

Image-based intrusion detection system for GPS spoofing cyberattacks in unmanned aerial vehicles[☆]

Mohamed Selim Korium^{a,*}, Mohamed Saber^c, Ahmed Mahmoud Ahmed^b, Arun Narayanan^a, Pedro H.J. Nardelli^a

^a Lappeenranta–Lahti University of Technology, Finland

^b Zagazig University, Egypt

^c Cairo University, Egypt

ARTICLE INFO

Dataset link: <https://iee-dataport.org/open-access/uav-attack-dataset>

Keywords:

Unmanned aerial vehicles
CNN architectures
Transfer learning
Learning and loss curves

ABSTRACT

The operations of unmanned aerial vehicles (UAVs) are susceptible to cybersecurity risks, mainly because of their firm reliance on the Global Positioning System (GPS) and radio frequency (RF) sensors. GPS and RF sensors are vulnerable to potential threats, such as spoofing attacks that can cause the UAVs to behave erratically. Since these threats are widespread and potent, it is imperative to develop effective intrusion detection systems. In this paper, we propose an image-based intrusion detection system for detecting GPS spoofing cyberattacks based on a deep learning methodology. We combine convolutional neural networks with Principal Component Analysis (PCA) to reduce the dimensionality of the dataset features, data augmentation to increase the size and diversity of the training dataset, and transfer learning to improve the proposed model's performance with limited data to design a fast, accurate, and general method. Extensive numerical experiments demonstrate the effectiveness of the proposed solution carried out using benchmark datasets. We achieved an accuracy of 100% within a running time of 120.64 s at 0.3529 ms latency and a detection time of 2.035 s in the case of the training dataset. Further, using this trained model, we achieved an accuracy of 99.25% within a detection time of 2.721 s on an unseen dataset that was unrelated to the one used for training the model. In contrast, other models, such as Inception-v3, showed lower accuracy on unseen datasets. However, Inception-v3 performance improved significantly after Bayesian optimization, with the Tree-structured Parzen Estimator reaching 99.06% accuracy. Our results demonstrate that the proposed image-based intrusion detection method outperforms the existing solutions while providing a general model for detecting cyberattacks included in unseen datasets.

1. Introduction

1.1. Background

Unmanned aerial vehicles (UAVs), commonly referred to as drones, are aircraft that fly automatically without any human pilot, crew, or passengers on board. UAVs were initially employed for military tasks, such as surveillance, bomb detection, delivery of armed payloads, and various missiles such as, AGM-114 Hellfire, used by the United States [1]. In recent times, UAVs have also attracted significant interest from civilian industries for their potential applications for various civil requirements such as entertainment, inspection, surveillance, and mapping, assistance during natural disasters, and intelligent urban traffic

control [2]. Further, according to some studies [3,4], the market value of UAVs is expected to reach \$1.85 billion by the year 2024.

Most civilian tasks rely heavily on the radio frequency (RF) band that enables data transmission, such as real-time video streaming, sensor data [5], and signals received from the Global Navigation Satellite System (GNSS) that provides positioning, navigation, and timing services [6]. Examples of a GNSS include the Global Positioning System (GPS), owned by the United States; the Global Navigation Satellite System (GLONASS), owned by Russia; the Galileo Global Navigation Satellite System, owned by the EU; and the Quasi-Zenith Satellite System, owned by Japan [7].

However, modern improvements in the connectivity of UAV networks for executing tasks have significantly increased cybersecurity

[☆] This paper was partly funded by Jane and Aatos Erkko Foundation via the project STREAM, by EU MSCA project COALESCE (n.101130739), and by Research Council of Finland via “X-SDEN” (n.349965), EnergyNet (n.321265/n.328869/n.352654) and ECO-NEWS (n.358928).

* Corresponding author.

E-mail address: Mohamed.Korium@lut.fi (M.S. Korium).

<https://doi.org/10.1016/j.adhoc.2024.103597>

Received 26 February 2024; Received in revised form 30 June 2024; Accepted 9 July 2024

Available online 16 July 2024

1570-8705/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

risks. This is due to the fact that UAVs primarily rely on GPS and RF sensors that are easily exposed to potential cybersecurity threats, such as spoofing attacks [6]. In a spoofing attack, counterfeit GPS signals control the command functions and steal sensitive data by replacing the GPS signals with a false signal that mimics the GPS format, causing the UAV to be set to a wrong trajectory [8]. The vulnerability of GPS systems to spoofing attacks can lead to severe consequences. For instance, researchers at the University of Texas took control of an \$80 million yacht in the Mediterranean Sea by changing the trajectory of the yacht for about 100 m [9]. Researchers at Virginia Tech, China, and Microsoft changed the destination of an autonomous vehicle by manipulating the navigation system using their own-built \$225 GPS spoofer in 2018 [10]. These studies have investigated the vulnerability of GPS systems to spoofing attacks by using software-defined radios (SDRs) that encompass many devices such as High Amplitude Complex Keying Radio Frequency One (HackRF One), Great Scott Gadgets HackRF software-defined radio (HackRF), Universal Software Radio Peripheral (USRP), and Analog devices Active Learning Module - Pluto (ADALM-Pluto). SDR devices are a type of radio communication system that receives and generates signals using software by converting signals into RF signals (GPS signals) on a general-purpose computer instead of traditional hardware components like filters and amplifiers, making them more adaptable and reconfigurable for attackers [11]. To conduct a GPS spoofing attack by using an SDR device, the cyber-malicious agent (i) first identifies the authentic GPS signal that the UAV receives; (ii) uses a GPS signal simulator like the Global Position System-Software-Defined Radio simulator (GPS-SDR-SIM) [12] or the Global Navigation Satellite System-Software-Defined Radio Library (GNSS-SDRLIB) software [13] to parse the CSV file created by the cyber-malicious agent that holds the spoofed location and number of movements in the Earth-centered, Earth-fixed coordinate system (ECEF) format; (iii) employs the SDR device to convert the GPS baseband signal data streams created by the GPS signal simulator into GPS signals (the desired signal by the hacker); and (iv) transmits the spoofed GPS by the SDR to the UAV after adjusting the frequency and signal power. As a result, the UAV will lose the authentic GPS signal that indicates the proper location and time and will lock onto the spoofed GPS signal in 15 to 300 s, depending on the embedded algorithms and the chosen spoofed GPS strategy. Therefore, GPS spoofing has severe effects on applications that rely on synchronization (accurate positioning and timing).

1.2. Related research, its limitations, and our contribution

In the context of UAV cyberattacks, previous studies have focused on conventional methods of signal processing and cryptography to analyze and defend against a spoofing GPS attack. The cryptography technique involves encrypting the satellite codes received and sent by the GPS receiver. This technique requires a public key to decrypt the encrypted code; a digital signature to verify the authenticity and integrity of the code; a key management system to generate the public and private keys; and a secure communication protocol to grant the authenticity and integrity transmission of the encrypted and signed code by a digital signature between the satellites and the GPS receiver. However, this technique has a substantial cost: it adds complexity and may not be compatible with older versions of receivers and signals. The signal distortion detection technique tries to find any anomalies by monitoring the quality and characteristics of the GPS signal. The technique is implemented by measuring the strength of the signal and the frequency compared with GPS signals under normal conditions. However, it cannot detect complicated spoofing attacks that could seamlessly shift to the GPS receiver without any abrupt changes or errors. The direction-of-arrival sensing technique uses multiple antennas to estimate the original GPS signal direction. It compares it with the authentic GPS signal direction or the known direction of the satellites. The drawback of this technique is that it is affected by environmental factors that may block the GPS signals [14–16].

Kwon et al. [17] used the probability density function (PDF) to evaluate the accelerometer reading from an inertial measurement unit (IMU) to detect a GPS spoofing attack. Their proposed method is carried out by monitoring the difference in the accelerometer readings with standard authentic GPS signals and spoofed GPS signals. In [18,19], the authors used the outputs of accelerometers that measure linear acceleration and the results of onboard gyroscopes that measure angular velocity or rotation rate, respectively. Both studies compared the outputs of IMU measurements that are computed from both the UAV and the GPS to determine whether the outputs match. If the outputs of the IMU measurement do not match the GPS outputs, it indicates that the GPS is spoofed. However, IMU measurements are prone to various errors, such as bias and noise, that may affect the measurements of angular rate and acceleration, which are integrated into the inertial navigation system (INS). This error accumulation results in an inaccurate state of the UAV, making it hard to distinguish between standard authentic GPS signals and spoofed GPS signals [20].

Other studies [21,22] have suggested that using artificial intelligence (AI) algorithms may have some advantages over all the above techniques because the algorithms may gain experience from diverse spoofed GPS scenarios. In [23], the authors proposed a Long Short-Term Memory (LSTM) algorithm to detect GPS spoofing attacks by predicting Unmanned Aircraft System (UAS) paths and comparing them with the authentic GPS positioning signal. This comparison allows the algorithm to confirm whether the UAS is following an authentic or a spoofed trajectory. Although the detection accuracy of the proposed model reached 78%, the method is not applicable if the UAV shifts smoothly to the spoofed GPS with a small error deviation compared with the accepted threshold that the cyber-malicious agent has predefined.

Semanjski et al. [24] used a known supervised learning algorithm, C-Support Vector Machine (C-SVM), to detect GPS spoofing attacks depending on the features of the receiver clock that synchronizes with the satellites to determine the location of the UAV in real-time. This allowed the authors to monitor the behavior of the receiver clock for both authentic and spoofed GPS signals. They claimed that the cross-validation accuracy of the C-SVM model and the overall accuracy were 97.8% on real-world data. However, they should have mentioned how they handled imbalanced datasets. Moreover, in their study, the GPS spoofing attack is an unusual occurrence, and thus, the number of normal samples is much higher than the malicious samples. As a result, since the C-SVM is very sensitive to imbalanced datasets, the model has a high overall accuracy but will perform inaccurately on minor classes.

In [25], the authors employed the Multilayer Perceptron model (MLP) to detect GPS spoofing attacks by using path loss measurements collected from nearby cellular base stations. Their model achieved an accuracy of 80% with one base station and 93% with three base stations. However, this method relies on the cellular network, which itself could be jammed, affecting the path loss measurements and leading to inaccurate detection. Moreover, UAVs are used to perform in high-risk environments, and there may be issues with cellular network connectivity in urban or dead zone areas.

In [26], machine learning algorithms, namely random forest, gradient boost, XGboost, and LightGBM, were used to detect GPS spoofing attacks based on a real-time dataset generated by a Universal Software Radio Peripheral (USRP) device and simulated spoofed GPS signals. The results showed that XGBoost achieved the highest accuracy of 95.52%. At the same time, the random forest outperformed other models in terms of false alarms and misdetection.

