PERFORMANCE COMPARISON OF WEAK AND STRONG LEARNERS IN DETECTING GPS SPOOFING ATTACKS ON UNMANNED AERIAL VEHICLES (UAVS)

by

Aydan Gasimova

Bachelor of Science, Information technologies and system engineering, Azerbaijan State Economic University, 2017

A Thesis

Submitted to the Graduate Faculty

of the

University of North Dakota

in partial fulfillment of the requirements

for the degree of

Master of Science

Grand Forks, North Dakota

December

2022

Copyright 2022 Aydan Gasimova

Name: Aydan Gasimova

Degree: Master of Science

This document, submitted in partial fulfillment of the requirements for the degree from the University of North Dakota, has been read by the Faculty Advisory Committee under whomthe work has been done and is hereby approved.

DocuSigned by:
Naima Kaabouch
Naima Kaabouch

DocuSigned by:
Emanuel Grant

Emanuel Grant

DocuSigned by:
Wen-Chen Hu
Wen-Chen Hu

This document is being submitted by the appointed advisory committee as having met all the requirements of the School of Graduate Studies at the University of North Dakota and is hereby approved.

DocuSigned by:
Chris Nelson
Chris Nelson
Dean of the School of Graduate Studies

12/9/2022

Date

PERMISSION

Title	Performance Comparison of Weak and Strong Learners in Detecting GPS Spoofing Attacks on Unmanned Aerial Vehicles (UAVs)
Department	School of Electrical Engineering and Computer Science
Degree	Master of science

In presenting this thesis in partial fulfillment of the requirements for a graduate degree from the University of North Dakota, I agree that the library of this University can make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by the professor who supervised my thesis work or, in her absence, by the Chairperson of the department or the dean of the School of Graduate Studies. It is understood that any copying or publication or other use of this thesis or part thereof for financial gain shall not be allowed without my written permission and that of my advisor. It is also understood that due recognition shall be given to me and to my advisor in any scholarly use which may be made of any material in my thesis.

Aydan Gasimova

12/8/2022

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

ACKNOWLEDGEMENTS

First, I would like to express my gratitude to my academic advisor, Dr. Naima Kaabouch, who had a crucial role in helping me to complete my master's degree. Her continued support and guidance always provided me with invaluable insights and strengthened my abilities. This work would have never been possible without her directions and professional feedback.

I would also like to express my gratitude to the committee members, Dr. Grant and Dr. Hu for their time and valuable feedback.

Lastly, I acknowledge the support of the National Science Foundation (NSF). The work performed in two years of my studies was supported through the NSF grant #2006674.

ABSTRACT

Unmanned Aerial Vehicle systems (UAVs) are widely used in civil and military applications. These systems rely on trustworthy connections with various nodes in their network to conduct their safe operations and return-to-home. These entities consist of other aircrafts, ground control facilities, air traffic control facilities, and satellite navigation systems. Global positioning systems (GPS) play a significant role in UAV's communication with different nodes, navigation, and positioning tasks. However, due to the unencrypted nature of the GPS signals, these vehicles are prone to several cyberattacks, including GPS meaconing, GPS spoofing, and jamming. Therefore, this thesis aims at conducting a detailed comparison of two widely used machine learning techniques, namely weak and strong learners, to investigate their performance in detecting GPS spoofing attacks that target UAVs. Real data are used to generate training datasets and test the effectiveness of machine learning techniques. Various features are derived from this data. To evaluate the performance of the models, seven different evaluation metrics, including accuracy, probabilities of detection and misdetection, probability of false alarm, processing time, prediction time per sample, and memory size, are implemented. The results show that both types of machine learning algorithms provide high detection and low false alarm probabilities. In addition, despite being structurally weaker than strong learners, weak learner classifiers

also, achieve a good detection rate. However, the strong learners slightly outperform the weak learner classifiers in terms of multiple evaluation metrics, including accuracy, probabilities of misdetection and false alarm, while weak learner classifiers outperform in terms of time performance metrics.

Chapter 1

INTRODUCTION

1.1 Motivation and Problem Statement

Unmanned Aerial Vehicle networks have increased in importance due to their high use in military and civilian applications [1]. Military uses include monitoring, area mapping, inspection, reconnaissance, and special missions. Civilian uses include agricultural observation, meteorological surveillance, cargo transportation, catastrophe detection, delivery services, and photography. According to the US Federal Aviation Administration (FAA), at peak operational periods in 2022, an average of 5400 aircraft will be flying in airspace at any given time [2]. In addition, the FAA has authorized the use of drones in public airspace for over 75 public institutions for a variety of objectives, including surveillance, public safety, research, and other purposes [3]. Furthermore, it is expected that around 250,000 UAVs will be functioning in the United States by 2035, which is considered more than the world's total of 45,000 commercial airliners [4].

To deal with this volume of air traffic, significant technological improvements in the design, automation, and surveillance of UAVs have been accomplished over the past two decades; however, these advancements demand exceedingly precise navigation and surveillance techniques. Existing airplane surveillance technologies are classified into three types [4], including Procedural Air Traffic Control (ATC), Primary

Surveillance Radar (PSR), and Secondary Surveillance Radar (SSR). In the first approach, flight crews are required to update their locations regularly through radio communications. ATC is mostly used in places with limited or no radar coverage, such as the seas. The second category, the PSR is a non-cooperative security system that identifies the location of an aircraft based on its distance and azimuth from the ground control station. This system is self-contained and does not depend on the data from the corresponding aircraft. The last method, the SSR is a partially autonomous surveillance system that calculates an aircraft's location when inquired by a ground station using an aircraft transmitter reply.

However, these systems, which include SSR and PSR, are exceedingly expensive to maintain. Furthermore, these technologies are extremely slow to operate and incapable of handling future growth in air traffic.

For the safety of return-to-home operations over short to long distance missions,

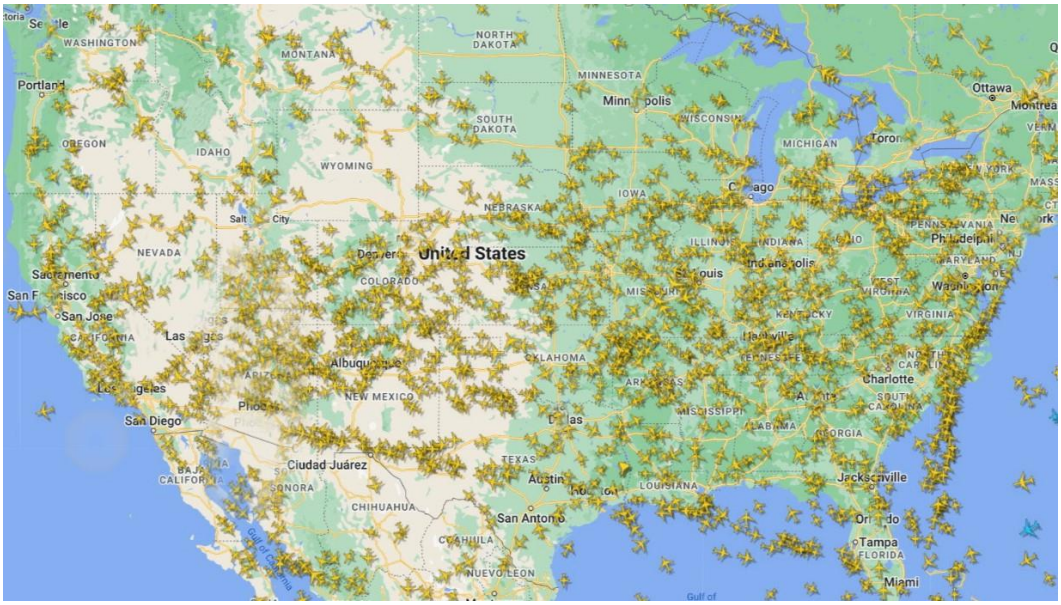


Figure 1.1. A visualization of the number of airplanes flying over the US at any given time [2]

UAVs rely on Global Positioning System (GPS) receivers, which use satellite signals to provide precise location and time services. These signals are sent at two different frequencies: L1 (1575.42MHz) for civil use and L2 (1227.60MHz) for military usage [7]. However, because civilian UAVs are not encrypted, they are vulnerable to cyber-attacks, such as meaconing, GPS jamming, and spoofing than military UAVs. The impacts of these attacks can vary from simple deceiving pilots to severe denial of service, which can significantly raise the risk of collisions and casualties.

There are multiple GPS spoofing detection methods in the existing literature. These can be classified into four categories, namely cryptography, additional UAV features, direction-of-arrival (DoA) sensing, and machine learning [9, 10]. These strategies aim to identify and validate the authenticity of the GPS receivers. However, these approaches have several drawbacks that make them ineffective for real-time applications to detect GPS spoofing attacks, low detection rate, high false alarm and misdetection rates, and dependence on external hardware devices or sensors. In addition, some studies simulated only simplistic types of GPS spoofing attacks making them less effective in detecting sophisticated types of GPS spoofing attacks. Moreover, some of the methodologies employed in those studies are infeasible for UAVs due to their limited power and low processing resources.

1.2 Thesis Goal and Objectives

To address the limitations of the existing detection approaches, the goal of this thesis is to conduct a performance comparison of strong and weak learner models in detecting GPS spoofing attacks that target UAVs. This analysis will be done in terms of multiple evaluation metrics, namely accuracy, probabilities of detection and

misdetection, probability of false alarm, processing time, prediction time per sample, and memory size.

Therefore, this thesis objectives are:

- Conduct a detailed comparison of machine learning methods to detect and identify GPS spoofing attacks,
- Extensively test the two types of machine learning models in terms of several metrics.

1.3 Contributions

Two different machine learning technique categories, weak and strong learners, are analyzed in-depth. It compares their performance in terms of seven evaluation metrics: accuracy, probability of detection, probability of false alarm, probability of misdetection, prediction time per sample, processing time, and memory size. Each method depends on a set of hyperparameters, which determines how well (or poorly) the algorithm performs. As a result, two papers have been published and presented at two IEEE conferences:

- A. Gasimova, T. T. Khoei, and N. Kaabouch, "A Comparative Analysis of the Ensemble Models for Detecting GPS Spoofing attacks on UAVs." In 12th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0310-0315. IEEE, 2022.
- T. T. Khoei, A. Gasimova, M. A. Ahajjam, K. A. Shamaileh, V. Devabhaktuni, and N. Kaabouch, “A comparative analysis of supervised and unsupervised models for detecting GPS spoofing attack on UAVs,” in 2022 IEEE International Conference on Electro Information Technology (eIT), 2022.

1.4 Thesis Organization

This thesis is organized as follows:

Chapter 2 provides a background overview of the UAV's vulnerabilities and types of attacks targeting those devices. In addition, existing detection techniques are discussed and analyzed in depth.

Chapter 3 describes the methodology used in this thesis work. This chapter discusses the attack detection process that was built to simulate three forms of GPS spoofing attacks: simple, intermediate, and advanced. In addition, this chapter explains the procedures for generating training datasets for each attack that was utilized by the machine learning algorithms. Moreover, the chapter describes the machine learning techniques employed in this thesis. Finally, hyperparameter optimization techniques are thoroughly investigated and analyzed.

Chapter 4 describes and analyses the results of GPS spoofing attacks. In addition, the results of feature selection and hyperparameter optimization techniques on each algorithm's detection performance are also described. Consequently, the results of the strong and weak learner classifiers are compared in detail in terms of seven evaluation metrics.

Chapter 5 finally concludes this thesis and discusses future works and open research directions.

Chapter 2

BACKGROUND OF GPS SPOOFING ATTACKS AND DETECTION METHODS ON UAVS

In this chapter, an overview of the attacks targeting UAVs is provided. In addition, an overview of a few detection strategies is given.

2.1 Overview of Unmanned Aerial Vehicles (UAVs)

The Global Positioning System (GPS) has become a highly prevalent source of surveillance, navigation information, and geolocation for over a billion devices over the past decades [11]. The use of GPS in the transport industry is one of the first in civil areas and is growing quickly as a result of the need for navigation and air traffic control. UAVs have common features and restrictions used in existing modeling methodologies as shown below (see Figure 2.1):

a) *Specifics of motion.* Drones are able to autonomously land and take off, maintain flight stability, and maintain the necessary altitude. However, certain elements of drone mobility may need to be considered while planning drone operations. One of these is the limitation of minimum turning radius while changing directions in flight [12], which is notably significant for fixed-wing drones. Small and micro drones are extremely vulnerable to meteorological variables such as wind, which may be represented as unpredictable travel times [13].

