon, K.C., Shim, D.S.: Performance analysis of direct gps spoofing detection method with ahrs/accelerometer. *Sensors* **20**(4), 954 (2020)
- Qiao, Y., Zhang, Y., Du, X.: A vision-based GPS-spoofing detection method for small UAVs. In: 2017 13th International Conference on Computational Intelligence and Security (CIS) (IEEE, 2017), pp. 312–316
- Varshosaz, M., Afary, A., Mojaradi, B., Saadatseresht, M., Ghanbari Parmehr, E.: Spoofing detection of civilian UAVs using visual odometry. *ISPRS Int. J. Geo Inf.* **9**(1), 6 (2019)
- Feng, Z., Guan, N., Lv, M., Liu, W., Deng, Q., Liu, X., Yi, W.: Efficient drone hijacking detection using two-step GA-XGBoost. *J. Syst. Architect.* **103**, 101694 (2020)
- Aissou, G., Slimane, H.O., Benouadah, S., Kaabouch, N.: Tree-based supervised machine learning models for detecting GPS spoofing attacks on UAS. In: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON) (IEEE, 2021), pp. 0649–0653
- Semanjski, S., Semanjski, I., De Wilde, W., Muls, A.: Use of supervised machine learning for GNSS signal spoofing detection with validation on real-world meaconing and spoofing data-Part I. *Sensors* **20**(4), 1171 (2020)
- Panice, G., Luongo, S., Gigante, G., Pascarella, D., Di Benedetto, C., Vozella, A., Pescapè, A.: A SVM-based detection approach for GPS spoofing attacks to UAV. In: 2017 23rd International Conference on Automation and Computing (ICAC) (IEEE, 2017), pp. 1–11
- Aissou, G., Benouadah, S., El Alami, H., Kaabouch, N.: Instance-based supervised machine learning models for detecting GPS spoofing attacks on UAS. In: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (IEEE, 2022), pp. 0208–0214
- Gasimova, A., Khoei, T.T., Kaabouch, N.: A comparative analysis of the ensemble models for detecting gps spoofing attacks on uavs. In: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) (IEEE, 2022), pp. 0310–0315
- Wang, S., Wang, J., Su, C., Ma, X.: Intelligent detection algorithm against uavs' gps spoofing attack. In: 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS) (IEEE, 2020), pp. 382–389
- Xue, N., Niu, L., Hong, X., Li, Z., Hoffaeller, L., Pöpper, C.: Deepsim: Gps spoofing detection on uavs using satellite imagery matching. In: Annual computer security applications conference (2020), pp. 304–319
- Albawi, S., Mohammed, T.A., Al-Zawi, S.: Understanding of a convolutional neural network. In: 2017 international conference on engineering and technology (ICET) (Ieee, 2017), pp. 1–6
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016), pp. 770–778
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **115**, 211 (2015)
- He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision (2015), pp. 1026–1034
- Mazzia, V., Salvetti, F., Chiaberge, M.: Efficient-capsnet: Capsule network with self-attention routing. *Sci. Rep.* **11**(1), 14634 (2021)
- Huang, W., Zhou, F.: DA-CapsNet: dual attention mechanism capsule network. *Sci. Rep.* **10**(1), 11383 (2020)
- Markoulidakis, I., Kopsiaftis, G., Rallis, I., Georgoulas, I.: Multi-class confusion matrix reduction method and its application on net promoter score classification problem. In: The 14th pervasive

technologies related to assistive environments conference (2021), pp. 412–419

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Tala Talaei Khoei received the B.S. degree in Computer Software Engineering from Azad University, Mashhad, Iran in 2014 and the M.S. degree in Information Technology from Southern New Hampshire University, NH, USA in 2016. She received the Ph.D. degree in Electrical Engineering from University of North Dakota, ND, USA in 2023. She joined the Khoury College of Computer Science, Northeastern University at January 2024. Her research interests include cyber-security of

smart grid, security of unmanned aerial vehicle, artificial intelligence, information security, data science, and big data.



Khair Al Shamaileh (Member, IEEE) received the B.Sc. degree in communications and electronics engineering and the M.Sc. degree in wireless communications engineering from the Jordan University of Science and Technology, in 2009 and 2011, respectively, and the Ph.D. degree in engineering from The University of Toledo, USA, in 2015. He joined the ECE Department, Purdue University Northwest, as an Assistant Professor, in 2016. His research interests include physical

layer security, microwave modeling, RF circuit design, sensor networks, localization algorithms, and applied optimization to engineering problems. His research has been sponsored by the National Science Foundation.



Vijaya Kumar Devabhaktuni received the B.Eng. degree in electrical and electronics engineering, the M.Sc. degree in physics from the Birla Institute of Technology and Science, Pilani, India, in 1996, and the Ph.D. degree in electronics from Carleton University, Ottawa, Canada, in 2003. His research interests include applied electromagnetics, biomedical applications of wireless sensor networks, computer-aided design, device modeling, image processing, infras-

structure monitoring, neural networks, RF/microwave design, unmanned aerial vehicles, and virtual reality.



Naima Kaabouch is Full Professor and Director of the Artificial Intelligence Initiative at the University of North Dakota, USA. She received her B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of Paris 11 and the University of Paris 6, France, respectively. Her main research interests include artificial intelligence, wireless communication and networking, cyber-security, and autonomous systems. She is the author of several research handbooks and numerous

peer reviewed articles and book chapters.