Gasimova et al. [27] proposed an ensemble of machine-learning models for bagging, boosting, and stacking. The performance of ensemble learning models has reached 95.28%, 95.43%, and 94.61% at prediction times 0.24s, 0.02s, and 0.01s for bagging, stacking, and boosting, respectively.

Xiao et al. [28] suggested a deep learning model, namely a recurrent Neural Network (RNN), for detecting GPS spoofing attacks. The detection method is based on selecting the optimal threshold for Normalized

Root Mean Square Error (NRMSE). The proposed model achieved an accuracy of 98.7% at 0.034s.

Richmond et al. [29] proposed deep learning models, namely, Long short-term memory (LSTM) and Autoencoder, based on flight log datasets. The dataset is generated by using PX4 Autopilot firmware with QGroundControl Application and has about 88 features. The results demonstrated that the models achieved 97.79% by LSTM for binary classifiers and 94.98% by Autoencoder for one-class classifiers.

Tala et al. [30] used two advanced ensembles learning, namely, Metric Optimized Dynamic selector (MOD) and Weighted Metric Optimized Dynamic selector (WMOD), for detecting GPS spoofing attacks for UAVs based on a generated real-time dataset with 13 GPS signal features. The proposed models achieved an accuracy of 99.8% for MOD and WMOD, 99.6% for bagging ensembles learning, 99.56% for boosting ensembles learning, and 99.7% for stacking ensembles learning.

Eshmawi et al. [31] also proposed a stacked ensemble approach, a combination of Convolutional Neural Network (CNN) and Support Vector Machine (SVM) to detect GPS signal spoofing within the context of small UAVs based on a dataset for GPS spoofing detection on autonomous vehicles. Although the datasets contain about 13 features related to GPS signals only with no UAV features such as roll, heading, groundspeed, and other features that are related to the inertia measurement unit (IMU), other sensors that are affected by the position of the UAV, the author used the dataset for detecting GPS signals for small UAVs. The proposed model achieved an accuracy of 99.72% for the SVM-CNN model.

Whelan et al. in [32,33], used anomaly detection models such as one-class support vector machine, local outlier factor, and autoencoder to detect GPS spoofing and jamming based on their proprietary dataset named UAV Attack Datasets. The models were trained on flight data only, without any attacks occurring, to recognize normal behavior, allowing the proposed models to detect any deviation to be flagged as a cyberattack. The performance of their models achieved a macro-averaged F1-score of 90.57% and 94.3% for GPS spoofing and jamming, respectively.

In general, these prior studies for detecting GPS spoofing attacks, whether using traditional signature-based detection strategies, intrusion detection systems, or Artificial intelligence methods, suffer from the following limitations:

- Most of the time, UAVs operate under normal conditions without encountering GPS spoofing attacks. This leads to generating a dataset with a vast number of benign data compared to GPS spoofing attack data. Therefore, UAV datasets often have imbalanced datasets, but still, many of the studies did not mention how they handled the imbalanced datasets;
- Few studies have used datasets for GPS spoofing detection in autonomous vehicles, which could be used for UAVs as well, but there are a few considerations related to the difference between the design for ground-based autonomous vehicles and the unique dynamics and environmental factors of UAVs. Because UAVs and ground vehicles operate in fundamentally different environments. UAVs contend with atmospheric conditions, altitude variations, and aerial obstacles, which influence GPS signal reception and spoofing characteristics differently than ground vehicles;
- Few studies considered the detection and execution time, although they are essential in cybersecurity, where time-sensitive activities must be carried out while detecting and thwarting UAV cyberattacks;
- In real-world scenarios, datasets are often more diverse and contain various anomalies and noise that were not present in the training data. Therefore, even if the proposed models in the literature achieved high accuracy on a specific dataset, they may need to perform more effectively on other datasets with different UAV features.

Most studies avoid testing their models on unseen datasets due to the diversity of unmanned aerial vehicles (UAVs) in terms of their design, capabilities, sizes, and functionalities. Designing a single algorithmic learning model to detect various GPS spoofing scenarios for these heterogeneous UAVs is a complex and challenging task. Conversely, it is not practical to develop a unique algorithmic learning model for each GPS spoofing attack scenario because that becomes exponentially complex as the number of UAVs and potential attacks increases. This requires significant computational resources, time, processing power, and storage, which could be more efficient and costly. Also, individual models may overfit specific UAVs or attack scenarios, failing to generalize well to slightly different cases. This reduces the overall effectiveness of the system in real-world applications, can strain resources, and negatively impacts data transmission and the environment where variability is expected. Therefore, in this study, we have innovatively addressed the critical gaps in the current literature by designing an image-based intrusion detection system for heterogeneous UAVs. The proposed image-based intrusion detection system is based on converting the numerical features into images, which is particularly beneficial in complex scenarios with numerous heterogeneous UAVs and varying GPS spoofing attacks because this conversion integrates multiple numerical features into a single coherent image, making it easier for the model to capture complex interactions. Given the diverse nature of unmanned aerial vehicles (UAVs) and GPS spoofing attack scenarios, which leads to a massive number of models, one for each, this approach significantly reduces the need for an enormous number of models. Instead, it leverages a single, powerful model capable of generalizing across different UAVs and spoofing attacks. In addition, this conversion could be easily augmented, which helps us generate a wider range of conditions and scenarios, improving our proposed model's performance on the unseen dataset and also fine-tuning (transfer learning) these models can help them adapt quickly and efficiently to new UAVs and spoofing scenarios that were not included in the training dataset. This allows us to have a single algorithmic learning that can handle a wide variety of scenarios instead of training separate models for each UAV. Therefore, the main contributions of this paper to bridge this gap in the previous research are:

- We introduce an image-based intrusion detection system that employs convolutional neural network architectures with transfer learning for detecting GPS spoofing attacks across heterogeneous UAVs;
- We discuss the uses of different oversampling techniques, such as the Synthetic Minority Oversampling Technique with the Edited Nearest Neighbors and Adaptive Synthetic Sampling algorithm, and we determine the most suitable oversampling technique for the given datasets;
- We employ data transformation with pipeline optimization method to streamline the time taken for image loading and processing, harmonizing feature dimensions to address discrepancies in the number of features for training and unseen datasets using principal component analysis, and Bayesian optimization with tree Parzen estimator to reduce the overfitting;
- We show, for the first time, how a high-accuracy model that overfits can cause a problem in detecting cyberattacks by not generalizing properly, and propose a method to reduce overfitting and to generalize well to the unseen dataset;
- We discuss the significance of illustrating the combination of the transfer learning, learning curves, and loss curves in one plot to illustrate the effectiveness of the chosen existing deep learning methods and evaluate the performance and convergence of the proposed models.

To the best of our knowledge, no previous work has proposed such an image-based intrusion detection method on GPS spoofing scenarios and tested their proposed model on a different dataset (unseen dataset) that includes different UAV types, feature sets, and number of features

to detect GPS spoofing attacks; illustrated a plot that combines learning and loss curves for training and validation before and after applying transfer learning; evaluated the proposed models with many different classification metrics; meticulously detailed the testing process; and rigorously evaluated the proposed models using unseen datasets. Developing a model to identify diverse GPS spoofing attacks poses a formidable challenge due to the various types of UAVs with different types of sensors and communication protocols. This diversity in UAV types and features increases complexity. It intensifies the intrusion detection problem, resulting in a shortage of model performance due to the diverse set of features in UAVs on which the model has not been trained. Therefore, prior research works that have implemented machine learning approaches or proposed their algorithms and achieved high accuracies have often demonstrated their models only on specific datasets. However, the generalizability of these models to unseen datasets remains uncertain.

The rest of the paper is organized as follows. Section 2 explains the key concepts behind the implementation of the models, introduces the proposed image-based intrusion detection methodology, and provides a comprehensive description of the characteristics of the training and unseen datasets. Section 3 demonstrates the proposed model evaluation metrics, the step time, and the curves (learning and loss) for the best-performing proposed models. Also, it introduces and explains the model evaluation metrics for the best-fit models, presents the numerical results of the application of proposed models on an unseen dataset, and discusses the advantages of our proposed image-based intrusion detection system and the challenges encountered during the experimentation. Section 4 concludes the paper.

2. Methodology

2.1. System architecture

In this paper, we aim to develop an image-based intrusion detection method that employs deep learning models with highly accurate forecasting information of GPS spoofing cyberattacks. The forecasting information is obtained from a publicly available dataset, *UAV Attack Datasets*,¹ created by Whelan et al. [8]. This dataset consists of logs from both benign flights and flights that experienced GPS spoofing attacks in two different formats: comma-separated values (CSV) and universal log (ULOG) files.

In their dataset, the UAVs were equipped with an open-source flight controller named Pixhawk 4 flight controller running the PX4 Autopilot software [34]. The UAVs were launched in Shanghai, China, using the QGroundControl software, which allows the user to execute the mission and monitor it in real-time [35]. During the flight, the UAVs received authentic GPS signals from the Keysight EXG N5172B signal generator. At the same time, an agent simultaneously used the GPS-SDR-SIM tools to generate fake GPS signals [36]. HackRF transmitted the spoofed GPS signals with the same frequency band according to international regulations and with a higher power than the authentic signal so that the UAV could lock onto the spoofed GPS. Moreover, to guarantee a successful hijack, the agent broadcasted white Gaussian noise on the GPS frequency band to jam the GPS signals and prevent the UAVs from receiving authentic GPS signals. Further, each UAV flight had different flight dynamics and characteristics. The generated datasets are useful for evaluating the performance of the models under GPS spoofing and ping DoS attacks because they contain both simulation and live flights of different types of UAVs, such as Holybro S500, Yuneec H480, DeltaQuad VTOL, Standard Tailsitter, Standard Plane, and 3DR IRIS [32,33].

Since the UAVs had different flight dynamics and characteristics, our methodology consists of an image-based intrusion detection model

Table 1

Descriptive statistics of the UAV attack and unseen datasets.

Descriptive statistics	Training dataset	Unseen dataset
Number of UAVs	5 different UAVs	1 UAV
Number of features	85	268
Cyberattacks	GPS spoofing attack	GPS spoofing and Ping Dos attack
Mean	1.16003e+07	2.9569e+06
Median	2.9149e-02	3.6226e-04
Standard Deviation	1.5951e+08	8.2505e+07
Interquartile Range	1.2620	1.0
Variance	2.5445e+16	6.8072e+15

that uses a CNN architecture integrated with principal component analysis, data augmentation, and transfer learning to detect GPS spoofing attacks (Fig. 1). The proposed deep learning model architecture is designed not only to achieve a high classification accuracy on the training dataset but also to generalize well to other datasets relevant to GPS spoofing attack contexts. Therefore, our proposed model is evaluated by using many different classification metrics. Moreover, the best-fit model is tested on an unseen dataset to ensure successful generalization on a new dataset with different UAV features. Additionally, we also present a plot that combines learning and loss curves for training and validation before and after applying transfer learning.