During landings and takeoffs, fixed-wing drone specifications for flying angles should be considered.

b) Limited weight. Drones used for delivering packages usually have payloads below 3 kg and carry a limited weight per flight [14]. Payload limitations are closely connected to the capability of the drone's energy storage system as well as the size, design, and cost of the UAV. Consequently, to fly the same route, a heavier drone requires more energy consumption than a lightweight drone.

c) *Limited flight range*. A drone's energy consumption is determined by a variety of elements, including the type of drone, flying height, flight circumstances (such as forward flight, etc.), payload, and meteorological conditions. The energy unit's restricted capacity is typically described as the maximum operation time, maximum flying range, or the maximum number of locations a

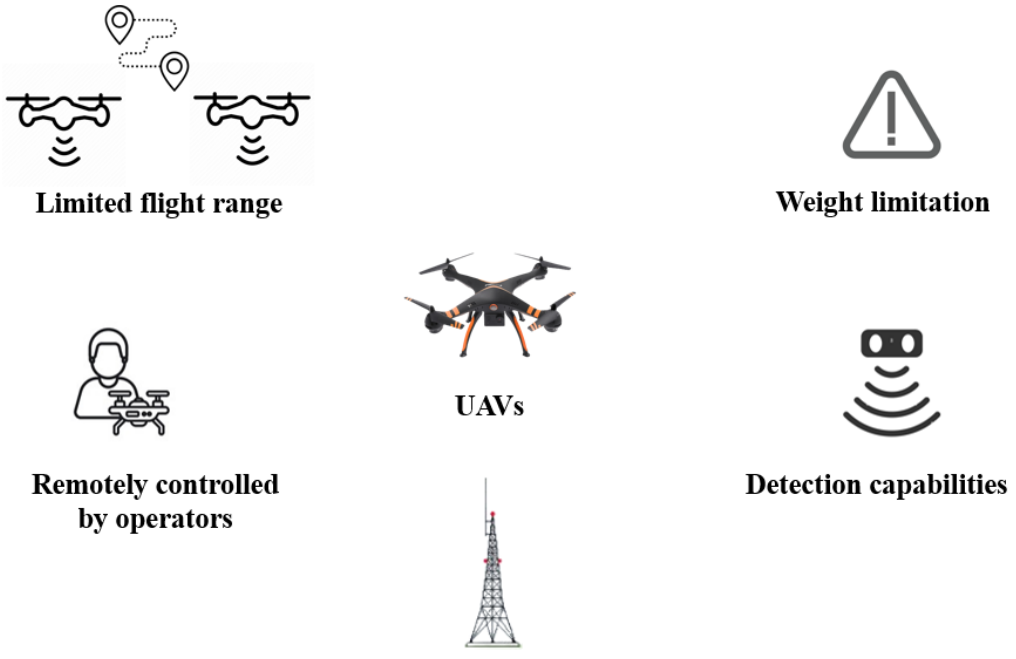


Figure 2.1. Operation of UAVs

drone may visit in a single flight [15-16].

d) *Specifics of information processing*. To receive commands and transmit the data they have received [17], drones must maintain communication connections with the ground station. Since line-of-sight communication is usually necessary, the signal deteriorates indoors or in the shadow of buildings.

e) *Remotely controlled by operators.* Operators are required for drone missions in a number of countries [18, 19]. An operator often undertakes a series of preparation activities prior to the drone's takeoff; and after landing, they may be required to manage and examine the drone.

2.2 Type of attacks targeting UAVs

As previously mentioned, the positioning and navigation tasks of UAVs highly depend on GPS. Nevertheless, due to the unencrypted nature of the GPS signals, these vehicles are prone to several cyber-attacks [20, 21]. In the current situation, these attacks can be divided into three types: meaconing or replay attacks, GPS jamming, and GPS spoofing. These attacks violate the three security requirements, namely integrity, authenticity, and availability in terms of security requirements. In the following, a description of each class is described in the context of UAV.

Integrity: It relates to protecting data from unauthorized modifications. These methods ensure that data is accurate and comprehensive. The autopilot mode of UAVs is entirely dependent on the GPS positions of UAVs, ground stations, and targets. As a result, GPS spoofing attacks target the integrity of GPS signals.

Authenticity: The identification of the nodes that are broadcast should always

be known to the recipient of the communication. Meaconing attacks, on the other hand, violate the authentication criterion since the transmitter of the repeated signals is not a GPS satellite.

Availability: This term refers to the fact that GPS signals must available to end users anywhere and at any time. For instance, an attacker may jam GPS receivers of UAVs or GC and launch a denial-of-service attack. As a result, jamming attacks target the availability of GPS signals.

2.2.1 Meaconing

In meaconing attacks, an attacker records and retransmits later the GPS to a target without affecting the content of the signals. Figure 2.2 depicts the delayed transmission for meaconing attacks [21]. By masking the actual received signal, the meaconing signal attempts to deceive the receiver into selecting an inaccurate navigation solution [22]. If the meaconing attack succeeds, the target receiver will report the position included in the re-transmitted data instead of the genuine position.

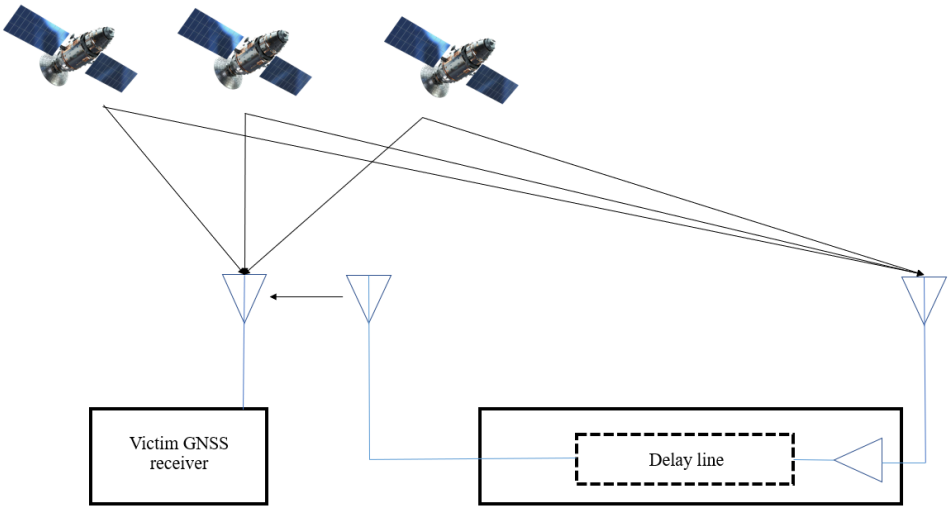


Figure 2.2. Overview of meaconing attack

2.2.2 Jamming Attacks

As mentioned in the preceding paragraph, GPS receivers are vulnerable to spoofing or jamming attacks where attackers can insert counterfeit GPS signals into the network [27]. GPS jamming attacks are frequently carried out by intentionally sending messages in order to prevent genuine members of a network from transmitting or receiving data, which can result in denial of service. Jamming is the deliberate broadcast of powerful radio frequency signals. Some research demonstrated that jamming attacks might interfere with GPS and Galileo satellite signals at the same time [23 - 25]. This attack can decrease the quality of datalinks and prevent GPS devices Therefore, these attacks are known as a significant problem in wireless networks.

2.2.3 GPS Spoofing attacks

GPS spoofing attacks attempt to transmit a GPS signal to the UAV's GPS receiver causing the UAV to place itself in the incorrect position. The GPS spoofing

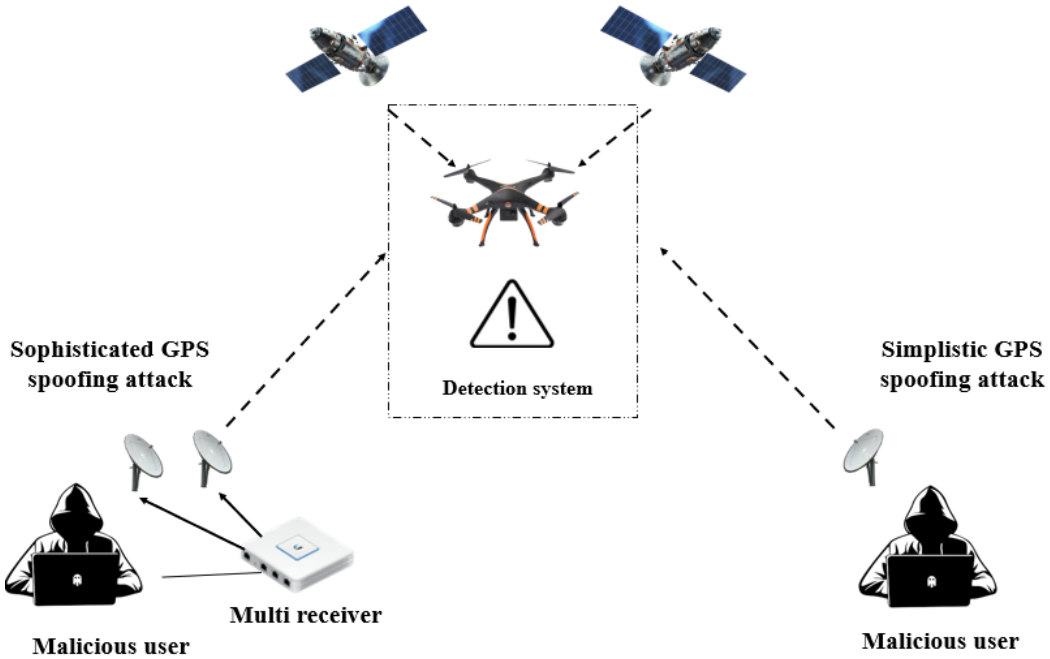


Figure 2.3. Overview of GPS spoofing attacks

attack procedure is shown in Figure 2.3. Consequently, the receiver is guaranteed to lock onto the fake signal rather than the actual GPS signal [28].

According to multiple studies, GPS spoofing attacks are classified into three types, namely simplistic, intermediate, and sophisticated, as shown in Figure 2.4. In the first category, a GPS signal simulator interfered with a radio frequency to imitate authentic GPS signals. These types of attacks are the most used techniques to spoof GPS receivers since they only use a commercial GPS signal simulator. In intermediate spoofing attacks, determines the target receiver's antenna position and velocity to create counterfeit signals. The last category, sophisticated spoofing assaults, is the most successful sort of GPS spoofing attack. In these attacks,

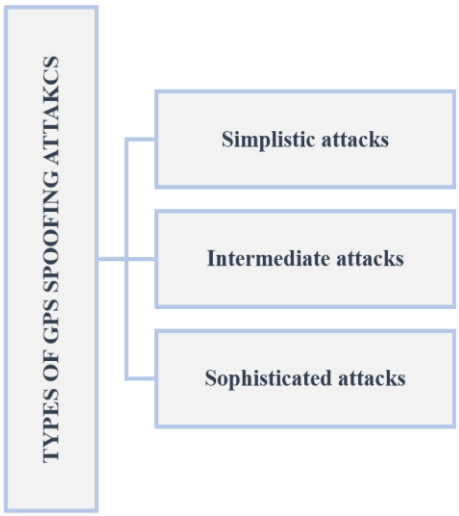


Figure 2.4. Types of GPS spoofing attacks

2.3 Existing GPS spoofing detection methods targeting UAVs

Over the last decade, several studies related to the security of UAVs have been conducted to investigate and analyze cyber-attacks . For instance, the authors of [29] investigated the testbeds and analyzed the impact of GPS attacks on UAVs. In [30], in

addition to analyzing the security of UAVs, the authors evaluated the types of attacks through a series of tests in a simulation environment. Moreover, they analyzed the behavior of the different GPS spoofing attacks on quadcopters in terms of security and safety issues. In recent works, the authors of [31 - 34] investigated the security issues of UAVs and analyzed some theoretical and practical solutions to detect and mitigate spoofing attacks.

To address these vulnerabilities, several studies have been conducted to detect GPS spoofing attacks. Existing methods can be classified into four categories, namely cryptography, additional UAV features, Direction-of-Arrival (DoA) sensing, and machine learning, as shown in Figure 2.5.

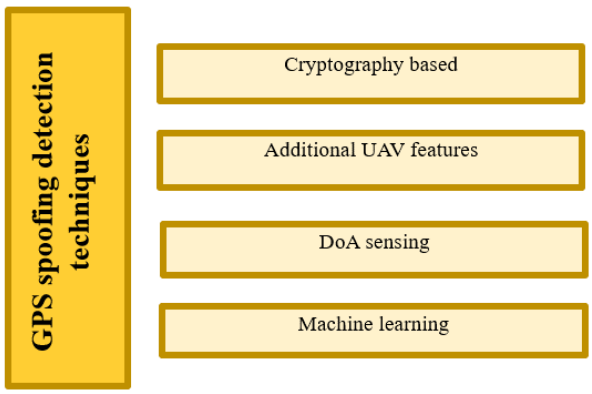


Figure 2.5. Types of GPS spoofing detection

2.3.1 Cryptography-based techniques

Cryptographic methods enable receivers to differentiate authentic GPS signals from counterfeit ones with a high probability. Nevertheless, they are impractical for civil applications due to the requirement [45 - 47]. In addition, these methods are not resistant to replay attacks, in which the attacker resends the legitimate signal with no

modifications after a long period, generating both time and position errors. Therefore, cryptography methods are not practical solutions to defend against attacks on GPS receivers.

2.3.2 Additional UAV features

Additional UAV features are based on the external UAV characteristics, including control system, acceleration, and inertial measurement unit (IMU). For instance, in [35], the authors developed a GPS spoofing technique to detect malicious attacks based on IMU. The results show a good detection rate. In [36], the authors proposed a GPS spoofing detection technique based on the monocular camera and IMU sensor of UAVs. Their results show the proposed hardware can detect spoofing attacks with good speed and detection rate. In [37], the authors proposed a low-complexity authenticity verification method developing a novel model for signal quality evaluation. Their results demonstrate the efficiency of the proposed method. Moreover, the authors of [38] proposed a vision-based UAV spoofing detection method that employs visual odometry. The obtained results of the proposed method prove the efficiency of spoofing detection on UAVs. However, these techniques have a few limitations, such as external networks and low accuracy, which are not practical for UAVs [49].

2.3.3 Direction-of-Arrival (DoA) sensing

DoA sensing takes advantage of the fact that the spoofer transmits malicious signals from a single antenna, and they come from the same source. Authentic GPS signals, on the other hand, originate from several satellites and hence from multiple angles [50, 51].

2.3.4 Machine learning techniques

Machine learning methods do not necessitate the acquisition of additional hardware, which makes them suitable for civilian UAVs. Therefore, some authors have proposed various GPS spoofing detection methods based on conventional machine learning (ML) models [39 - 43]. For instance, the authors of [39] proposed two dynamic selection techniques, including Metric and Weighted Metric Optimized Dynamic selectors to detect GPS Spoofing attacks targeting UAVs. They evaluated the performance of the proposed approach in terms of detection, misdetection, false alarm probability, processing time, and accuracy. Their results show acceptable results. The authors of [40] proposed a GPS spoofing detection method based on an adaptive K-nearest Neighbors classifier and synchronization-free GPS-Probe method. The simulation results indicate that the proposed methodology can successfully identify malicious GPS spoofing attacks on UAVs with high detection accuracy.

The authors of [41] proposed a two-step genetic algorithm-based extreme Gradient Boosting method to detect GPS-spoofing attacks. The results show that the proposed method can achieve high detection results to detect GPS spoofing attacks on UAVs. In [42], the authors developed a one-class support vector machine classifier technique to detect anomalies targeting wireless network systems. Lastly, the authors of [43] proposed another ML-based model using SVM to detect spoofed GPS signals on UAVs. The result of the proposed technique shows a high detection rate to detect malicious GPS signals.

Chapter 3

METHODOLOGY

In this thesis, supervised machine learning algorithms are employed. These algorithms must first be trained on accurate training data. The training dataset in this thesis consists of a significant number of legitimate and malicious GPS samples. The following sections describe the training dataset acquisition, data preprocessing techniques, and feature selection methods. Finally, the hyperparameter techniques and classification models are discussed.

3.1 Attack Detection Procedure using Machine Learning algorithms

Figure 3.1 provides the ML model workflow for GPS Spoofing detection. In the first phase, real GPS signals are collected using software-defined radio units and spoofed signals are simulated. Three types of attacks, simplistic, intermediate, and sophisticated were generated through simulations.

In the second phase, data preprocessing techniques are employed, such as data imputation and data transformation. In this study, value imputation and data normalization are used to obtain the best model results.

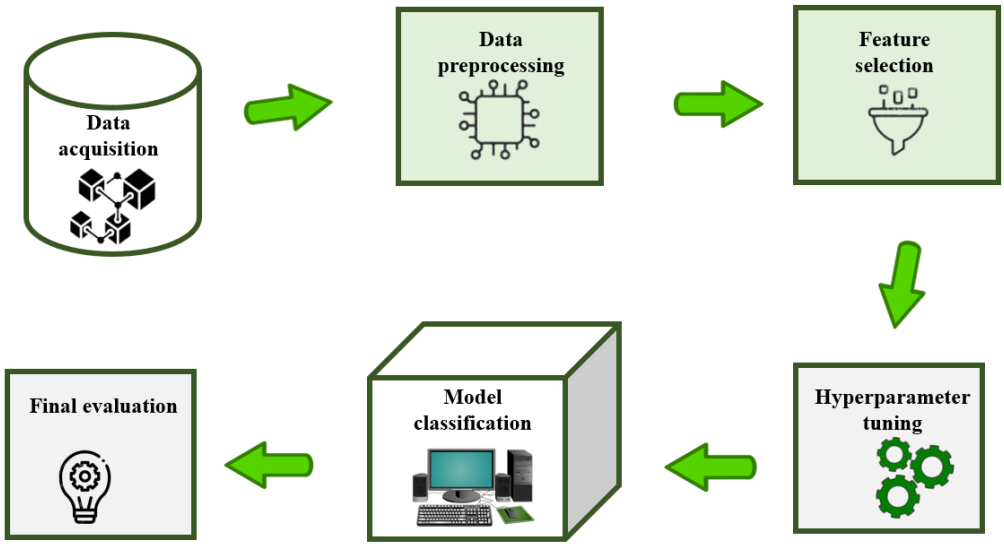


Figure 3.1. Supervised machine learning workflow

In the third phase, feature selection methods are employed to discard redundant and low-importance features in the corresponding dataset to increase the accuracy of the ML models. Two ensemble feature selection methods are used, namely Mutual information and Spearman correlation. These methods are capable of identifying correlated and irrelevant features.

In the next stage, hyperparameter tuning methods are applied for all considered ML models, to obtain optimal results. Next, eight machine learning models are implemented. These models are bagging, boosting, stacking, multinomial NB, complementary NB, Bernoulli NB, Gaussian NB, and Gaussian Process NB. The algorithms were implemented using a Python library– Scikit-learn. To find the best hyperparameters, two different techniques, grid search, and genetic algorithm are employed. In the final phase, the model learning is conducted, and the performance is performed by applying specific metrics.

3.2 Training Dataset

Supervised machine learning algorithms must be trained on reliable training datasets. In this thesis, we used a dataset developed in [56]. This dataset was generated by collecting real GPS signals using software-defined radio units. The hardware employed for the implementation was an Ubuntu 16.04 LTS with 8G RAM. In addition, MATLAB was used to simulate GPS attacks, simple, intermediate, and sophisticated.

Table 3.1 gives the thirteen characteristics retrieved from the data, with their descriptions. These features are extracted through three extraction phases, beginning with the pre-correlation phase and ending with the delay-locked and post-correlation stages. The feature extraction procedure begins with estimating the carrier-to-noise ratio (C/N0) of the received signal [56]. The remaining characteristics are extracted during the observables block.

The corresponding dataset is balanced and consists of 10,056 samples, including 5028 attack samples equally divided between the three types of GPS spoofing attack signals. Data corresponding to GPS spoofing attacks are encoded as 1, and the remaining are encoded as 0.

Table 3.1. List of extracted features from the corresponding dataset [56]

Extracted features	**Abbreviations**	**Descriptions**	**Receiver stage**
Carrier to Noise Ratio	C/N0	Indicator of the signal that carries the GPS information	Pre-correlation
Magnitude of the Early Correlator	EC	Magnitudes of the Early correlator are used for timing recovery	During correlation

Magnitude of the Late Correlator	LC	Magnitudes of the Late correlator are used for timing recovery	During correlation
Magnitude of the Prompt Correlator	PC	Estimation of phase and frequency differences	During correlation
Prompt in-phase correlator	PIP	In-phase signal of the prompt correlator	During correlation
Prompt Quadrature component	PQP	Quadrature signal of the prompt correlator	During correlation
Carrier Doppler in Tracking loop	TCD	Carrier Loop Doppler Measurements	During correlation
Carrier Doppler	DO	Change in frequency for a GPS receiver	Post-correlation
Pseudo-range	PD	Time difference between transmission and reception time	Post-correlation
Receiver Time	RX	Time of reception after the start of the time of the week	Post-correlation
Time of the week	TOW	Time of the transmission of the navigation message	Post-correlation
Carrier Phase Cycles	CP	Frequency difference between the received carrier and a receiver-generated carrier phase	Post-correlation
Satellite vehicle number	PRN	Identification of different satellites orbiting the earth	Post-correlation

3.3 Data Preprocessing

Data preprocessing consists of numerous stages, including class rebalancing

and size reduction, feature elimination, missing value imputation, and data normalization, as shown in Figure 3.2. The corresponding dataset did not contain any missing sample; therefore, we did not employ any data imputation method.

The raw data can be normalized using a variety of techniques, including mean and standard deviation-based procedures, decimal scaling normalization, and median absolute deviation normalization [57]. In this thesis, we employed the Quantile transform scalar and Min-max scalar to rescale all samples between 0 and 1 in the corresponding dataset. The Quantile scalar technique is applied to each feature independently to transform the features to follow a uniform or a normal distribution. The min-max normalization converts the data into a comparable scale, which improves classifier performance.

$$X = \frac{x - \mathrm{Min}(x)}{\mathrm{Max}(x) - \mathrm{Min}(x)} \tag{3.1}$$

Where x is the initial value, Min(x) and Max(x) are the minimum and maximum values of the feature vector.

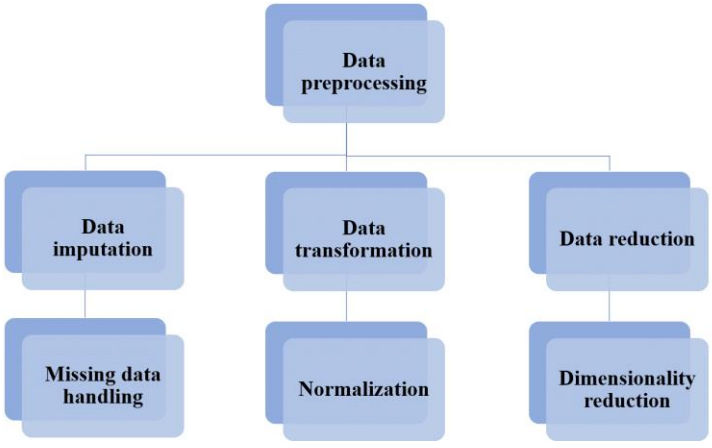


Figure 3.2. Data preprocessing stages

3.4 Feature selection techniques

It is necessary to choose a set of features that can accurately differentiate between genuine and counterfeit GPS signals. Correlated features in the dataset can have a negative impact on classification model performance. In this thesis, we used a heterogeneous method for selecting features, Spearman's Correlation, and Mutual information.

The Spearman's Correlation. When the result is close to 1 or -1, the characteristics have a strong correlation, either positive or negative [58]. A positive correlation coefficient indicates a positive linear correlation, while a negative correlation coefficient indicates a negative linear correlation. The correlation coefficient is given by:

$$\mu = 1 - \frac{\sum_{i=1}^{n}(d_i)^2}{n(n^2-1)} \tag{3.2}$$

Where d_i stands for the difference between the two ranks per observation, i is the observation's index, and n is the total number of occurrences. In this work, a feature is considered correlated if it accomplishes a coefficient over 0.9.

Mutual information. This technique, also known as entropy, is applied to every feature; characteristics chosen as significant features have high entropy values, whereas features chosen as low relevance have low entropy values. In this study, each feature with an entropy of less than 0.1 is removed from the dataset. The mutual information technique is given by [59]:

$$I(X, Y) = H(Y) - H(Y \mid X) \tag{3.3}$$

Where X and Y are random variables, H (Y) is the entropy that is used to quantify

a random variable's level of uncertainty, while H (Y | X) is the conditional entropy that expresses how much uncertainty is still present in Y.

Non-stationary distribution. Non-Stationary Data Modification is required to maintain a static connection between machine learning models and non-stationary data. In this work, we investigate the features with non-stationary distributions and use interpolation to convert the data into stationary data [56]. This process determines the sequential differences between samples, as shown below:

$$R = \frac{x_{i+1} - x_i}{n_{i+1} - n_i} \tag{3.4}$$

Where R is the change rate and $n_{i+1} - n_i$ is the distance difference between two samples, which in our case is equal to 1.

3.5 Weak learner Machine learning techniques

Weak learner classifiers are models that can predict an intended result; however, they are not flexible enough to estimate accurately for all predefined classes and all predicted instances. They concentrate on successfully forecasting a set of target cases or a single target. Weak learnability excludes the requirement that the learner reaches high accuracy; rather, it must simply generate a hypothesis that performs slightly better than random estimation.

The reliability of the ML models depends on various parameters, such as problem scope, data distribution, outliers, data quantity, and feature dimensionality. However, one of the major important parameters is the bias and variance of the relationship. Generally, they have an inverse relationship with each other, such as high bias-low variance or low bias-high variance [60]. High-bias models oversimplify the

models and mostly focus on the test data. It always leads to high errors in training and test data. To get optimal results, during the estimation of the models, the bias-variance tradeoff should be balanced. However, weak learner classifiers usually obtain high bias during classification. Figure 3.3 shows the sample of bias-variance connection, where the error (E_{out}) is the complexity of the model.