The proposed image-based intrusion detection models consist of the following seven main stages:

- (1) **Data Collection:** Network traffic datasets are collected in one comma-separated values file;
- (2) **Data Preprocessing:** Data is preprocessed by removing duplicated rows, handling the missing values, converting raw data into a format that deep learning models can use for detection, reducing the number of features, and balancing the dataset;
- (3) **Data Representation:** Converting tabular data into three-dimensional array images and saving them as a PNG file;
- (4) **Data Splitting:** Splitting the train-test-validation split for the dataset as follows: An 80%-13.33%-6.67% train-validation-test split is used to generate a training set with 80% of the data samples, a validation set with 2/3 of the original 20% of the data samples, and a test set with 1/3 of the original 20% of the data samples;
- (5) **Data Augmentation:** A data augmentation process is applied randomly to each image in the training set in the following manner. The image is randomly flipped horizontally and/or vertically with a probability of 0.5; the image is randomly rotated up to 20% of its width and height; and the image is randomly zoomed in or out by up to 10%;
- (6) **Model Training:** CNN architecture models, such as MobileNetV2, Xception, VGG16, VGG19, ResNet152, InceptionResNetV2, and InceptionV3, are used with transfer learning;
- (7) **Model detection:** The detections for each CNN architecture model are evaluated, and the learning and loss curves for training and validation are demonstrated.

2.2. Data preprocessing

In order to obtain a high-performance IDS model for detecting GPS attacks, we used two datasets created by Whelan et al. [8] in our paper: (i) a training dataset containing about 85 sensor files with information on actuator controls, battery status, vehicle air data, and attitude, for five different UAVs; and (ii) an unseen dataset containing about 268 sensor files for one UAV named 3DR IRIS. The differences between the datasets are presented in Table 1.

However, the datasets have a few limitations:

- Some duplicate rows may cause bias and reduce the dataset's standard deviation. Therefore, it is necessary to remove these duplicated rows before conducting any further experiments.

¹ <https://ieee-dataport.org/open-access/uav-attack-dataset>

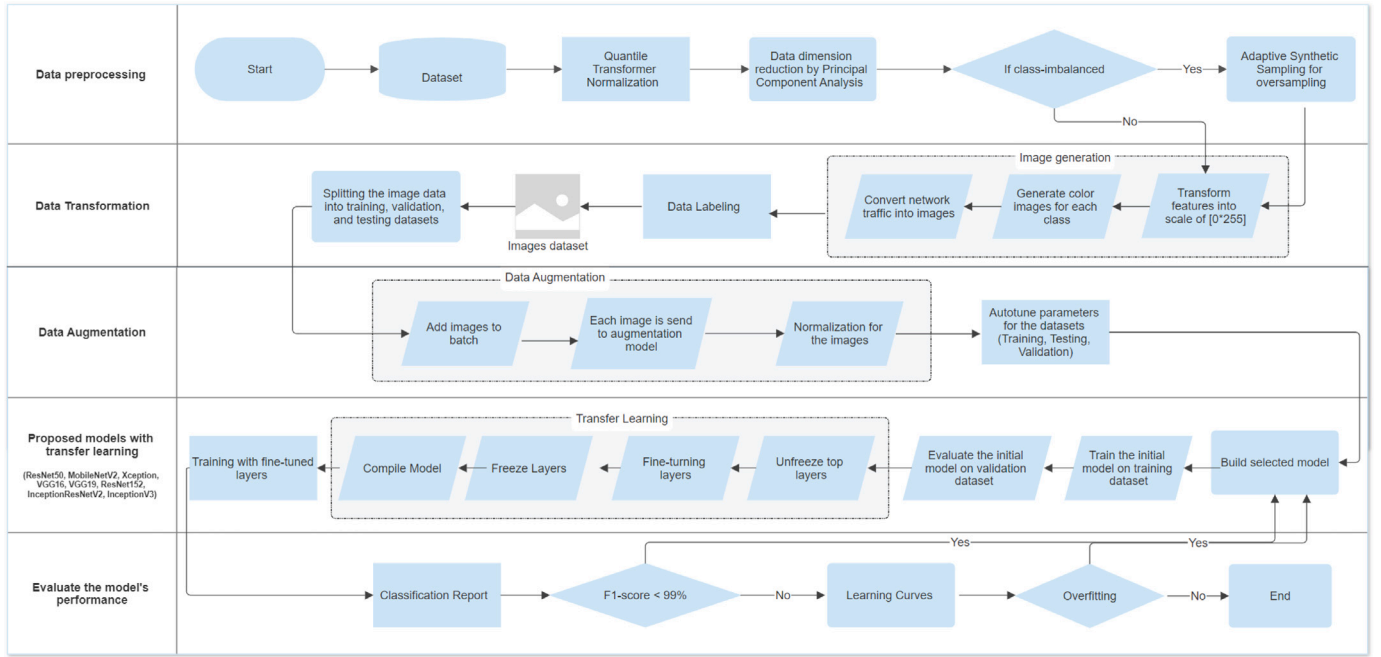


Fig. 1. Architecture of proposed image-based intrusion detection methodology.

- The datasets are vulnerable to high-class imbalance, which can significantly impact accuracy and have a high false positive rate.
- Both datasets have a high number of outliers in some features, such as *quaternion q[0,1,2,...]*, which describes the position of the UAVs in space; *delta_q_reset[0]*, which indicates the difference between previous and current quaternions; and *dist_bottom_valid*, which validates the distance to the ground.
- The training and testing datasets have a high and different number of features, 85 and 268, respectively, which increases the model complexity and chances of overfitting [37].

If the datasets are not correctly preprocessed to address and mitigate the limitations of the datasets, the input data might have a different format or shape than what is expected by the proposed models. The models might then take a long time to converge. Moreover, increased data complexity often leads to poor performances, with many models over- or under-fitting. Therefore, data preprocessing is one of the most critical stages, having a significant impact on the performance of our proposed model. The preprocessing stage involves converting raw data into a format that deep learning algorithms can use to generate predictions, eliminate noise and outliers, and reduce the impact of irrelevant features. Hence, to create an intrusion detection system model with high performance and the ability to generalize well to a new dataset, a few measures must be taken.

Raw datasets with different feature scales make it difficult for deep learning models to converge quickly or at all. Therefore, the dataset is normalized by *Quantile Transformer Normalization*, which was introduced around the early 2000s by Bolstad et al. to ensure that all the numeric features have a comparable range of 0 to 1 [38]. The labels of the datasets are converted into numerical values between 0 and $n - 1$ by using a *LabelEncoder* from the Sklearn library [39].

Because of the significant number of features and data logs in the dataset, it is worth reducing the dimensions of the data before training it with deep learning models to minimize the complexity and improve the performance of the model. Therefore, we implemented PCA method, which British mathematician Karl Pearson first introduced in 1901 [40] as a dimensional reduction technique to reduce the number of features into principal components while keeping the most relevant data [41]. PCA can be performed mathematically in many

ways; however, in our study, it was carried out by multiplying the standardized data with the matrix of the eigenvectors of the covariance matrix, resulting in a new dataset with fewer dimensions [42]. This involves (i) standardizing the range of the continuous initial variables to transform all the variables into the same scale to prevent a biased result; (ii) calculating the covariance matrix to address the relationship between the variables in the data as follows:

$$\frac{1}{n-1}(X - \bar{X})^T(X - \bar{X}) \quad (1)$$

where n is the number of samples, X is the data matrix, and \bar{X} is the mean vector; and (iii) calculating the eigenvectors with eigenvalues to determine the principal components based on the proportion of variance described by each component.

$$\Sigma v = \lambda v \quad (2)$$

Where Eq. (1) is used to obtain eigenvalues (λ) and corresponding eigenvectors (v) for obtaining the desired reduced dimensionality [43], these calculations are necessary for defining the number of principal components to transform the data into a new space [42]. We applied PCA to the dataset used in this paper, and we found that the optimum number of principal components is 32, as shown in a scree plot in Fig. 2.

Imbalanced datasets have an uneven or unequal distribution of observations, indicating that certain class labels may have a large number of observations and others have fewer, as shown in Fig. 3 for the UAV attack dataset. Imbalanced datasets are mainly solved by oversampling the minority class or undersampling the majority class, which can be achieved through techniques such as synthetic minority oversampling techniques (SMOTE) and SMOTE with Edited Nearest Neighbors (SMOTE-ENN) [44]. The core idea of the balancing methods is encapsulated in the following formula:

$$X_{\text{new}} = X_{\text{inst}} + \lambda \cdot (X_j - X_{\text{inst}}), \quad j = 1, 2, \dots, k \quad (3)$$

where $\lambda \in [0, 1]$ is a random number and X_j is a randomly selected sample from the set $\{X_1, X_2, \dots, X_k\}$ of k nearest neighbors of X_{ins} [45].

Random sampling is a basic technique that duplicates the instances and may cause over-fitting [46]. The more advanced technique, SMOTE-ENN, is a hybrid method that combines both oversampling and undersampling techniques to address the imbalance class. SMOTE-ENN

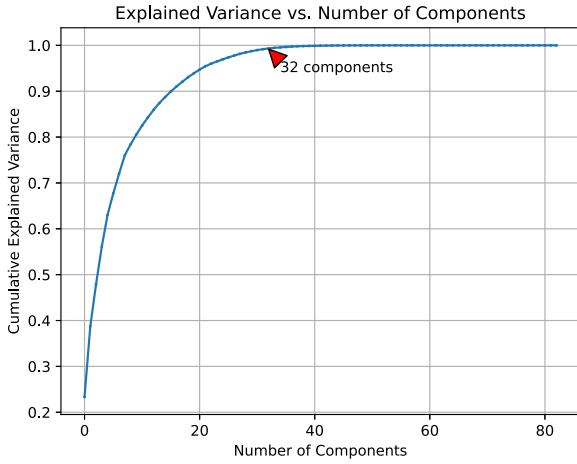


Fig. 2. Scree Plot of cumulative explained variance for principal component analysis. The optimum number of principal components for the dataset used in this paper is 32.

creates new instances for the minority classes and then employs the edited closest neighbors (ENN) algorithm technique to eliminate noisy and borderline samples by comparing each synthetic sample with its k -nearest neighbors (KNN) [44]. In other words, in SMOTE-ENN, SMOTE aims to reduce overfitting and, the ENN algorithm reduces noisy samples. However, removing noisy samples by using the ENN may also eliminate some informative samples, which results in a loss of valuable information and a low classification performance [47]. Therefore, we used the Adaptive Synthetic Sampling algorithm (ADASYN) method for oversampling imbalanced datasets, which addresses the limitations of the SMOTE and the SMOTE-ENN in the following manner:

- ADASYN generates more synthetic examples using a density distribution function in the difficult regions of the feature space, where the class distributions are sparse, dense, and more complex to learn [48];
- ADASYN performs better on highly imbalanced datasets, as demonstrated by the bar plot depicting results for the UAV attack dataset in Fig. 3. ADASYN's ability to randomly and diversely choose the nearest neighbors that have more influence on the synthetic sample can significantly reduce overfitting. This, in turn, leads to improved model generalization, a crucial factor when dealing with unseen datasets in real-world scenarios.