In this thesis, Naive Bayes (NB) weak learner category is used for detecting GPS spoofing threats. The key advantage of the NB model is its efficiency since training and classification may be completed with a single cycle through the data. The NB model is based on the Bayesian theorem that is based on the conditional independence assumption of characteristics. The Bayes theorem can be described as follows if B is an event and P(B) > 0 [17]:

$$P(A \mid B) = P(A)\frac{P(B|A)}{P(B)} \tag{3.5}$$

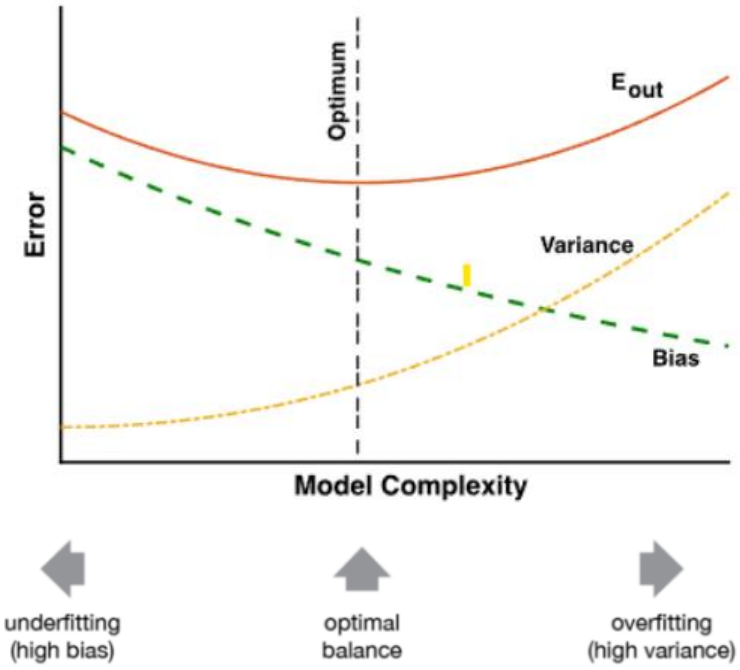


Figure 3.3. Illustration of the bias-variance connection [60]

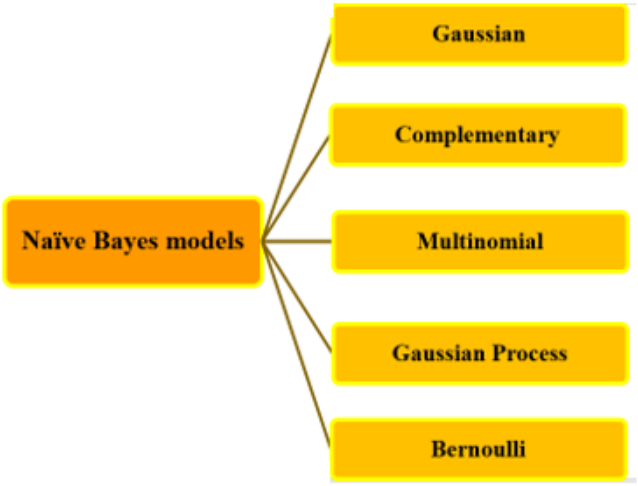


Figure 3.4. Implemented weak learner category models

Where A and B are different events, P(A|B) denotes the probability that one event will occur while another has already occurred, P(B|A) denotes the probability that event B will occur, and P(A) and P(B) denote the likelihood of the two independent events A and B [62]. Consequently, in this thesis, four Bayesian-based models are investigated. These models are multinomial, complementary, Gaussian, Bernoulli, and Gaussian processes, as shown in Figure 3.4. These models and their implementations are discussed in the next sections.

3.5.1 Multinomial Naïve Bayes

This technique is based on the NB tree, which estimates that each feature has a multinomial distribution. We assume a number of classes, $c \in \{1, 2, \ldots, m\}$, each with a fixed set of multinomial distribution parameters. The vector for a class C is expressed below:

$$\theta = \{\theta_{c1}, \theta_{c2} \ldots \theta_{cn}\} \tag{3.6}$$

Where n is the size of the dataset, $\sum_i \theta_{ci} = 1$, and θ_{ci} is the probability that instance i occurs in that class. Consequently, the maximum likelihood estimate is equated below:

$$\hat{\theta}_{ci} = \frac{N_{ci} + \alpha_i}{N_c + \alpha} \tag{3.7}$$

Where N_{ci} is the number of times sample i appears in the dataset, N_c is the total number of the samples in class c, α_i is a smoothing index, and α is the sum of α_i.

After equating the MLNB estimate in (3.7) the classification rule can be expressed as follows:

$$L_{MNB}(d) = \text{argmax}_c \left[\log \hat{p}(\theta_c) \quad + \quad \sum_i f_i \ log \ \frac{N_{ci+} \alpha_i}{N_c + a} \right] \tag{3.8}$$

Where $\hat{p}(\theta_c)$ is the prior class estimate, and f_i is the frequency of each sample that occurred in class c. However, the obtained results can be overpowered by the combination of each parameter; therefore, the weights for this classifier are equated using log parameter estimates:

$$\hat{w}_{ci} = log \hat{\theta}_{ci} \tag{3.9}$$

3.5.2 Complement Naïve Bayes

This model is based on the concept of improving the traditional MLNB classifier [63]. For the MLNB algorithm, the training data from a specific class, c, is utilized to estimate the weights. However, the CNB model estimates using data from all classes c. Therefore, the CNB model estimates more effectively since, in each iteration, it employs a more consistent amount of training data per class and reduces bias in the weight estimations. The CNB estimate is:

$$\hat{\theta}\check{c}\imath = \frac{N_{\check{c}i+}\, a_i}{N_{\check{c}} + a} \tag{3.10}$$

Where N$\check{c}$ is the total number f_i sample occurrences in classes, N$\check{c}i$ is the number of times sample i occurred in classes, and α_i and α are smoothing parameters. The classification rule is [64]:

$$l_{CNB}(d) = \text{argmax}_c \left[\log p(\vec{\theta}_c) \quad - \quad \sum_i f_i \ log \ \frac{N_{\check{c}i+}\, a_i}{N_{\check{c}} + a} \right] \tag{3.11}$$

Where $p(\vec{\theta}_c)$ is the prior class estimate and f_i is the frequency of each sample that occurred in class c. In this model, in this classification rule, the negative sign refers to the fact that the samples that poorly match the complement parameter estimates can be assigned to class c. Finally, the weight of the samples is calculated with the same equation as MLNB equated (3.9).

3.5.3 Gaussian Naïve Bayes

GNB is one of the commonly used types of Naïve Bayes model that support continuous data and performs on a Gaussian normal distribution. These classifiers estimate the conditional probabilities that belong to a specific class which assumes that the predictor variables do not consider the covariance among all variables [65]. According to Bayes theorem, the posterior probability is calculated as:

$$P(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2} \right) \tag{3.12}$$

Where y is the class label, μ (the sample mean) and σ (the standard deviation) are to be estimated from the training data. Given the predictor values, the algorithm calculates a distinct distribution for each predictor X_1,....X_p, and observations are

assigned to the class with the highest posterior probability [66].

3.5.4 Bernoulli Naïve Bayes

Data is a binary vector over the space of samples in the multivariate Bernoulli event model [68]. The probability of the given data is [67]:

$$P(x_i \mid y) = x_i P(x_i \mid y) + (1 - P(x_i \mid y))(1 - x_i) \tag{3.13}$$

Where $P(x_I \mid y)$ is the probability for each of these sample events, and y is any given class. It is assumed the likelihood of each sample existing in a dataset is independent of the presence of other words. Given a dataset, it can be seen as a collection of numerous independent Bernoulli experiments, representing the probability for each of these sample events.

3.5.5 Gaussian Process Naïve Bayes

The last member of the NB category, the Gaussian Process classifier, is a kernel-based NB classifier which can handle high-dimensional dataset issues. The core idea behind Gaussian process prediction is the Gaussian Process assumption. It is placed over the function f(x) and the latent function. Generally, the inference is separated into two steps. First, it computes the probability of the variable associated with a test case:

$$p(f_* \mid X, y, x_*) = \int p(f_* \mid X, x_*, f)\, p(f \mid X, y)\, df \tag{3.14}$$

Where $p(f_* \mid X, y)$ is the posterior over the latent variables, $p(f_* \mid X, x_*, f)$ is Gaussian. Subsequently using this distribution for the latent function (f_*) to produce a probabilistic prediction is expressed below:

$$\overline{\pi_*} \triangleq p(y_* = +1 \mid X, y, x_*) = \int \sigma(f_*)\, p(f_* \mid X, y, x_*)\, df_* \tag{3.15}$$

Where df_* is the given class.

3.6 Strong learner machine learning techniques

Traditional machine learning algorithms may not always provide a successful performance, specifically when the data is composite or imbalanced [69]. To improve the current performance, strong learners are implemented to boost the results of weak learners. A model is assumed highly learnable if there is a polynomial-time technique that produces a low error with high results.

In this thesis, ensemble models are implemented. In ensemble learning, several base models are integrated as weak learners to address the underlying complexity of the data. By training several models and integrating their predictions, these strategies increase the performance of the ensemble model. In general, these fundamental models cannot perform independently due to significant bias or excessive variation. The power of ensemble learning is that it may reduce the bias-variance balance in order to build strong learners that perform better.

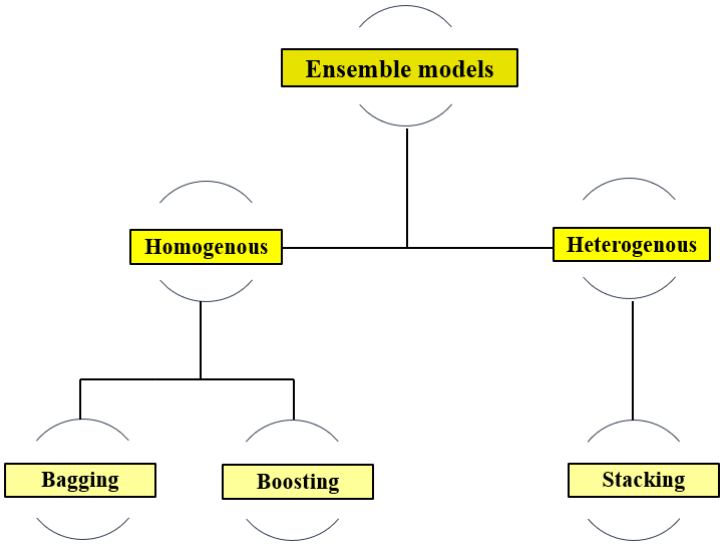


Figure 3.5. Ensemble model classification

An ensemble model has two distinct benefits. First, the relative predictive effectiveness of an algorithm is frequently situation-based, making it impossible to identify a single method. As a result, rather than selecting a single standalone model, it creates several models employing numerous standalone models and combines the prediction results to form an ensemble prediction model. Second, when provided with new conditions, an ensemble model frequently produces more accurate findings.

In general, these techniques can be classified into three categories, namely, bagging, stacking, and boosting models. These models are also classified into two categories, namely homogenous and heterogenous (refer to Figure 3.5). In this thesis, we investigate and analyze all types of ensemble models.

1) Homogeneous ensemble models

Homogeneous ensemble models are developed when all the standalone models in the ensemble have a single-type base learning algorithm. The main difficulty in these types of classifiers is generating diversity using the same algorithm. These strategies, which have mostly been applied to homogeneous ensemble models, can also be employed to develop a wider variety of heterogeneous ensembles [70]. The bagging and boosting models are considered homogenous models.

2) Heterogeneous ensemble models

In heterogeneous ensemble models, all the independent models are created using distinct algorithms. The first option for creating numerous models is to use the same machine learning model many times using the same train data. The second option for creating several models is to create separate machine learning models. However, the key challenge for these sorts of learners is determining the best strategy to integrate

predictions from multiple models in the ensemble. The stacking approach is classified as a heterogeneous ensemble method. The staking procedure will be covered in the next section.

3.6.1 Bagging classifier

This model has several ML estimators that use decision trees and individual learners to make a prediction (Figure 3.6 shows the working process of this model). One advantage of such a method is reducing the base algorithm's choice and increasing the model's accuracy. To predict, it is assumed (y, x) case in £ be independently derived from the probability distribution [71]:

$$Q(j \mid x) = P(f(x, L) = j). \tag{3.16}$$

Where Q(j | x) refers to the independent samples of the set L, f prediction class j with relative frequency Q(j | x). The overall probability of predictor classes generated state at x is [72, 73]:

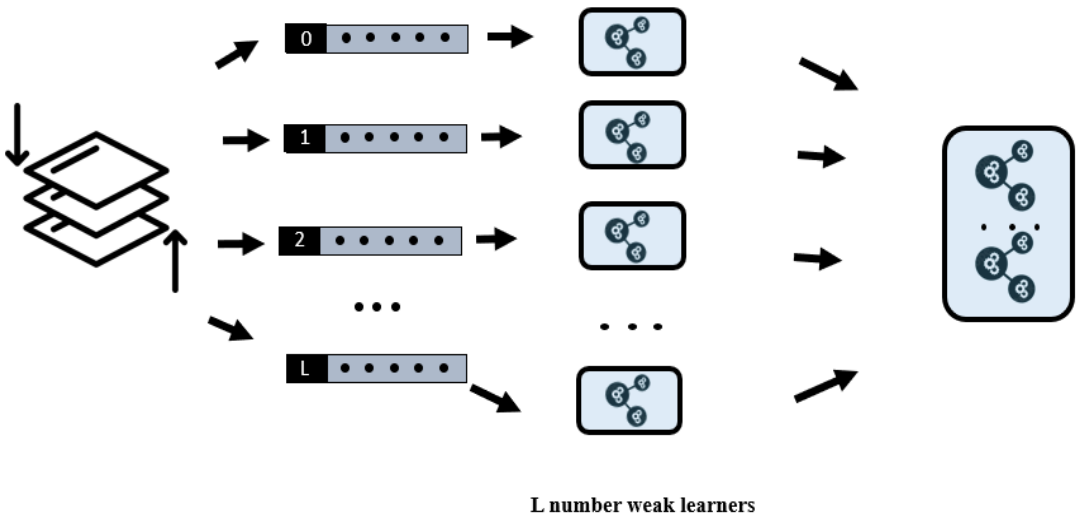


Figure 3.6. Bagging classifier work process

$$R= [\sum_j Q(j \mid x)(P(j \mid x)] \quad (3.17)$$

Where $(P(j \mid x)$ is the probability distribution. Therefore, the classification is estimated as follows:

$$\sum_j I(\text{argmax}_i Q(i \mid x)=j)P(j \mid x) \quad (3.18)$$

Where $\text{argmax}_i Q(i \mid x)$ is the aggregate predictor and I is the indicator function.