The key step in the ADASYN algorithm for synthetic sample generation is as follows [49]:

$$s_i = x_i + (x_{zi} - x_i) \times \lambda \quad (4)$$

where the process involves creating synthetic samples (s_i) based on adding a scaled difference between the original instance and its neighbor ($x_{zi} - x_i$), (x_i) is an instance in the n dimensional feature space X (original instance); x_{zi} is a randomly chosen neighbor from the minority class (selected neighbor); and λ is a random number that acts as a scaling factor used to scale the difference between the original instance and its random neighbor.

After preprocessing the dataset, we employed the network traffic image transformation technique, which is one of the data transformation methods used to convert the data into three channels of color images (red, green, and blue) because the CNN architecture models work effectively on images with an 8-bit depth (0 to 255) providing high-resolution data [50]. We carried out the network traffic image transformation by dividing the data samples into chunks based on the feature sizes. In our case, the training dataset has about 3072 features in an image. This is because each image is reshaped to be $(32 \times 32$

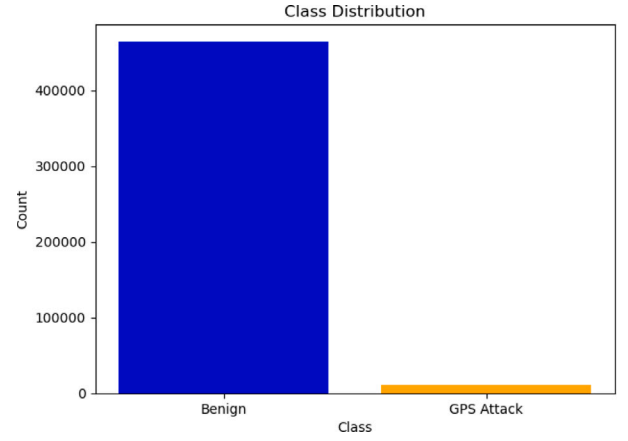


Fig. 3. Bar plot for the UAV dataset.

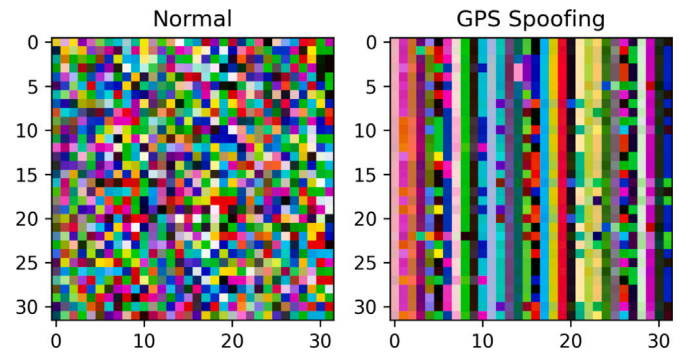


Fig. 4. Image generation for each class.

pixels) \times 3 color channels (RGB)), and each chunk has 32 consecutive data samples. The image generation for the Benign and GPS spoofing attacks is shown in Fig. 4.

The datasets used in our paper were split by the train-validation-test split method, which involves splitting the datasets into three subsets: training, validation, and test. The aim of using this method is to: (i) avoid the over-fitting issue where the model achieves a high accuracy only on the training dataset and a low accuracy on an unseen dataset. Therefore, by having a subset of the testing set, the model can be expected to perform well on an unseen dataset; (ii) avoid the under-fitting issue where the model learns poorly from the training dataset. By using a large subset of the training set, the model will learn from and capture the complexity of the dataset; and (iii) achieve the best trade-off between bias and variance by using the subset of the validation set.

Data augmentation is one of the methods that are used to increase the training data by generating additional training samples so that the model can generalize well to unseen datasets (e.g., by employing the Keras Sequential API). The common data augmentation techniques used in deep learning are rotating, flipping, scaling, cropping, or adding noise to the existing training data to improve the model performance and reduce overfitting [51]. In our studies, we used the augmentation method not only to prevent overfitting but also to create more training samples, which helped us generate more different GPS spoofing scenarios. This method leads us to have a proposed model that could generalize well to an unseen dataset, even if the numerical values, the UAV type, and the GPS spoofing attack scenarios are different from the dataset.

Loading datasets for training the deep learning models after the data preprocessing and augmentation can result in a high computational time. Therefore, the last critical step for reducing the computational

time is to use data pipeline optimization, which involves various techniques like parallel processing, caching, prefetching, and compression. This step aims to prepare and load the dataset without using a high capacity of the CPU and GPU resources while reducing the data transmission and the delay of memory access [52]. In our study, we reduced the computational time by using the prefetching technique that overlaps the data preprocessing and model training stages to speed up the training process. This step aims to fetch the elements of the dataset in advance so that they are ready to be utilized by the next step [53].

2.3. Transfer learning with convolutional neural network architectures

CNN architectures are a type of artificial neural network (ANNs) that have become a critical and popular architecture in deep learning, particularly for image classification tasks, as they can extract features from the input image and classify them into one of several predefined classes [54].

CNN architectures that are used for image classification tasks have several convolutional layers, followed by activation, pooling, dropout, and fully-connected layers. Convolutional layers are used to extract features from the input image by using a set of filters or kernels. Each filter applies a convolution operation to create a feature map that highlights particular features in the input image to generate accurate predictions. By stacking multiple convolutional layers, the network can learn increasingly complex features from the input image. Activation layers, such as the Rectified Linear Unit (ReLU), create nonlinearity in the network by selecting all the positive values in the feature map and setting all the negative values to zero. This allows the network to learn more features from images and improve its accuracy on image classification tasks. The pooling layer in the CNN is used to retain the most critical data while downsampling the size of the feature created by the convolutional layers. Limiting the number of parameters while keeping the most crucial data helps minimize the network's computational complexity and prevent overfitting. Dropout layers are used to avoid overfitting by randomly dropping some of the neurons in the network during training. Fully-connected layers are used to connect all the features extracted by the previous layers and generate the final output.

Most of the CNN models contain three layers: a bottom layer that learns basic features such as edges; a middle layer that learns more intricate patterns such as shapes; and a top layer that learns high-level representations tailored to the dataset and task. By utilizing a pretrained CNN model, we can fine-tune only the top layers for our task while still leveraging the knowledge feature patterns learned by the bottom and intermediate layers. This technique is called transfer learning, which enables a new model to leverage the knowledge acquired by a pretrained model that has been trained on a large dataset [55]. This knowledge is gained by transferring the weights of the pretrained model to the new model trained on another dataset [56]. Moreover, by fine-tuning a pretrained model, we can achieve high accuracies while using fewer computational resources. The fine-tuning phase converges more quickly in this case than when training the CNN model from scratch, and the risk of overfitting is reduced by preventing the model from remembering the training data. Therefore, in the context of CNN image classification, transfer learning has been successfully applied to many image processing tasks because the captured patterns trained by the CNN model (pretrained model) on a large dataset are applicable to many different tasks.

ImageNet is a large-scale dataset that has more than one million labeled images belonging to 1000 different classes, and it is commonly used as a benchmark for image classification tasks. CNN architectures that are trained on the ImageNet dataset are considered some of the best models for image classification tasks due to their ability to learn from the complex features of raw image pixels and the use of transfer learning, which enables them to be fine-tuned for a wide

range of tasks [55]. Therefore, we used eight CNN architectures—MobileNetV2, VGG16, VGG19, InceptionV3, InceptionResNetV2, Xception, and ResNet152—and compared their performances. MobileNetV2 is a lightweight architecture that has achieved an accuracy of 71.8% on the ImageNet dataset. It has 155 layers consisting of depthwise separable convolutions to reduce the number of parameters and linear bottlenecks to reduce the computational time and cost required to train the model; the last two layers are fully connected layers. VGG16 and VGG19 were developed by the Visual Geometry Group (VGG) at the University of Oxford. Both the models have achieved a top-5 error rate of 7.4% and 6.8% and an accuracy of 71.5% and 71.1%, respectively, on the ImageNet dataset. The key difference is that VGG19 has more filters in each layer, which makes it a more complex model. InceptionV3, developed by Google in 2015, has 48 convolutional layers and three fully connected layers, and it has achieved an accuracy of 78.0% on the ImageNet dataset. It uses the batch normalization technique that normalizes the activations of each layer to improve the accuracy of the model and reduce the overfitting issues of the model. The InceptionResNetV2 model was introduced by Google in 2016; it is an updated version of InceptionV3. The key difference is that InceptionV3 has 48 layers, while InceptionResNetV2 has 572 layers, which makes it possible to capture more complex features in the images. Xception is an extension of Inception that uses depthwise separable convolutions and a global average pooling layer at the end of the model to reduce overfitting and improve generalization. It has achieved an accuracy of 79.0% on the ImageNet dataset. ResNet152 is a deep architecture that has 152 layers and a total of 60.4 million parameters. It has achieved an accuracy of 78.6% on the ImageNet dataset and uses residual connections to address the vanishing gradient problem. The design of the bottleneck structures of those algorithms reduces the dimensionality of the feature maps, reduces the computational cost, and the risk of overfitting [57].

2.4. Performance metrics and system specifications

To evaluate the performance of the proposed deep learning models on a highly imbalanced dataset for predicting GPS spoofing attacks, we focused on the following:

- We studied the following classification metrics to measure how well the proposed models will classify the spoofed GPS signal events:
 - (1) accuracy (ACC), which shows the accuracy of classification for each stage;
 - (2) detection rate (DR/recall), which is the ratio between the detected attack data and the total abnormal data;
 - (3) harmonic precision–recall mean (F1-score), which is used as a statistical measure to rate the model performance as it depends on two factors, precision (PRE) and DR/recall;
 - (4) receiver operating characteristic area under the curve (ROC AUC), which measures how the proposed models will perform to know the difference between authentic and spoofed GPS signals (positive and negative classes);
 - (5) precision–recall area under the curve (PR AUC), which measures the trade-off (balance) between precision and recall;
 - (6) balanced accuracy (BAC), which is the average recall of the benign and spoofed GPS signals and is a useful metric for imbalanced datasets;
 - (7) Matthews correlation coefficient (MCC), which is a statistical tool based on chi-square statistics used to measure the difference between the predicted values and the actual values; and
 - (8) Jaccard score (JSC), which measures the similarity between the labels, i.e., benign and spoofed GPS signals.
- We employed two loss function metrics: (1) logarithmic loss function (Log Loss), which measures how good the performance of a model is by computing the differential between the predicted

Table 2
Partial transfer learning configuration and running time for the CNN models.

Models	Total layers	Frozen layers	Unfrozen layers	RT (s)	DT (s)
ResNet-50	175	1 to 150	151 to 175	117.43	1.951
MobileNet-v2	154	1 to 132	133 to 154	77.3	1.289
Xception	132	1 to 113	114 to 132	120.41	2.035
VGG-16	19	1 to 16	17 to 19	120.85	3.241
VGG-19	22	1 to 18	19 to 22	120.64	3.086
ResNet-152	515	1 to 443	444 to 515	309	5.102
InceptionResNet-v2	780	1 to 671	672 to 780	181.4	9.203
Inception-v3	311	1 to 267	268 to 311	106.4	1.964

probabilities (the outputs of the model) and the actual values (true labels), and (2) zero-one loss, which measures how many times the predicted label does not match the true label.

- We used the learning and loss curves for training and validation to monitor the model's performance, i.e., whether it is overfitting, underfitting, or is a good fit. This allows us to track the deterioration of the proposed models' learning performance.
- We used the confusion matrix to summarize the performance of the CNN models on a set of test and validation datasets. This is critical when dealing with imbalanced datasets and helps us to examine the model prediction errors by monitoring the false positive and false negative values.
- We computed the models' running time using the "time" library and the elapsed time per step (step time) to calculate how long the models need to process one batch during the testing phase.
- Moreover, we computed the time taken for a single prediction (Atomic prediction latency), and the number of predictions that our proposed image-based intrusion detection system can process, measured in predictions per second (p/s) (Prediction Throughput). These metrics were measured using a CPU.
- We evaluated the generalization of our models by testing them on unseen datasets.

The deep learning models proposed in this study were implemented on the UAV attacks dataset using Keras libraries and Scikit-learn. The models were trained using the computational resources provided by CSC - IT Center for Science Ltd. This nonprofit organization offers IT services for research, education, and public administration in Finland. The computer configuration utilized for these experiments consists of Xeon Gold 6230 (2 x 20 cores @ 2.1 GHz) with an NVIDIA V100 (Tesla V100-SXM2 GPU) and 300 GB of memory.

3. Results and discussion

To construct a generalized robustness model that avoids overfitting, we trained the CNN models with transfer learning to exploit the knowledge and features gained from the significant number of images and classes in the ImageNet classification dataset. In our study, we applied partial transfer learning by freezing the lower layers of the pretrained model and fine-tuning the higher layers for GPS spoofing attack tasks.

After the partial transfer learning was implemented, the overall performance of the proposed models improved by approximately 0.8% to 2.08%. The partial transfer learning configuration and running time for the proposed models are presented in Table 2.

The results of evaluating the proposed models for both the testing and validation datasets presented in Table 3 demonstrated that all the models achieved high accuracies. Even though the proposed models and the models in the literature achieved high accuracies, this does not necessarily mean that they could generalize well to an unseen dataset. Therefore, we provide the learning and loss curves for our best-performing models to help us monitor the improvement or deterioration of the learning performance of the model by plotting both the training and validation loss and accuracy versus epochs, as shown in Figs. 5, 6, and 7.

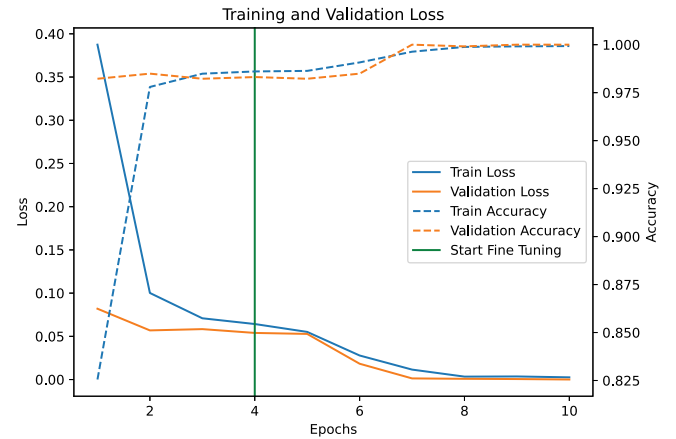


Fig. 5. Learning and loss curves for the ResNet-50 model with an 117.43 s runtime.

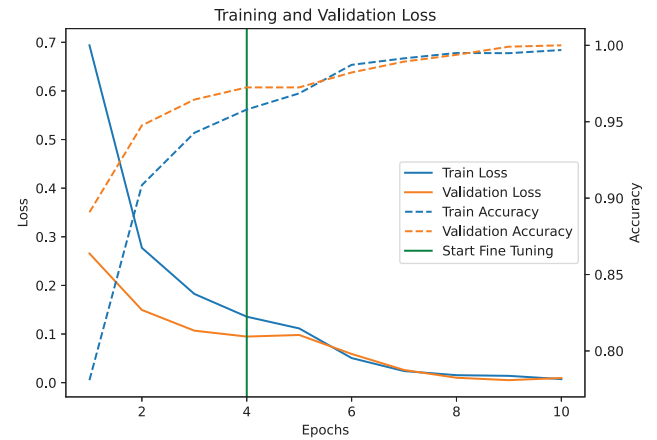


Fig. 6. Learning and loss curves for the VGG-19 model with an 120.64 s runtime.

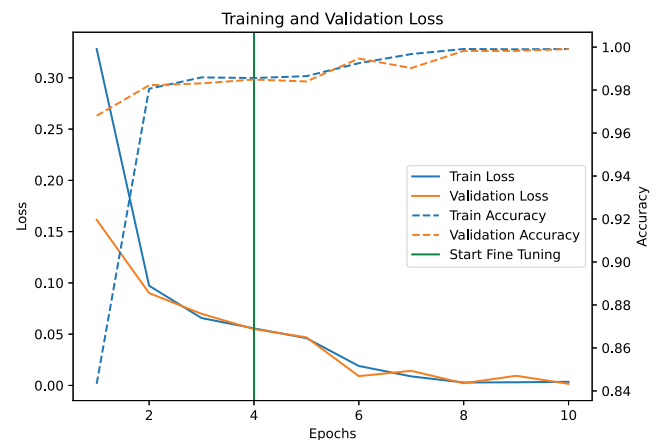


Fig. 7. Learning and loss curves for the ResNet-152 model with a 309 s runtime.

The green vertical line in the learning and loss curves serves as a visual indicator of the partial transfer learning on the proposed models. The curves before the green line indicate that the models are overfitting when the training occurs without partial transfer learning. After the green line, the training incorporates partial transfer learning, and the models are no longer overfitting. Thus, the learning and loss curves emphasize the influence of partial transfer learning on the model's overall performance; this is also shown in Table 3.

Although we employed many classification metrics to evaluate our proposed models, we focused on the F1-score metric and confusion matrix to evaluate the performance of our models. In most real-world classification situations, the data might have an imbalanced class distribution, which can mislead other metrics, such as accuracy. Still, the F1-score metric and the confusion matrix provide a balanced evaluation. The F1-score considers both precision and recall metrics, thus making it a better indicator of a model's ability to generalize well to unseen data, and the confusion matrix provides a precise evaluation of how well the proposed models will perform in terms of correct or incorrect predictions.

The evaluation of the proposed models for both testing and validation datasets of the UAV attack dataset, as presented in Table 3, demonstrates that all the models achieved high accuracies, and the VGG-19 and ResNet-152 models emerged as the best on the UAV attack dataset for the following reasons:

- The VGG-19 and ResNet-152 models successfully identified a large number of actual GPS attacks that occurred (positive cases). The true positives in the confusion metrics indicate that the VGG-19 and ResNet-152 models detected 319 and 334 actual cyberattacks for the testing dataset, respectively, and 657 and 645 for the validation dataset, respectively. On the other hand, the confusion metrics for the other models, such as VGG-16 and Inception-v3, indicated that the models failed to detect some GPS spoofing attacks.
- The training logs indicate that the proposed VGG-19 and ResNet-152 models have low training and validation losses of 0.0027 and 0.0032, respectively, and 0.0019 and 0.0016, respectively. These results suggest that both models performed well on the training and validation datasets.
- The Matthews correlation coefficient and the loss function metrics, such as log loss and zero-one, indicate that the model achieved perfect prediction by learning the underlying patterns and could generalize well to a different dataset.
- The learning and loss curves for both models indicate that the models are not overfitting, as shown in Figs. 6 and 7.

Achieving a high performance rate, as shown in the classification report in Table 3, is a desirable outcome, but it does not indicate a complete evaluation of the model's performance. A comprehensive evaluation involves considering the model's ability to generalize well and to identify complex patterns and features of different underrepresented attacks, such as GPS spoofing attacks, apart from the normal network behavior (benign traffic). As a result, we also tested our models with an unseen dataset that contains 268 features for one UAV named 3DR IRIS+. The differences between the training and the unseen datasets are listed in Table 1.

The evaluation of the proposed models on the unseen datasets, as presented in Table 4, highlights the superior performance of the VGG-19 and ResNet-152 models, which achieved the highest accuracies and F1-scores. However, the confusion metrics demonstrated that the models failed in detecting instances of Ping DoS attacks. This is because our models were specifically designed to detect GPS spoofing attacks, not Ping DoS attacks. The learning curve for ResNet-50 also revealed that the model did not exhibit overfitting (Fig. 5), but the model exhibited suboptimal performance on unseen datasets as compared to ResNet-152. This is because ResNet-152 has more layers, parameters

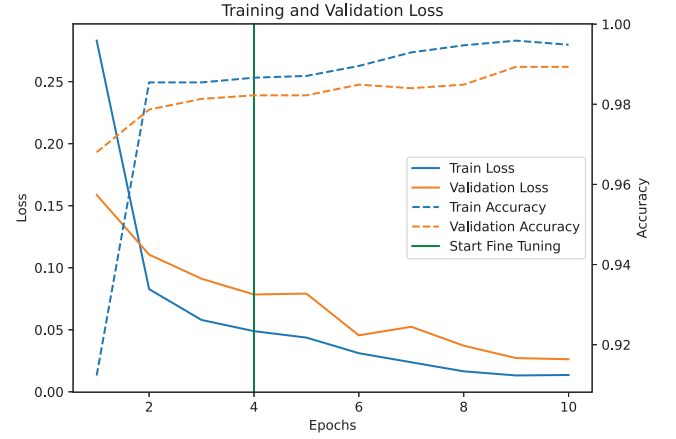


Fig. 8. Learning and loss curves for the Inception-v3 model with an 106.4 s runtime indicated overfitting.