3.6.2 Boosting classifier

The boosting model is trained using weights that are adjusted based on how well each cycle's previous iteration performed. The boosting model utilizes algorithms called decision trees to enhance classification results, which combines many models of varying performance levels.

The system in the function estimation problem consists of a random output variable y and a set of random input variables y and a set of random input variables x = $\{x_1 \ldots x_n\}$. Overall, the boosting algorithm is expressed as below:

$$F_m(x)= F_{m-1}(x) + \rho_m h(x; a_m) \quad (3.19)$$

Where $h(x; a_m)$ is a member of the parameterized class of functions, $F_{m-1}(x)$ is the current estimate, and ρ_m is a line search performed.

3.6.3 Stacking classifier

The stacking technique combines several separate weak classification algorithms by combining their meta-model outputs as inputs to a final estimate to improve accuracy and other evaluation metrics [73]. This technique is developed in the following steps (also refer to Figure 3.7):

1. Base models are analyzed and predicted on each training fold (OOF).
2. The OOF predictions are sent to the meta-learner.
3. The meta-learner is trained on these OOF predictions and can be run on the test set to make final predictions.

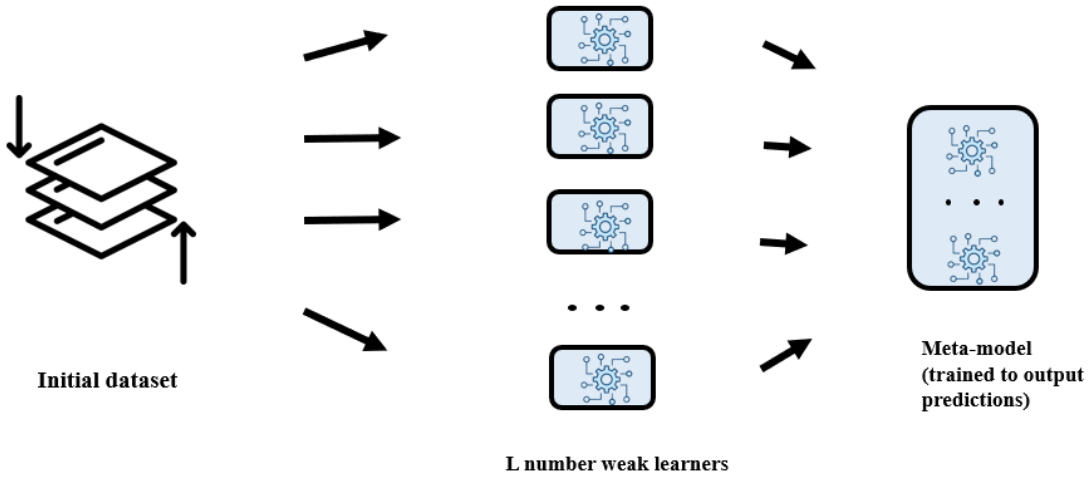


Figure 3.7. Stacking classifier work process

3.7 Hyperparameter tuning techniques

Hyperparameter tuning methods improve the efficiency of machine learning techniques. There are two types of approaches: manual search methods and automatic search algorithms. Manual search approaches are employed in order to find important parameters that have a significant impact on the results. Nonetheless, these approaches can be time consuming and cannot be used on high-dimensional data. Therefore, automatic search models address the limitation of manual learning methods [74, 75]. Several approaches are used in automatic search models, including grid search, Bayesian search, and genetic algorithm. In this thesis, two optimization techniques were used for the weak and strong classifiers, namely grid search and

genetic algorithm. In the next section, we will investigate those techniques in depth.

3.7.1 Grid search tuning method for strong learners

Although the ensemble models can achieve high results with a default value, their performance can be improved significantly using parameter optimization techniques. For this purpose, the grid search optimization technique is applied to the strong learner algorithms. Initially, this method determines the hyperparameter ranges by collecting preliminary data. Then, using the specified key points, it generates a value list for parameters, trains the data for all values in the defined range, and returns the best value [77]. Consequently, the hyperparameters are found by their minimum and maximum value and the number of their steps. [78].

3.7.2 Genetic algorithm tuning method for weak learners

A genetic algorithm is adaptive with a global search algorithm, which is based on the process of evolution. This algorithm has several advantages, including probabilistic in nature, robustness, and global optimization performance [79]. In addition, it implements biological functions for engineering problems to create efficient, high-quality, and optimized solutions.

A genetic algorithm, which is based on the process of evolution, is flexible with a global search algorithm. This method is implemented using three types of operators: selection, crossover, and mutation, as depicted in Figure 3.8 [79]. In the first operator, the major part is regarded as a chromosome, which is equal to the individuals, in which each coding unit is referred to as a gene. An evaluation function determines the fitness value of each chromosome. Figure 3.8a depicts the probability-based operators used

during the selection step. The higher the threshold, the more likely crossover, and the fewer iterations. The cross-genes for individuals that can be transferred are specified as the problem constraints. The mutation operator selects the transformed gene location and then alters the gene.

In previous methods, grid search experiences the curse of dimensionality, which causes the efficiency to decline as the number of hyperparameters increases [26]. For this purpose, the genetic algorithm is used for tuning the hyperparameters to obtain the ultimate results in weak learners.

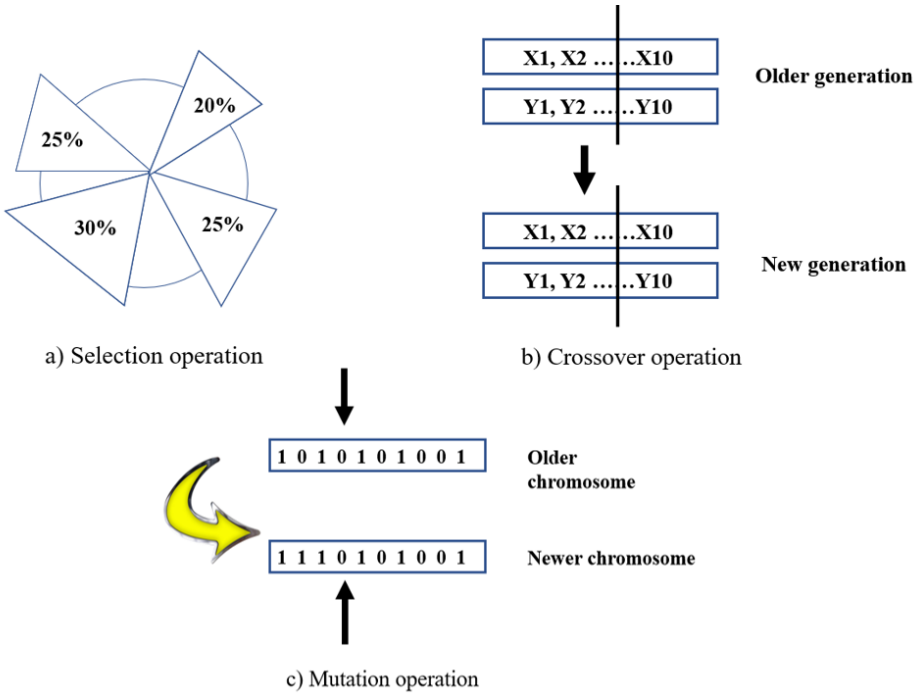


Figure 3.8. Genetic algorithm work process

Chapter 4

RESULTS AND DISCUSSIONS

This chapter provides and analyses the results of the algorithms discussed in the previous chapter to detect three types of GPS spoofing attacks targeting GPS receivers.

4.1 Performance Analysis Metrics

To compare the performance of different methods for both classifier sets, four main evaluation metrics are used. These consist of accuracy (*A*CC), probability of misdetection (*P*M), probability of detection (*P*D), and probability of false alarm (*P*FA). These metrics are calculated as follows:

$$\mathrm{ACC} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} * 100 \tag{4.1}$$

$$\mathrm{PD} = \frac{T_p}{T_p + F_N} * 100 \tag{4.2}$$

$$\mathrm{PFA} = \frac{F_p}{T_P + F_N} * 100 \tag{4.3}$$

$$\mathrm{PMD} = \frac{F_N}{T_N + F_P} * 100 \tag{4.4}$$

Where ACC is the accuracy that consists of the probability that both authentic and attacked signals are detected correctly, PD is the probability of correctly detected legitimate signals, PMD is the probability of incorrectly detected malicious signals as legitimate, *P*FA is the probability of incorrectly detected legitimate signals.

In addition to the main evaluation metrics, three additional metrics are used that

are related to size, weight, and power (SWaP) constraints on UAVs. These consist of memory size, processing time, and prediction time per sample. The description of each metric is given as follows:

- Memory time: Monitors the consumption of the memory for each model separately.
- Processing time: Refers to the prerequisite time to train and test the models and it is highly dependent on the implemented ML classifier.
- Prediction time: Refers to each instance that predicts a GPS spoofing attack during the testing phase.

4.2 Analysis of optimization parameters

In this thesis, we implemented two hyperparameter optimization methods to boost the performance of the strong and weak learner classifiers. For the NB models, we applied the genetic algorithm, while for the ensemble models, the grid search tuning technique is implemented. Those techniques, along with their best parameters, are illustrated in detail throughout the following sections.

A. *Optimization parameters for weak learners*

Since the weak learner classifiers were employed, we used a genetic algorithm

Table 4.1. List of the best parameters for weak learners

Strong learner models	**Best parameters**
MLNB	alpha=28
CNB	alpha=86
GNB	var_smoothing= 25.0
BNB	binarize=39.0, alpha= 62.0
GPNB	n_restarts_optimizer=23, random_state=65

technique to optimize the parameters. These hyperparameters are described in Table 4.1 for each of the five NB models.

B. Optimization parameters for strong learners

After implementing the models and metrics, we applied the grid search as a hyperparameter tuning technique to obtain the best results for each strong learner model. These hyperparameters are given in Table 4.2 for each of the three ensemble models.

Table 4.2. List of the best parameters for strong learners

Weak learner models	**Best parameters**
Stacking	n estimators = 42.
Bagging	final estimator verbose= 1.
Boosting	max depth= 10, min impurity decrease = 10.

4.3 Feature selection results

Figure 4.1 gives the results of Spearman's correlation coefficient for each pair of features. As one can observe, several features are highly correlated. We selected the threshold of 0.9 to identify highly correlated features. As a result of this method, EC, LC, DO, and TOW are considered highly correlated with PC, TCD, and RX, respectively.

Figure 4.2 provides the most important features according to the mutual information algorithm. As shown in this figure, PRN is the most important feature, with a score of 0.7, while PQP is the least important feature, with a score of 0.0001, compared. to the other features. However, as shown in Figure 1b, PC, TCD, and RX

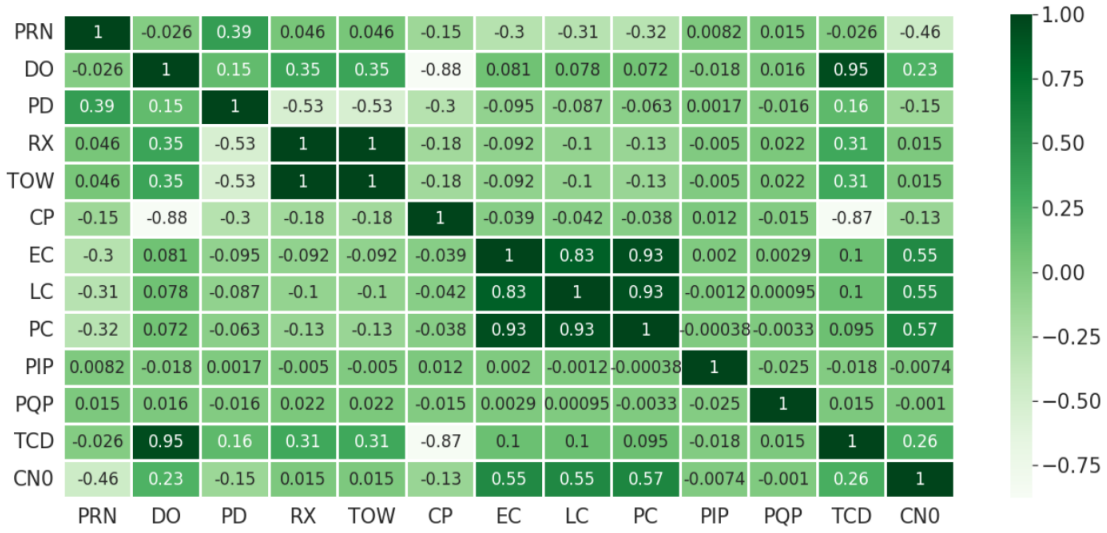


Figure 4.1. Spearman's Correlation Coefficient Heatmap for the used models

features are less important compared to their correlated pairs, namely LC, EC, DO, and TOW. Hence, these three features are removed from the corresponding dataset. Consequently, ten features, namely PRN, DO, TOW, PD, CP, LC, EC, PIP, PQP, and CN0, are considered relevant and uncorrelated features for classifying GPS spoofing attacks on UAVs (Refer to Table 4.3).

The results of the selected ensemble algorithms are shown in Figure 4.3 in terms of ACC and PD. As one can see, the stacking model has the highest ACC **4.4.2**

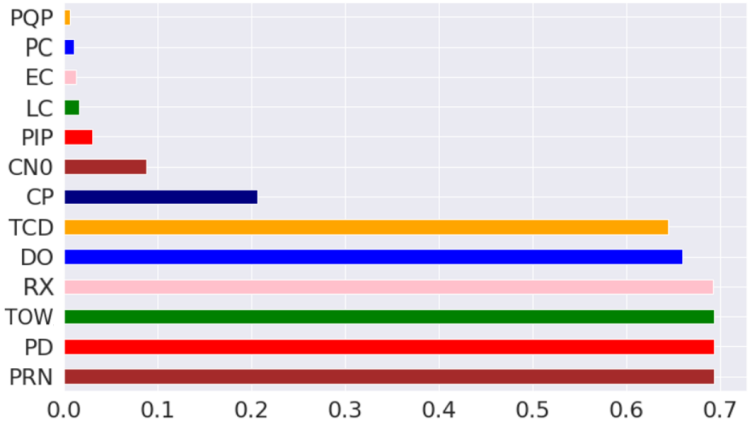


Figure 4.2. Mutual information feature selection method

Table 4.3. List of selected features

Feature pairs	Discarded features
(RX, TOW)	PC, TCD, and RX
(TCD, DO)	
(LC, PC)	
(EC, PC)	

4.4 Result of the weak learners

(95.43%), followed by the bagging (95.28%), and then the boosting model (94.61%). Therefore, these results show the stacking model provides the best accuracy for detecting GPS spoofing attacks. However, the accuracy is not sufficient to compare the efficiency of ML models in detecting GPS spoofing attacks. The number of falsely detected alarms and misdetected samples can decrease the performance of ML models. Figure 4.3b shows the results of the selected models in terms of PD. As one can see, the stacking classifier has the highest detection probability at 99.56%, the bagging classifier has a detection probability of 99.24%, and the boosting model has a detection probability of 96.55%, which is considered the lowest result compared to the two other

ensemble models.

Figure 4.4a shows the probability of misdetection of the selected ensemble models. As one can see, the stacking classifier has a PMD of 0.36%, the bagging model shows a PMD of 0.64%, and the boosting model has a PMD of 2.95%. Consequently, the stacking model has the lowest PMD, whereas the boosting model has the highest and worse PMD. Figure 4.4b illustrates the results of the PFA of the selected models. The stacking classifier has the best result in terms of the PFA (0.43%), followed by the

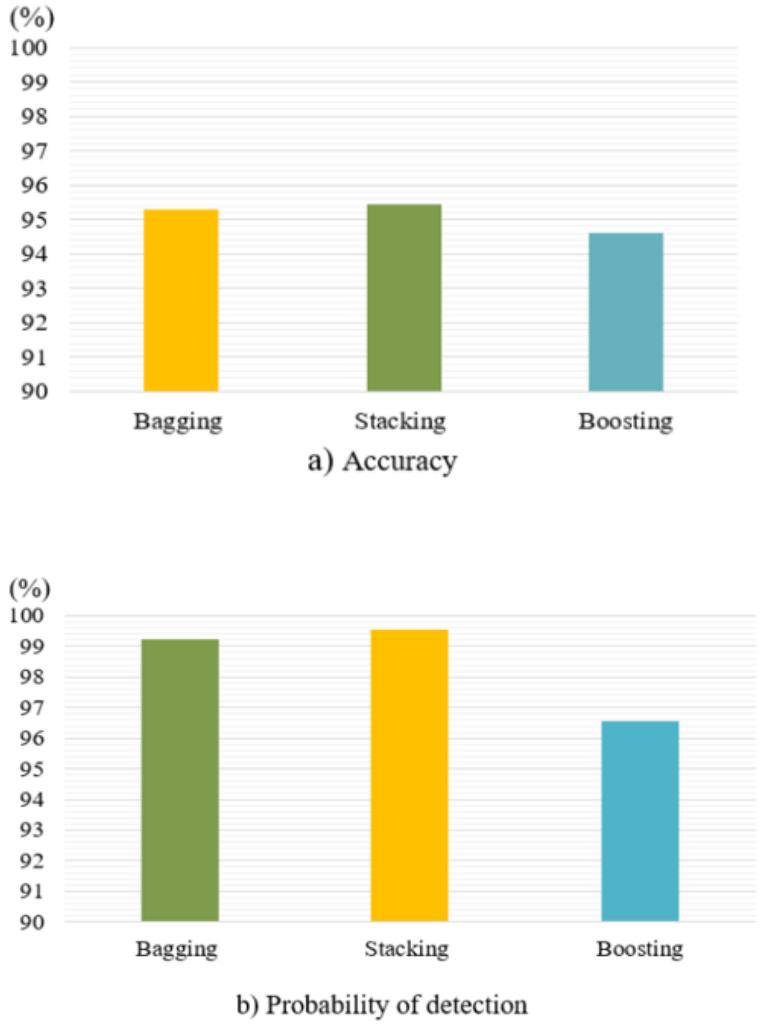


Figure 4.3. Evaluation metrics of the strong learners in terms of ACC and PD

bagging model (1.07%) and then boosting classifier (5.08%).

As shown in the tables above, the stacking model obtains the best results in terms of all performance evaluation metrics among strong learner classifiers. It achieves a 95.43% ACC, a 99.56% PD, a 0.36% PMD, and 0.03% PFA. In contrast, the boosting model provides the best result in terms of all evaluation metrics. This model shows a 94.61% ACC, a 96.55% PD, a 2.95% PMD, and a 5.08% PFA. In contrast, the stacking model has a PFA of 1.6%, which is 0.51% higher than the PFA

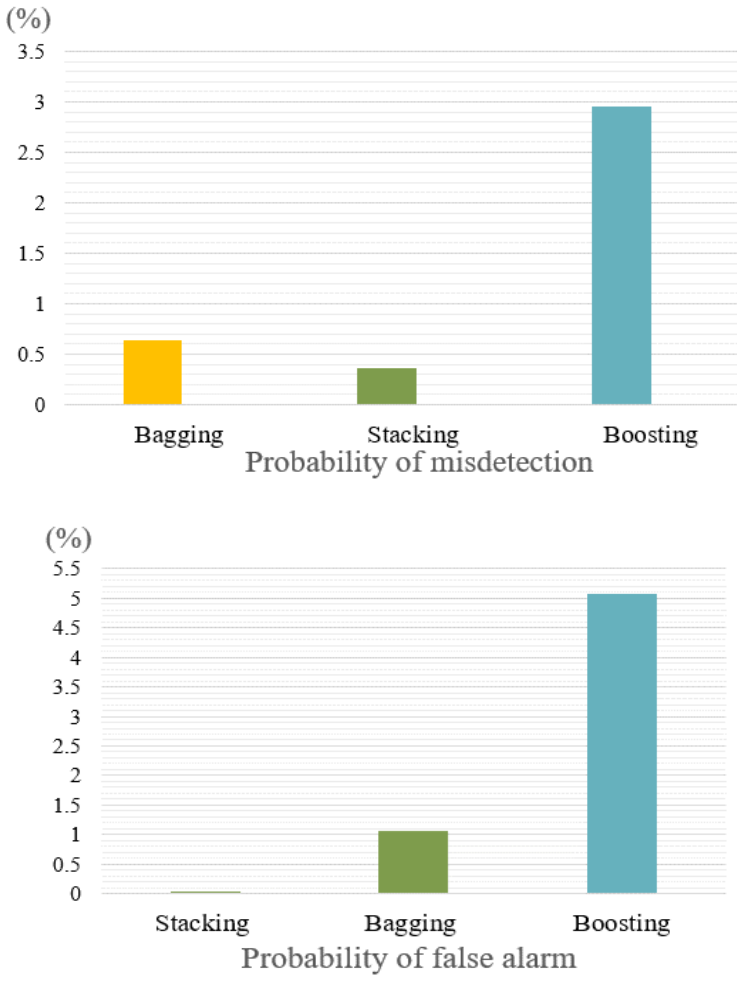


Figure 4.4. Evaluation metrics of the strong learners in terms of PMD and PFA

of these proposed methods. In addition, this stacking model has an ACC of 99.7% and a PD of 99.8%, which are 0.1% lower than the ACC and PD of the proposed methods.

Figure. 4.5 gives the results of the memory size, processing time, and average prediction time of each sample for each model. As one can see in this table, the stacking classifier presents the worst outcomes in terms of processing time and average prediction time compared with the other ensemble techniques. This is followed by the bagging model (190.4 MB) and then by the boosting method (190.5 MB). The stacking model has a processing time of 13.06 seconds, the bagging model has 0.74 seconds, and boosting model has 1.5 seconds. As a result, the bagging classifier provides the best results in terms of processing time, followed by bagging and stacking models. Finally, the stacking classifier has the worst average prediction time of 0.24 seconds per instance.

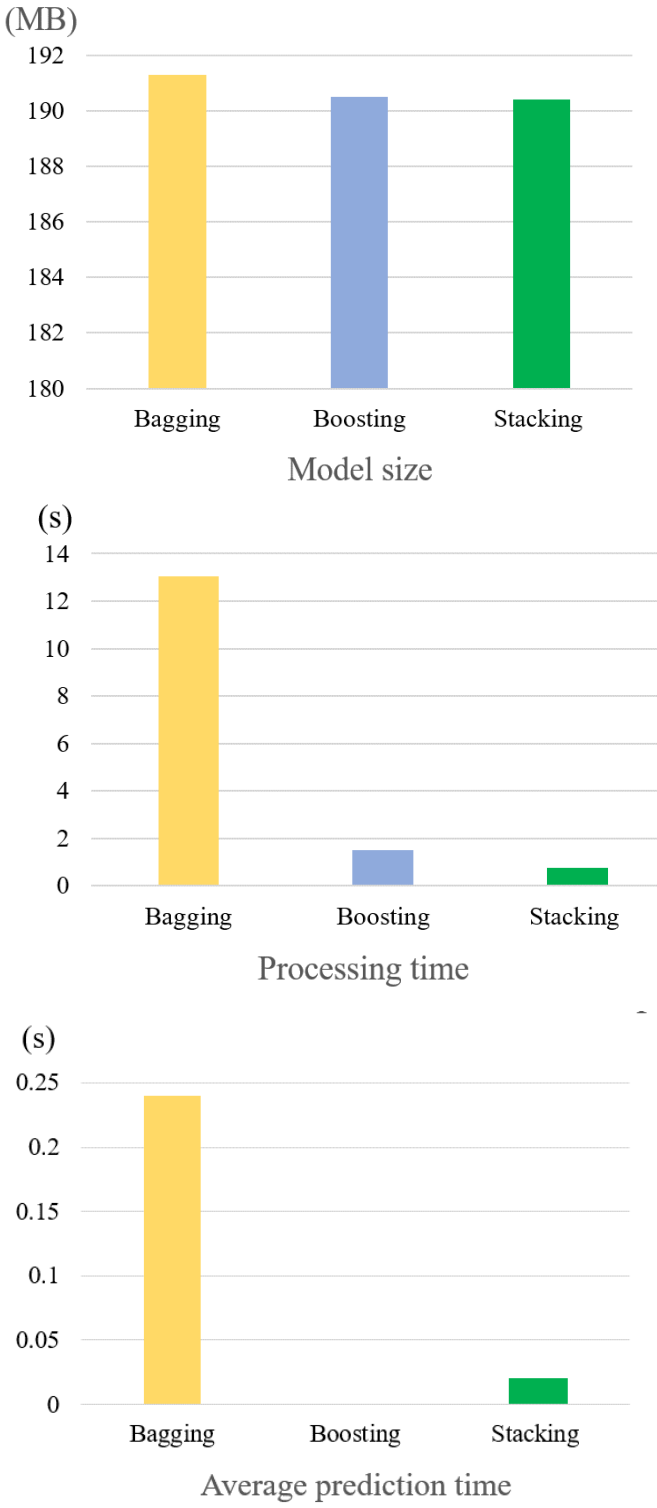


Figure 4.5. Performance comparison to target the SWaP limitations for strong learners

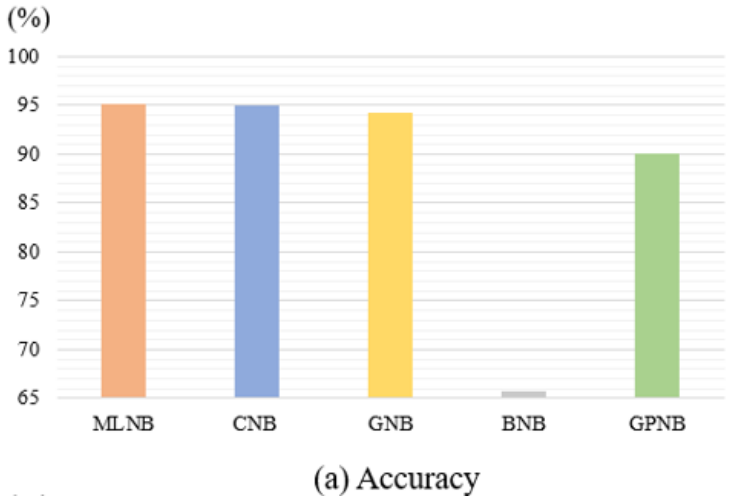
4.5 Result of the weak learners

The results of the selected NB algorithms are shown in Figure 4.6 in terms of ACC and PD. Figure 4.6a shows the accuracy of all five classifiers. As one can see, the MLNB model obtains the highest ACC (95.10%), followed by the CNB (95.00%), then the GNB (94.30%), GPNB (90.00%), and BNB (65.80%). Therefore, these results show that the MLNB model provides the best accuracy for detecting GPS spoofing attacks among weak learners.