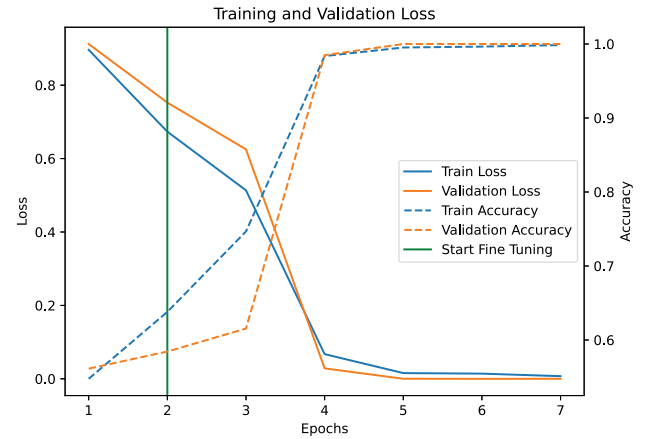


Fig. 9. Learning and loss curves for the Inception-v3 model with Hyperopt with a 75.1 s runtime indicated a good-fit model.

(152 layers and 60 million parameters), and a bottleneck structure in each residual block compared to ResNet-50 (50 layers and 25 million parameters) [58], which allows the model to generalize well [59].

At the same time, the inception-v3 model showed poor performance on the unseen dataset with an accuracy of 46.437% and an F1-score of 29.452%, as presented in Table 4. This is due to the fact that the learning and loss curves demonstrated that the model is overfitting (Fig. 8), although it achieved a high accuracy of 99.6% on the training dataset, as shown in Table 3. Therefore, it is critical to monitor the learning and loss curves even if the model achieves high evaluation metrics on the training dataset. Hence, the parameters for Inception-v3 are tuned using a commonly used approach, the Bayesian optimization technique with the Tree-structured Parzen Estimator (BO-TPE) algorithm in the Hyperopt library because of its sample efficiency [60]. The chosen parameters for the Inception-v3 model are shown in Table 5. After implementing Hyperopt (BO-TPE), the accuracy of Inception-v3 has increased to 99% as shown in Table 6.

From the learning and loss curve for the Inception-v3 model in Fig. 8, the generalization gap between the validation and training losses indicates a lack of convergence. Therefore, to improve the learning and loss curve, we added the regularization parameter “weight decay” into the hyperparameter tuning process using the BO-TPE algorithm in Hyperopt, as shown in Table 5. As a result, the generalization gap has decreased, as shown in Fig. 9 (compared with Fig. 8).

Our novel approach involves an image-based intrusion detection system that leverages deep learning algorithms with transfer learning. This system is designed to detect GPS spoofing with exceptional

Table 3

Classification report for the testing and validation datasets of the UAV attack dataset.

Classification report for the testing dataset with transfer learning								
Classification Metrics	Models							
	ResNet-50	MobileNet-v2	Xception	VGG-16	VGG-19	ResNet-152	Inception ResNet-v2	Inception-v3
Accuracy (%)	99.652	91.184	99.652	99.479	100	100	99.652	99.652
Precision (%)	99.654	93.127	99.654	99.483	100	100	99.654	99.654
Recall (%)	99.657	91.840	99.652	99.479	100	100	99.652	99.652
F1-score (%)	99.656	91.871	99.652	99.478	100	100	99.652	99.652
F1-score for each type of attack (%)	[99.690, 99.604]	[92.205, 91.438]	[99.705, 99.576]	[99.554, 99.373]	[100, 100]	[100, 100]	[99.693, 99.6]	[99.698, 99.590]
Confusion Matrix	[[322 0] [2 252]]	[[278 47] [0 251]]	[[339 0] [2 235]]	[[335 0] [3 238]]	[[319 0] [0 257]]	[[334 0] [0 242]]	[[325 0] [2 249]]	[[331 0] [2 243]]
ROC AUC (%)	99.606	92.769	99.578	99.377	100	100	99.601	99.591
PR AUC (%)	99.559	84.228	99.503	99.276	100	100	99.550	99.538
Balanced Accuracy (%)	99.606	92.769	99.578	99.377	100	100	99.601	99.591
Matthews Correlation Coefficient	0.99297	0.84880	0.99284	0.98933	1.0	1.0	1.0	0.99291
Jaccard Score	0.99212	0.84228	0.99156	0.98755	1.0	1.0	0.99203	0.99183
Log Loss	0.12515	2.94106	0.12515	0.18772	2.224e−16	2.220e−16	2.220e−16	0.12515
Zero-One Loss	0.00347	0.08159	0.00347	0.00520	0.0017	0.0	0.0	0.00347
Specificity	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Step time (s)	1.35	0.19	1.43	1.46	2.90	1.79	2.78	1.31
Classification Report for validation dataset with transfer learning								
Classification Metrics	Models							
	ResNet-50	MobileNet-v2	Xception	VGG-16	VGG-19	ResNet-152	Inception ResNet-v2	Inception-v3
Accuracy (%)	100	90.674	99.289	99.555	100	99.911	99.467	99.111
Precision (%)	100	92.234	99.298	99.555	100	99.911	99.472	99.125
Recall (%)	100	90.674	99.289	99.555	100	99.911	99.467	99.111
F1-score (%)	100	90.722	99.288	99.555	100	99.911	99.466	99.117
F1-score for each type of attack (%)	[100, 100]	[91.154, 90.140]	[99.383, 99.161]	[99.612, 99.480]	[100, 100]	[99.922, 99.895]	[99.536, 99.373]	[99.223, 98.962]
Confusion Matrix	[[642 0] [0 484]]	[[541 105] [0 480]]	[[645 0] [8 251]]	[[642 0] [5 479]]	[[657 0] [0 469]]	[[645 0] [1 480]]	[[644 0] [6 476]]	[[639 0] [10 477]]
ROC AUC (%)	100	91.873	99.168	99.483	100	99.896	99.377	98.973
PR AUC (%)	100	82.051	99.047	99.410	100	99.880	99.288	98.834
Balanced Accuracy (%)	100	91.873	99.168	100	99.483	99.896	99.377	98.973
Matthews Correlation Coefficient	1.0	0.82894	0.98555	0.99096	1.0	0.99818	0.98915	0.982025
Jaccard Score	1.0	0.820512	0.98336	0.98966	1.0	0.99792	0.98755	0.979466
Log Loss	2.2204e−16	3.36108	0.25608	0.16005	2.2204e−16	0.03201	0.19206	0.32010
Zero-One Loss	0.0	0.0932	0.007104	0.00444	0.00173	0.00088	0.00532	0.00888
Specificity	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Step Time (s)	1.34	1.19	2.42	3.42	4.53	3.77	9.129	1.30
Computational Performance								
Atomic prediction latency (ms)	0.4117	0.3529	0.3882	0.3882	0.3529	0.4352	0.4235	0.3882
Prediction Throughput (p/s)	2428.79	2833.40	2575.92	2576.01	2833.50	2297.54	2361.33	2575.90

accuracy while maintaining low computational performance. The key innovation is the development of a single algorithmic learning model that can perform effectively across different types of heterogeneous UAVs. Based on the results demonstrated in [Tables 3](#) and [4](#), our best proposed model for detecting GPS spoofing attacks is the VGG-19 model.

The proposed model achieved an accuracy of 100% on the training dataset, with a 120.64 s runtime, 3.08 s detection time, and no over-fitting issues. Moreover, it demonstrated robust generalization with a

99.25% accuracy and a 2.72 s detection time on an unseen dataset. This high accuracy confirms the model's effectiveness in detecting GPS spoofing attacks across different datasets for different types of heterogeneous UAVs. The computational performance for the VGG-19 model on the UAV attack dataset and unseen dataset demonstrates that the time taken for the model to process and return the results for a single prediction (atomic prediction latency) is 0.3529 ms and 0.5150 ms, respectively. Our proposed image-based intrusion detection system can handle and process approximately 2833.50 predictions per second for

Table 4

Classification report for the proposed models on unseen datasets.

Metrics for unseen dataset	Models			
	ResNet-50	ResNet-152	VGG-19	Inception-v3
Acc (%)	75.250	99.0	99.25	46.437
PR (%)	83.673	98.018	98.260	21.564
Rec (%)	75.250	99.0	99.25	46.437
F1 (%)	74.416	98.999	99.249	29.452
F1\attack (%)	[71.345, 78.217]	[99.083, 98.899]	[99.326, 99.153]	[63.422, 0.0]
MCC	0.59248	0.98003	0.98491	0.0
Jacc	0.64227	0.97823	0.98321	0.0
Log Loss	8.92080	0.36043	0.27032	19.305
ZOL	0.24750	0.01000	0.00749	0.53562
CM	[[493 392] [4 711]]	[[865 0] [16 719]]	[[885 0] [12 703]]	[[743 0] [857 0]]
ST (ms)	3.34	7.78	3.48	3.31
DT (s)	2.453	5.781	2.721	2.142
Latency (ms)	0.5474	0.5774	0.5150	58.9979
Through (p/s)	1830.70	1731.78	1941.62	609.85

Table 5

Hyperparameter (BO-TPE) optimization configuration for Inception-Residual Version 3 model.

Hyperparameter	Search Range	Optimal Value
Frozen	[1, 780]	50
Epochs	[1, 20]	7
Patience	[2, 5]	4
Learning Rate	[0.00003, 0.00009]	8e−05
Dropout	[0.2, 0.5]	0.4
Weight Decay	[1e−9, 1e−3]	5.578e−4
Optimizer	[Adam, SGD, RMSprop]	Adam
RT (s)	2184	

the UAV attack dataset and 1941 predictions per second for the unseen dataset (prediction throughput). This indicates that the proposed model can quickly analyze and determine the authentic GPS signal and the spoofed GPS signals and can handle a large volume of multiple UAVs or GPS signals that need to be monitored simultaneously. Further, to ensure a comprehensive and valid comparison with other methods, we compared our performance method with different methods by using the same dataset (UAV attack dataset) for GPS spoofing cyberattacks. We used the same evaluation metrics as in Whelan et al. [33], including Precision, Recall, and F1-score, as shown in Table 7.

The potential limitations and challenges that we faced are due to the variation in UAV types, feature sets, the number of features, and other differences. This poses the challenge of determining a strategy to develop a single model that can effectively detect GPS spoofing attacks across different UAV types and GPS spoofing scenarios. For that, most researchers often do not test their models on unseen datasets due to the diversity of UAVs. Also, selecting an appropriate technique to build a high-performance model that is highly accurate with low running and detection time, without overfitting issues, and generalizes well to unseen datasets is a formidable challenge. The fact that there is a multitude of models, diverse feature reduction methods, such as random forest regressor (RFR) and PCA, and different data normalization techniques, such as min–max, Z-score, and quantum transformer normalization, makes the decision-making process for selecting the optimal method suitable for the dataset complex and intricate. Transfer

Table 6

Classification report for the Inception-v3 model on unseen datasets before and after Hyperopt (BO-TPE) optimization.