Figure 4.6b shows the results of the selected models in terms of PD. The GNB classifier has the highest detection probability of 99.80%, the CNB classifier has a detection probability of 92.00%, the MLNB classifier has a detection probability of 91.50%, the GPNB model has a detection probability of 80.33%, and the BNB model obtains a detection probability of 56.30%, which is the lowest result compared to the two other ensemble models.

Figure 4.7a shows the probability of misdetection of the selected NB models. The MLNB and CNB classifiers have a PMD of 1.00%; the GNB model shows a PMD of 2.50%, the GPNB model shows 16.67%, and the BNB model shows 43.47%. Consequently, the MLNB and GNB models obtain the lowest PMD, whereas the BNB model has the highest and worse PMD.

Figure 4.7b illustrates the results of the PFA of the selected models. In addition, the MLNB classifier has the best result in terms of the PFA (0.10%), followed by the



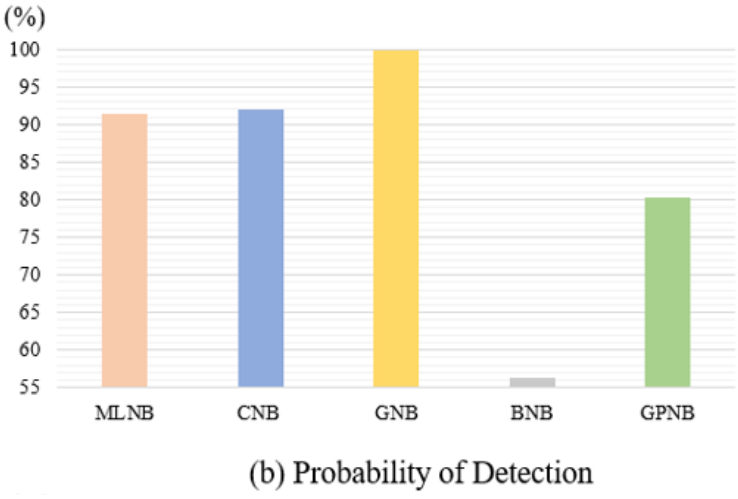


Figure 4.6. Evaluation metrics of the weak learners in terms of ACC and PD

CNB model (0.16%) and then the GNB classifier (3.20%), GPNB classifier (3.40%), and BNB model (23.20%). As shown in the tables above, the MLNB classifier obtains the best results in terms of all performance evaluation metrics among strong learner classifiers. It achieves a 1.00% of PMD, a 95.10% of ACC, and 0.10% PFA, while the GNB classifier obtains the lowest and better PD of 99.80%. In contrast, the BNB model provides the worst results in terms of all evaluation metrics. This model shows a

65.80% ACC, a 56.30% PD, a 43.47% PMD, and a 23.20% PFA.

Figure 4.8 gives the results of the size in memory, the processing time, and the average prediction time of each sample for each model. As one can observe, the MLNB and CNB have the lowest and best memory size (170.2 MB) among all NB classifiers. The BNB and GPNB classifiers show the best and worst performance in terms of memory size, respectively, while they achieve the best results in terms of prediction

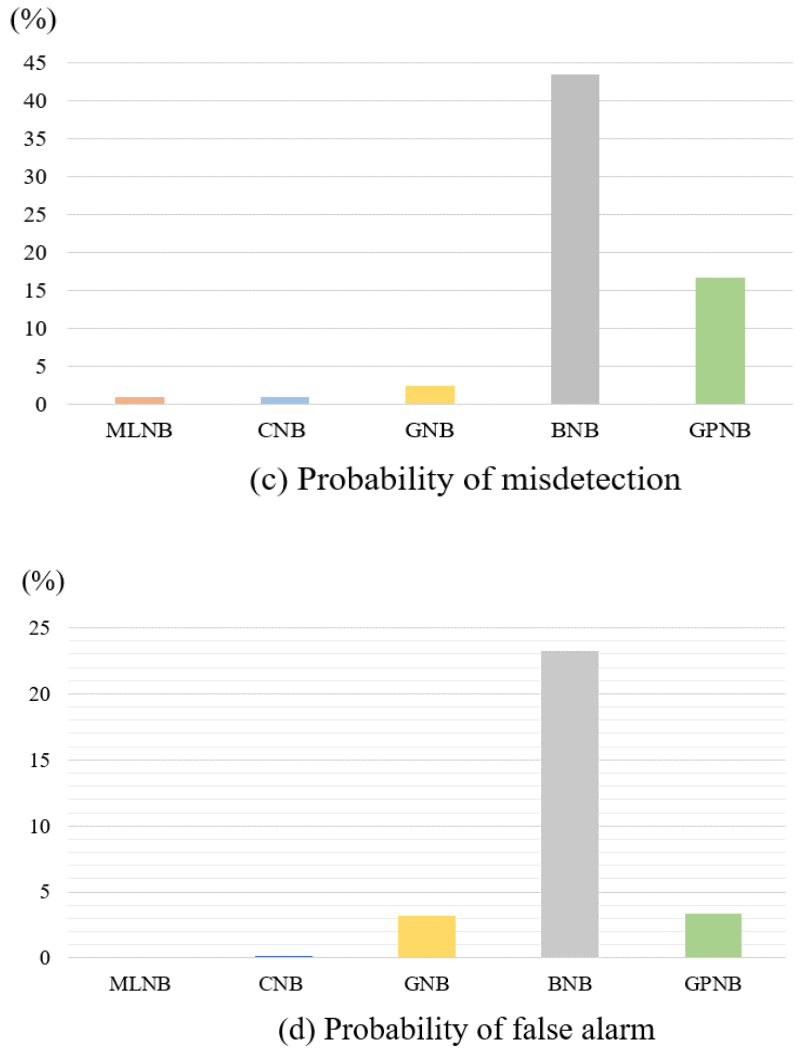


Figure 4.7. Evaluation metrics of the weak learners in terms of PMD and PFA

time compared to other NB algorithms. The BNB model achieves the lowest and best prediction time (0.003s). In contrast, the GPNB model achieves the worst outcome in terms of prediction time, followed by the CNB model. The CNB and MLNB classifiers achieve a prediction time of 0.099s and 0.095s, respectively. However, the BNB model achieves the shortest and best processing time compared to other NB classifiers. According to the results, the GNB classifier also obtains moderately similar results to the CNB classifier in terms of processing time at 0.580s. In contrast, the GPNB classifier obtains the longest and worst processing time of 1.890s.

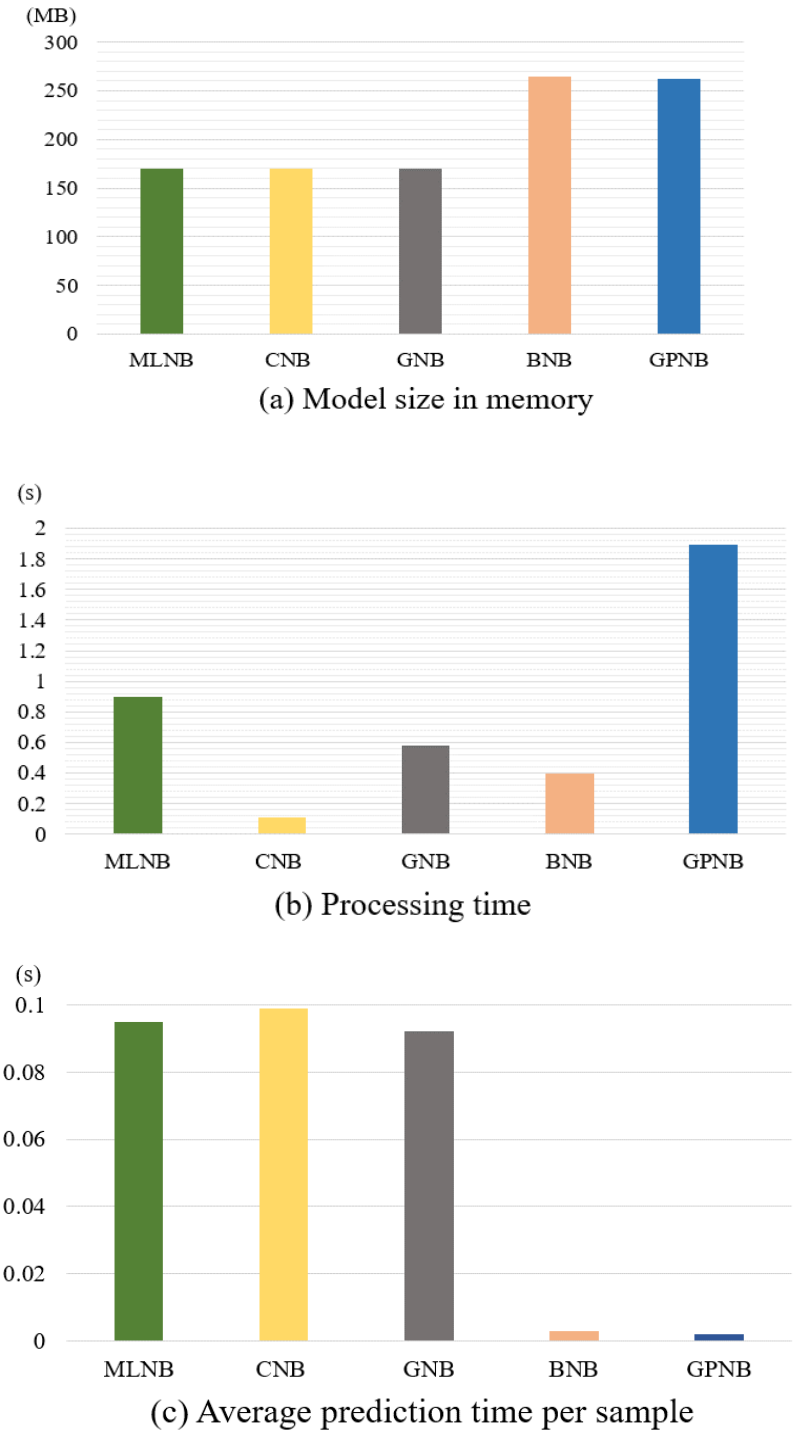


Figure 4.8. Performance comparison to target the SWaP limitations for weak learners

4.6 Comparison results between strong and weak learners in terms of the main evaluation metrics

In this section, we present the results of the two classifier sets, namely ensemble and Naïve Bayes, in terms of the main evaluation metrics (ACC, PD, PMD, PFA). The detailed overview is illustrated in Table 4.4. It is worth mentioning that despite structural differences between the strong and weak learners, the overall outcomes of both classifiers are relatively close to each other. One can observe that the member of the ensemble model set, stacking obtains an ACC of 95.43%, which is considered the highest accuracy among all classifiers.

For the PD metric, the weak learner classifier: the GNB model slightly outperforms (0.5 times) the best-performer stacking ensemble model. However, for the PMD metric, the stacking classifier achieves the lowest and best results (0.36%) among all implemented classifiers. For the last metric (PFA), the MLNB model performs 10.7 times better compared to stacking (the ensemble model). Consequently, the stacking model achieves the best results in terms of ACC, PMD, and PFA among all classifiers.

In addition, the second best performers, bagging and CNB models achieve a similar result as the best performers, namely stacking and MLNB. The bagging model obtains an ACC of 95.20%, while the weak learner member, CNB, obtains an ACC of 95.00%. In addition, the stacking model outperforms the CNB model by a factor of 2.8 in terms of the PMD. In the last metric, PFA, the CNB model performs three times better compared to the ensemble model (bagging).

The third best-performers for each classifier set show that the member of the ensemble model (boosting) only slightly outperforms (1.2 times) the weak learner

model (GNB) in terms of ACC, while it achieves noticeably better results than the boosting model, in terms of the other three metrics, including PD, PMD, and PFA. The boosting model obtains 1.1 times higher PD, three times lower PMD, and 1.6 lower PFA.

The last model of the NB, the GPNB classifier, achieves the worst results among all the ML classifiers in terms of all four metrics. Consequently, it is worth mentioning there is an immense difference between the ACC and PD results of the worst-performer of the weak (BNB model) and strong (boosting model) learner categories. The boosting classifier obtains overall about 25 times higher and better results than the BNB classifier in terms of all classifiers.