Metrics	Inception-v3 Non-optimized	Inception-v3 optimized by HBO-TPE algorithm
ACC (%)	46.437	99.06
PR (%)	21.564	99.077
Rcc (%)	46.437	99.062
F1-score (%)	[63.422, 0.0]	[99.168, 98.926]
F1\attack (%)	0.0	98.112
MCC	0.0	98.112
Jacc	0.0	0.9787
Log Loss	19.3058	0.0093
ZOL	0.53562	0.009375
CM	[743 0] [857 0]]	[894 0] [15 691]]
ST (s)	3.31	3.30
DT (s)	2.931	2.478
Latency (ms)	58.9979	0.5618
Through (p/s)	609.85	1779.87
Number of prediction	8511	6415

Table 7

Comparison between the latest proposed based on UAV attack dataset.

Metrics	Classes	Autoencoder [33]	VGG-19
PR (%)	Benign	99.36	100
	Malicious	74.72	100
Rcc (%)	Benign	94.81	100
	Malicious	96.18	100
F1-score (%)	Benign	97.03	100
	Malicious	84.10	100
Latency (ms)		0.4175	0.3529
Through (p/s)		2340	2833.50

learning could also lead to poor performance or overfitting if many layers have been fine-tuned. It is crucial to select pretrained models carefully, as using a complex one may require high computational resources if the features learned by the pretrained model are irrelevant for detecting GPS spoofing attacks.

Our proposed image-based intrusion detection system for detecting GPS spoofing cyberattacks attempts to overcome these limitations by visually encoding the UAV attack dataset from multivariate numerical data into visual formats (images). We will be able to integrate multiple numerical features into a single coherent image, making it easier for the model to capture complex interactions. This allows us to learn and generalize the patterns for various GPS spoofing scenarios across different heterogeneous UAVs. Different GPS spoofing scenarios are generated by data augmentation since this technique is more accessible when applied in the image domain. Augmentations such as rotation, scaling, and flipping have improved our proposed model's robustness and generalization to unseen data, as mentioned in Section 2.2.

In addition to the main contributions presented in Section 1.2, the objectives and novel contributions of the paper are not limited to only improving the classification accuracy but also the following:

- We demonstrated that achieving high accuracy with an overfitting model, as shown in Table 3 and Fig. 8, cannot generalize well to a new dataset, as illustrated in Table 4 in Section 3. Also, we demonstrated that despite models showing high accuracy and good learning curves on the training data, the model may still struggle with unseen datasets. This is due to having fewer layers and parameters, which may be insufficient to capture the complex patterns required for effective generalization. Therefore, testing the model on an unseen dataset is essential.
- As part of our research methodology, we meticulously considered the classification metrics and learning curves. In Section 3, we explained how to reduce the overfitting and change the characteristics of the validation loss curve for the Inception-v3 model by implementing Hyperopt (BO-TPE) as shown in Fig. 9 (compared with Fig. 8).
- We have demonstrated proposed model solutions that can effectively detect GPS spoofing attacks for heterogeneous UAVs with high accuracy. This application of our research underscores its real-world relevance and potential impact in the field.

4. Conclusion

UAVs are attracting significant interest among both military and civilian industries due to their ability to perform critical tasks in high-risk environments. At the same time, cyberattacks such as GPS spoofing attacks are also increasing significantly, constituting a serious threat to human lives. In this paper, we have proposed an image-based intrusion detection system for detecting GPS spoofing attacks based on CNN architecture models with transfer learning. Our method was able to successfully learn and identify normal sensor values based on historical flight logs. The proposed models were evaluated on imbalanced datasets that include many different UAVs with different flight dynamics and characteristics affecting the GPS receiver's performance. The numerical results demonstrated that the selected methods enhance the performance of the model, leading to improved detection capabilities when tested on unseen datasets. The experimental results showed that the VGG-19 and ResNet-152 models with transfer learning were highly effective on the unseen dataset, achieving an accuracy of 99.25% and 99%, respectively. However, the computational performance demonstrated that our proposed image-based intrusion detection system based on the VGG-19 model could handle and process approximately 2833.50 predictions per second for the UAV attack dataset and 1941 predictions per second for the unseen dataset (prediction throughput), which is higher in terms of efficiency and speed compared to ResNet-152 model. Regarding the false negatives, the VGG-19 model had the

lowest count, with 12 instances compared to other models. Inception-v3 initially underperformed on the unseen dataset with an accuracy of 46.43%. Remarkably, through Bayesian optimization, with Tree-structured Parzen Estimator and transfer learning on the Inception-v3 model, the testing accuracy on the unseen dataset has reached 99.06%, suggesting that the chosen hyperparameters are more effective in capturing the underlying patterns in the data while avoiding overfitting. Conversely, the ResNet-50 model achieved a high accuracy of 99.65% on the UAV attack dataset and demonstrated a promising learning curve; its accuracy dropped to 75.25% on the unseen dataset due to its shallower architecture compared to the ResNet-152 and Inception-v3 deeper architecture models. Nonetheless, the overall results of the proposed models, including VGG-19, ResNet-152, and Inception-v3, demonstrated their feasibility for application to real-time data. In the future, we plan to focus on different datasets and applications using various approaches, such as an online adaptive model that can update itself on new available datasets. This will enable the model to remain accurate and relevant to new datasets.

CRedit authorship contribution statement

Mohamed Selim Korium: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Mohamed Saber:** Writing – review & editing. **Ahmed Mahmoud Ahmed:** Software. **Arun Narayanan:** Investigation, Methodology, Writing – review & editing. **Pedro H.J. Nardelli:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Pedro Nardelli reports financial support was provided by Academy of Finland.

Data availability

<https://iee-dataport.org/open-access/uav-attack-dataset>.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.adhoc.2024.103597>.

References

- [1] T.F. Watts, I. Bode, Machine guardians: The terminator, AI narratives and US regulatory discourse on lethal autonomous weapons systems, *Coop. Confl.* 59 (1) (2024) 107–128.
- [2] D.D. Nguyen, J. Rohacs, D. Rohacs, Autonomous flight trajectory control system for drones in smart city traffic management, *ISPRS Int. J. Geo-Inf.* 10 (5) (2021) 338.
- [3] I. Guvenc, F. Koohifar, S. Singh, M.L. Sichertu, D. Matolak, Detection, tracking, and interdiction for amateur drones, *IEEE Commun. Mag.* 56 (4) (2018) 75–81.
- [4] G. Nacouzi, J.D. Williams, B. Dolan, A. Stickells, D. Luckey, C. Ludwig, J. Xu, Y. Shokh, D.M. Gerstein, M. Decker, Assessment of the proliferation of certain remotely piloted aircraft systems: response to section 1276 of the national defense authorization act for fiscal year 2017, RAND Corporation Santa Monica, CA, 2018.
- [5] N. Norhashim, N.M. Kamal, Z. Sahwee, S.A. Shah, D. Sathyamoorthy, The effects of jamming on global positioning system (GPS) accuracy for unmanned aerial vehicles (UAVs), in: 2022 International Conference on Computer and Drone Applications, (ICONDA), IEEE, 2022, pp. 18–22.
- [6] S. Liu, X. Cheng, H. Yang, Y. Shu, X. Weng, P. Guo, K.C. Zeng, G. Wang, Y. Yang, Stars can tell: A robust method to defend against {GPS} spoofing attacks using off-the-shelf chipset, in: 30th USENIX Security Symposium, (USENIX Security 21), 2021, pp. 3935–3952.

- [7] Y.V. Yasyukevich, B. Zhang, V.R. Devanaboyina, Advances in GNSS positioning and GNSS remote sensing, *Sensors* 24 (4) (2024) 1200.
- [8] J. Whelan, A. Almelhadi, J. Braverman, K. El-Khatib, Threat analysis of a long range autonomous unmanned aerial system, in: 2020 International Conference on Computing and Information Technology, (ICCIIT-1441), IEEE, 2020, pp. 1–5.
- [9] D.P. Shepard, J.A. Bhatti, T.E. Humphreys, A.A. Fansler, Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks, in: Proceedings of the 25th International Technical Meeting of the Satellite Division of the Institute of Navigation, (ION GNSS 2012), 2012, pp. 3591–3605.
- [10] K.C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, Y. Yang, All your {GPS} are belong to us: Towards stealthy manipulation of road navigation systems, in: 27th USENIX Security Symposium, (USENIX Security 18), 2018, pp. 1527–1544.
- [11] J. Gaspar, R. Ferreira, P. Sebastião, N. Souto, Capture of UAVs through gps spoofing using low-cost SDR platforms, *Wirel. Pers. Commun.* 115 (2020) 2729–2754.
- [12] S. Liaquat, M. Faizan, J.N. Chattha, F.A. Butt, N.M. Mahyuddin, I.H. Naqvi, A framework for preventing unauthorized drone intrusions through radar detection and GPS spoofing, *Ain Shams Eng. J.* 15 (5) (2024) 102707.
- [13] Y. Gao, G. Li, A new asynchronous traction signal spoofing algorithm for PLL-assisted DLL receiver, *GPS Solut.* 27 (3) (2023) 141.
- [14] A.J. Kerns, D.P. Shepard, J.A. Bhatti, T.E. Humphreys, Unmanned aircraft capture and control via GPS spoofing, *J. Field Robotics* 31 (4) (2014) 617–636.
- [15] M.L. Psiaki, T.E. Humphreys, GNSS spoofing and detection, *Proc. IEEE* 104 (6) (2016) 1258–1270.
- [16] M.L. Psiaki, T.E. Humphreys, Protecting GPS from spoofers is critical to the future of navigation, *IEEE spectrum* 10 (2016).
- [17] K.-C. Kwon, D.-S. Shim, Performance analysis of direct gps spoofing detection method with ahrs/accelerometer, *Sensors* 20 (4) (2020) 954.
- [18] J.-H. Lee, K.-C. Kwon, D.-S. An, D.-S. Shim, GPS spoofing detection using accelerometers and performance analysis with probability of detection, *Int. J. Control Autom. Syst.* 13 (2015) 951–959.
- [19] Z. Feng, N. Guan, M. Lv, W. Liu, Q. Deng, X. Liu, W. Yi, An efficient UAV hijacking detection method using onboard inertial measurement unit, *ACM Trans. Embedded Comput. Syst.* 17 (6) (2018) 1–19.
- [20] Y. Dang, C. Benzaid, B. Yang, T. Taleb, Deep learning for GPS spoofing detection in cellular-enabled UAV systems, in: 2021 International Conference on Networking and Network Applications, (NaNA), IEEE, 2021, pp. 501–506.
- [21] S. Dasgupta, T. Ghosh, M. Rahman, A reinforcement learning approach for global navigation satellite system spoofing attack detection in autonomous vehicles, *Transp. Res. Rec.* 2676 (12) (2022) 318–330.
- [22] S.C. Bose, GPS spoofing detection by neural network machine learning, *IEEE Aeronaut. Electron. Syst. Mag.* 37 (6) (2021) 18–31.
- [23] S. Wang, J. Wang, C. Su, X. Ma, 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.
- [24] S. Semanjski, A. Muls, I. Semanjski, W. De Wilde, Use and validation of supervised machine learning approach for detection of GNSS signal spoofing, in: 2019 International Conference on Localization and GNSS, (ICL-GNSS), IEEE, 2019, pp. 1–6.
- [25] Y. Dang, C. Benzaid, B. Yang, T. Taleb, Deep learning for GPS spoofing detection in cellular enabled unmanned aerial vehicle systems, 2022, arXiv preprint arXiv:2201.00568.
- [26] G. Aissou, H.O. Slimane, S. Benouadah, N. Kaabouch, 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.
- [27] A. Gasimova, T.T. Khoei, N. Kaabouch, 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.
- [28] M.R. Manesh, J. Kenney, W.C. Hu, V.K. Devabhaktuni, N. Kaabouch, Detection of GPS spoofing attacks on unmanned aerial systems, in: 2019 16th IEEE Annual Consumer Communications & Networking Conference, CCNC, IEEE, 2019, pp. 1–6.
- [29] R.A. Agyapong, M. Nabil, A.-R. Nuhu, M.I. Rasul, A. Homaifar, Efficient detection of gps spoofing attacks on unmanned aerial vehicles using deep learning, in: 2021 IEEE Symposium Series on Computational Intelligence, SSCI, IEEE, 2021, pp. 01–08.
- [30] T. Talaei Khoei, S. Ismail, N. Kaabouch, Dynamic selection techniques for detecting GPS spoofing attacks on UAVs, *Sensors* 22 (2) (2022) 662.
- [31] M. Umer, I. Ashraf, Y. Park, et al., Enhanced machine learning ensemble approach for securing small unmanned aerial vehicles from GPS spoofing attacks, *IEEE Access* (2024).
- [32] J. Whelan, T. Sangarapillai, O. Minawi, A. Almelhadi, K. El-Khatib, Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles, in: Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 23–28.
- [33] J. Whelan, A. Almelhadi, K. El-Khatib, Artificial intelligence for intrusion detection systems in unmanned aerial vehicles, *Comput. Electr. Eng.* 99 (2022) 107784.
- [34] E. Ebeid, M. Skriver, J. Jin, A survey on open-source flight control platforms of unmanned aerial vehicle, in: 2017 Euromicro Conference on Digital System Design, (DSD), IEEE, 2017, pp. 396–402.
- [35] M. Bakirci, M.M. Ozer, Enhancing forensic analysis with autonomous UAV deployment for aerial investigation, in: 2024 12th International Symposium on Digital Forensics and Security, ISDFS, IEEE, 2024, pp. 1–6.
- [36] T. Gu, Z. Fang, Z. Yang, P. Hu, P. Mohapatra, Mmsense: Multi-person detection and identification via mmwave sensing, in: Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems, 2019, pp. 45–50.
- [37] M.A. Talukder, M.M. Islam, M.A. Uddin, K.F. Hasan, S. Sharmin, S.A. Alyami, M.A. Moni, Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction, *J. Big Data* 11 (1) (2024) 33.
- [38] A.F. Mirza, Z. Shu, M. Usman, M. Mansoor, Q. Ling, Quantile-transformed multi-attention residual framework (QT-MARF) for medium-term PV and wind power prediction, *Renew. Energy* 220 (2024) 119604.
- [39] E. Bisong, Introduction to scikit-learn, in: Building Machine Learning and Deep Learning Models on Google Cloud Platform, Springer, 2019, pp. 215–229.
- [40] N. Trendafilov, M. Gallo, Multivariate Data Analysis on Matrix Manifolds, Springer, 2021.
- [41] H. Rajadurai, U.D. Gandhi, An empirical model in intrusion detection systems using principal component analysis and deep learning models, *Comput. Intell.* 37 (3) (2021) 1111–1124.
- [42] H. Abdi, L.J. Williams, Principal component analysis, *Wiley Interdiscip. Rev.: Comput. Stat.* 2 (4) (2010) 433–459.
- [43] C. Labrín, F. Urdinez, Principal component analysis, in: R for Political Data Science, Chapman and Hall/CRC, 2020, pp. 375–393.
- [44] S.A. Alex, J.V.V. Nayahi, S. Kaddoura, Deep convolutional neural networks with genetic algorithm-based synthetic minority over-sampling technique for improved imbalanced data classification, *Appl. Soft Comput.* 156 (2024) 111491.
- [45] M. Injadat, A. Moubayed, A.B. Nassif, A. Shami, Multi-stage optimized machine learning framework for network intrusion detection, *IEEE Trans. Netw. Serv. Manag.* 18 (2) (2020) 1803–1816.
- [46] A. Newaz, F.S. Haq, A novel hybrid sampling framework for imbalanced learning, 2022, arXiv preprint arXiv:2208.09619.
- [47] X. Zhang, J. Ran, J. Mi, An intrusion detection system based on convolutional neural network for imbalanced network traffic, in: 2019 IEEE 7th International Conference on Computer Science and Network Technology, ICCSNT, IEEE, 2019, pp. 456–460.
- [48] G. Wei, W. Mu, Y. Song, J. Dou, An improved and random synthetic minority oversampling technique for imbalanced data, *Knowl.-Based Syst.* 248 (2022) 108839.
- [49] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
- [50] J. Krupski, W. Graniszewski, M. Iwanowski, Data transformation schemes for cnn-based network traffic analysis: A survey, *Electronics* 10 (16) (2021) 2042.
- [51] H. Naveed, S. Anwar, M. Hayat, K. Javed, A. Mian, Survey: Image mixing and deleting for data augmentation, *Eng. Appl. Artif. Intell.* 131 (2024) 107791.
- [52] V. Sze, Y.-H. Chen, T.-J. Yang, J.S. Emer, Efficient processing of deep neural networks: A tutorial and survey, *Proc. IEEE* 105 (12) (2017) 2295–2329.
- [53] Z. Wang, L. Cheng, H. Wang, W. Zhao, X. Song, Energy optimization by software prefetching for task granularity in GPU-based embedded systems, *IEEE Trans. Ind. Electron.* 67 (6) (2019) 5120–5131.
- [54] Q.A. Al-Hajja, C.D. McCurry, S. Zein-Sabatto, Intelligent self-reliant cyber-attacks detection and classification system for IoT communication using deep convolutional neural network, in: Selected Papers from the 12th International Networking Conference: INC 2020 12, Springer, 2021, pp. 100–116.
- [55] D. Petrov, T.M. Hospedales, Measuring the transferability of adversarial examples, 2019, arXiv preprint arXiv:1907.06291.
- [56] M.M. Leonardo, T.J. Carvalho, E. Rezende, R. Zucchi, F.A. Faria, Deep feature-based classifiers for fruit fly identification (Diptera: Tephritidae), in: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI, IEEE, 2018, pp. 41–47.
- [57] C.F.G.D. Santos, J.P. Papa, Avoiding overfitting: A survey on regularization methods for convolutional neural networks, *ACM Comput. Surv.* 54 (10s) (2022) 1–25.
- [58] A. Khetan, Z. Karnin, Prunenet: Channel pruning via global importance, 2020, arXiv preprint arXiv:2005.11282.
- [59] X. Liu, Y. Kou, M. Fu, Hyperspectral image shadow enhancement using three-dimensional dynamic stochastic resonance and classification based on ResNet, *Electronics* 13 (3) (2024) 500.
- [60] S. Ament, S. Daulton, D. Eriksson, M. Balandat, E. Bakshy, Unexpected improvements to expected improvement for bayesian optimization, *Adv. Neural Inf. Process. Syst.* 36 (2024).



Mohamed Selim Korium is currently pursuing doctoral degree in electrical engineering with the School of Energy Systems at LUT University (Lappeenranta, Finland. He is also a researcher of the Cyber-Physical Systems Group in LUT School of Energy Systems at the Laboratory of Control Engineering and Digital Systems at LUT University, where he has been actively working in deep reinforcement learning for AVs and mobile robots. He received the B.Sc. degree in mechatronics and robotics engineering from the Egyptian Russian University, Egypt, M.Sc. degree in Mechanical Engineering from LUT University.



Mohamed Saber received the M.Sc. degree in Machine Learning and Data Analysis and Postgraduate Diploma in computer science from Cairo University, Egypt. His thesis work focused on network traffic analysis and cyberattack detection using machine learning algorithms. In the industry, he has more than seven years of experience in machine learning, software engineering, and systems analysis. His research interests are anomaly detection using large language models and graph neural networks, and federated self-learning for IoT.



Ahmed Mahmoud Ahmed is a doctorate student in mechatronics and robotic engineering at Zagazig University in Egypt. In addition to being a teacher assistant at an Egyptian-Russian university, he is the chief of simulation and modeling at Sphinx for Engineering Works in Egypt. He is currently engaged in research in deep learning, machine learning, and digital twins. He graduated from Zagazig University with a B.Sc. in mechanical engineering. He also holds an M.Sc. in deep learning, machine learning, and digital twins from Nile University.



Arun Narayanan received the B.E. degree in Electrical Engineering from the Visvesvaraya National Institute of Technology, Nagpur, India and M.Sc. in Energy Technology from Lappeenranta University of Technology (LUT), Arun Narayanan Finland, in 2002 and 2013, respectively. He subsequently completed his D. Sc. (Tech.) from the School of Energy Systems, LUT University, in 2019. He is currently a postdoctoral researcher with LUT University, Lappeenranta, Finland, in the Cyber-Physical Systems Group. His research interests include renewable-energy-based smart microgrids, electricity distribution and markets, demand-side management, energy management systems, and information and communications technology. He focuses on applying optimization, computational concepts, and artificial intelligence techniques to renewable electrical energy problems.



Pedro H.J. Nardelli received the B.S. and M.Sc. degrees in electrical engineering from the State University of Campinas, Brazil, in 2006 and 2008, respectively. In 2013, he received his doctoral degree from University of Oulu, Finland, and State University of Campinas following a dual degree agreement. He is currently Assistant Professor (tenure track) in IoT in Energy Systems at LUT University, Finland, and holds a position of Academy of Finland Research Fellow with a project called Building the Energy Internet as a large-scale IoT-based cyber-physical system that manages the energy inventory of distribution grids as discretized packets via machine-type communications (EnergyNet). He leads the Cyber-Physical Systems Group at LUT and is Project Coordinator of the CHIST-ERA European consortium Framework for the Identification of Rare Events via Machine Learning and IoT Networks (FIREMAN). He is also an adjunct professor at University of Oulu in the topic of “communications strategies and information processing in energy systems.” His research focuses on wireless communications particularly applied in industrial automation and energy systems. He received a best paper award of IEEE PES Innovative Smart Grid Technologies Latin America 2019 in the track “Big Data and Internet of Things”. He is also IEEE Senior Member. More information.

<https://sites.google.com/view/nardelli/>