To sum up, the comparison of the best-performers of the ensemble and NB models obtain relatively close results to each other in terms of four main evaluation metrics. The strong learners obtain the best outcomes for the three main performance metrics, namely ACC, PFA, and PMD, while weak learners obtain higher results in terms of PMD, which is considered the best result as shown in Table 4.5.

Table 4.4. Comparison of the strong and weak learners in terms of four main evaluation metrics

	Models	**ACC (%)**	**PD (%)**	**PMD (%)**	**PFA (%)**
Strong learners	Bagging	95.28	99.24	0.64	1.07
	Stacking	95.43	99.56	0.36	0.03
	Boosting	94.61	96.55	2.95	5.08
Weak learners	MLNB	95.10	91.50	1.00	0.10
	GNB	94.30	99.80	2.50	3.20
	CNB	95.00	92.00	1.00	0.16
	BNB	65.80	56.30	43.47	23.2
	GPNB	90.00	80.33	16.67	3.40

Table 4.5. Best performance results among weak and strong learners in terms of main evaluation metrics

Main evaluation metrics	**Best performance results among weak and strong learners**	
	Strong learners	**Weak learners**
ACC	Stacking	-
PD	-	GNB
PMD	Stacking	-
PFA	Stacking	-

4.7 Comparison results between strong and weak learners in terms of the size and time metrics

In this section, we will evaluate the strong and weak learners in terms of time performance and size metrics, as demonstrated in Table 4.6. When we investigate the memory size of all algorithms, we can observe the results of the evaluation metrics are close to each other for each classifier family. CNB and BNB models obtain the same memory size of 170.2MB, which is considered the best result among weak learners. This is also considered the best outcome for both strong and weak classifier categories. The highest and lowest outcomes are achieved by the stacking and BNB model (0.74s and 0.4s) in terms of processing time. It is worth mentioning that, despite the stacking model consisting of five weak learners, it achieves the best processing time among strong learners.

When we compare the worst results of the strong and weak learner family models, we can observe that the outcome of the weak learner model, GPNB, is 7.9 times lower than the bagging classifier, which is considered noticeably a better performance in terms of processing time. The best result of the ensemble model boosting obtains five times lower results compared to the best performance of the NB model: BNB. Similar to the best results, the worst outcome of the average prediction time for the ensemble model: bagging is 2.4 times higher compared to the NB model: CNB, which is considered the worst outcome among all models.

To conclude the analysis of time performance and memory size, weak learners: MLNB, BNB, and GPNB classifiers obtain the best performance, while strong learners

achieve lower results. The results of the four metrics for all models are given in Table 4.7.

Table 4.6. Performance comparison of the strong and weak learners in terms of size and performance metric

	Classification Model	**Processing Time (s)**	**Model Size (MB)**	**Average Prediction Time (s)**
Strong learners	Boosting	1.50	190.5	0.010
	Bagging	13.06	191.3	0.240
	Stacking	0.74	190.4	0.020
Weak learners	MLNB	0.90	170.2	0.095
	CNB	0.11	170.2	0.099
	GNB	0.58	170.3	0.090
	BNB	0.40	264.3	0.003
	GPNB	1.89	261.7	2.100

Table 4.7. Best results among weak and strong learners in terms of size and performance metrics

Time performance and memory size metrics	**Best results among weak and strong learners**	
	Strong learners	**Weak learners**
Processing time	-	BNB
Memory size	-	MLNB
Prediction time per sample	-	BNB

Chapter 5

CONCLUSIONS

Unmanned Aerial Vehicles (UAVs) or drones have become increasingly popular in a variety of fields. As a result, more than 10,000 commercial drones will be operating over the next ten years [3]. This is mostly owing to their cost and budget benefits over commercial helicopters [80, 81]. Furthermore, technical innovation allows for simple manipulations using cellphones to fly mini-drones rather than utilizing remote controls.

UAVs rely on GPS receivers for the safety of return-to-home missions during medium to long-distance flights. The GPS radio frequency connection, known as the L1 channel, is used for civilian UAV applications [56]. However, these signals are unencrypted, making them subject to GPS jamming and spoofing. In GPS spoofing attacks, a malicious user can transmit counterfeit GPS signals in this attack and can change the UAV's flight without being noticed. A successful GPS spoofing assault can inflict significant material damage as well as human injury.

Several approaches for detecting GPS attacks have been proposed. However, some of these technologies are inefficient since they are unreliable and have low accuracy, detection probability, false alarm, and misdetection. Moreover, additional hardware and protocol adjustments are required to use these solutions.

To address these problems, this thesis provides a comparison based on

supervised machine learning techniques to detect GPS spoofing attacks [2]. In Chapter 2, Global Positioning Systems and attacks against these systems were reviewed. Moreover, the advantages and limitations of cutting-edge security systems were also evaluated. It was demonstrated that the existing strategies for the security of UAV systems have various issues.

In Chapter 3, the approaches for building attack scenarios and training datasets were detailed. In this thesis, we used a dataset which is implemented in [56]. Real-time experiments were conducted to collect a dataset. Several features are retrieved from the data based on three types of attacks, including simplistic, intermediate, and sophisticated. In addition, feature selection techniques, namely Mutual information and Spearman correlation were used to improve the quality of the dataset. Moreover, two hyperparameter optimization techniques, namely grid search and genetic algorithm were implemented to determine the optimal hyperparameters for each model.

In Chapter 4, the performance of the two different machine learning categories, strong and weak learners, was analyzed. For the evaluation, four main metrics were used, namely probability of detection, probability of misdetection, probability of false alarm, and accuracy. These metrics are effective for the evaluation and selection of the most suitable model to detect GPS spoofing attacks. In addition to these metrics, three others were used that are related to the size, weight, and power constraints. These include processing time, prediction time for each instance, and memory size.

According to the results, a strong learner classifier (stacking) achieves the best results in terms of accuracy, probability of misdetection, and false alarm are achieved, which are 95.43%, 0.36%, and 0.03%, respectively. In contrast, the weak learner classifier (the

GNB model) achieves better results in terms of the probability of detection, which is equal to 99.80%. In addition, according to the comparison results, the MLNB algorithm obtains a 0.4s processing time, the BNB algorithm obtains 0.003s, and the GPNB algorithm obtains a 170.2MB memory size, which is considered the lowest and best results among all ML classifiers implemented in this study.

In conclusion, strong learner classifiers outperform weak learners in terms of the main evaluation metrics, namely accuracy, misdetection probability, and false alarm probability, while weak learner classifiers obtain the best results in terms of the metrics to target size, weight, and power limitations. To conclude, we can state both detection algorithm categories provide good results in detecting GPS spoofing attacks on UAVs. However, there are still several unresolved issues with the security of UAVs that have to be addressed with acceptable approaches.

One promising research direction is the implementation of online unsupervised machine learning techniques, which can categorize and train unlabeled data. A potential research area in this respect is to conduct a comparative analysis to investigate the performance of deep learning algorithms to identify the GPS spoofing attacks targeting UAVs. Another area of future study is to explore and create countermeasure techniques after an attack has been identified.

BIBLIOGRAPHY

[1] M.R.Manesh and N.Kaabouch, “Cyber-attacks on unmanned aerial system networks: Detection, countermeasure, and future research directions,” Computers & Security, vol. 85, pp. 386–401, 2019.

[2] Federal Aviation Administration, “Air Traffic by the Numbers,” June 2019. Availableat:https://www.faa.gov/air_traffic/by_the_numbers/media/Air_Traffic_by_the_Numbers_ 2019.pdf.

[3] flightradar24 Live Air Traffic [online] Available at: www.flightradar24.com

[4] D. B. Flamholz, A. M. Annaswamy, and E. Lavretsky, “Baiting for defense against stealthy attacks on cyber-physical systems,” in AIAA Scitech 2019 Forum, 2019.

[5] G. C. Giannatto, C. Markowsky, and G. Giannatto, “Potential vulnerabilities of the nextgen air traffic control system,” in Proceedings of the International Conference on Security and Management (SAM), 2014.

[6] McCallie, J. Butts, and R. Mills, Security Analysis of the ADS-B Implementation in the Next Generation Air Transportation System, Int. J. Critical Infrastructure Protection, vol. 4, no. 2, pp. 78–87, 2011.

[7] S. Z. Khan, M. Mohsin, and W. Iqbal, “On GPS spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions,” " PeerJ Computer Science, vol. 7, p. e507, 2021.

[8] H. Sedjelmaci, S. M. Senouci, and N. Ansari, “A hierarchical detection and

response system to enhance security against lethal cyber-attacks in UAV networks," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 48, no. 9, pp. 1594–1606, 2017.

[9] S. Tohidi and M. R. Mosavi, "Effective detection of GNSS spoofing attack Using A multi-layer perceptron neural network classifier trained by PSO," 25th International Computer Conference, Computer Society of Iran (CSICC), pp. 1–5, 2020.

[10] R. Mitchell and R. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," IEEE transactions on systems, man, and cybernetics systems, vol. 44, no. 5, pp. 593–604, 2013.

[11] G. Falco, "Cybersecurity principles for space systems," Journal of Aerospace Information Systems, vol. 16, no. 2, pp. 61–70, 2019.

[12] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," Networks, vol. 72, no. 4, pp. 411–458, 2018.

[13] G. S. C. Avellar, G. A. S. Pereira, L. C. A. Pimenta, and P. Iscold, "Multi-UAV routing for area coverage and remote sensing with minimum time," Sensors (Basel), vol. 15, no. 11, pp. 27783–27803, 2015.

[14] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," IEEE Trans. Syst. Man Cybern. Syst., vol. 47, no. 1, pp. 70–85, 2017.

[15] F. Guerriero, R. Surace, V. Loscrí, and E. Natalizio, "A multi-objective approach for unmanned aerial vehicle routing problem with soft time windows

constraints," Appl. Math. Model., vol. 38, no. 3, pp. 839–852, 2014.

[16] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: several worst-case results," Optim. Lett., vol. 11, no. 4, pp. 679–697, 2017.

[17] I. Jawhar, N. Mohamed, J. Al-Jaroodi, and S. Zhang, "A framework for using unmanned aerial vehicles for data collection in linear wireless sensor networks," J. Intell. Robot. Syst., vol. 74, no. 1–2, pp. 437–453, 2014.

[18] K. Li, W. Ni, X. Wang, R. P. Liu, S. S. Kanhere, and S. Jha, "Energy-efficient cooperative relaying for unmanned aerial vehicles," IEEE Trans. Mob. Comput., vol. 15, no. 6, pp. 1377–1386, 2016.

[19] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery, Transp," Transp. Res. C Emerg. Technol, vol. 54, pp. 86–109, 2015.

[20] M. Ahmad, M. A. Farid, S. Ahmed, K. Saeed, M. Asharf, and U. Akhtar, "Impact and detection of GPS spoofing and countermeasures against spoofing," in 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019, pp. 1–8.

[21] M. Lenhart, M. Spanghero, and P. Papadimitratos, "Distributed and Mobile Message Level Relaying/Replaying of GNSS Signals In Proceedings of the 2022 International Technical Meeting of The Institute of Navigation," pp. 56–67, 2022.

[22] C. Sun, J. W. Cheong, A. G. Dempster, H. Zhao, and W. Feng, "GNSS spoofing detection by means of signal quality monitoring (SQM) metric combinations," IEEE Access, vol. 6, pp. 66428–66441, 2018.

[23] J. Wilson, R. S. Wahby, H. Corrigan-Gibbs, D. Boneh, P. Levis, and K. Winstein, “Trust but verify: Auditing the secure Internet of things,” in Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, 2017, pp. 464–474.

[24] B. Yuksek, E. Saldiran, A. Cetin, R. Yeniceri, and G. Inalhan, "A model-based flight control system design approach for the micro aerial vehicle using integrated flight testing and HIL simulation," In AIAA Scitech 2019 Forum, p. 1480, 2019.

[25] M. Reham, “Security vulnerabilities of cyberphysical unmanned aircraft systems,” IEEE Aerospace and Electronic Systems Magazine, vol. 33, no. 9, pp. 4–17, 2018.

[26] S. Wang, J. Wang, C. Su, and X. Ma, “Intelligent detection algorithm against UAVs’ GPS spoofing attack,” in 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), 2020.

[27] C.-P. Roberto, A. Bhattacharya, G. Bovet, and D. Giustiniano, “LSTM-based GNSS Spoofing Detection Using Low-cost Spectrum Sensors,” IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2020), 2020.

[28] L. Meng, Y. Lin, R. Shuangyin, T. Gaigai, Z. Long, Y. Feng, and Y. Wu. "An approach of linear regression-based UAV GPS spoofing detection." Wireless Communications and Mobile Computing, 2021.

[29] H. Zeeshan, and S. Khalid, "Survey on effective GPS spoofing countermeasures." International Conference on Innovative Computing Technology (INTECH), pp. 573-577. IEEE, 2016.

[30] H. Hildmann, and K. Ernö "Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security, and public safety." Drones 3, no. 3, pp. 59, 2019.

[31] A. Y. Javaid, F. Jahan, and W. Sun, “Analysis of global positioning system-based attacks and a novel global positioning system spoofing detection/mitigation algorithm for unmanned aerial vehicle simulation,” Simulation, vol. 93, no. 5, pp. 427–441, 2017.

[32] D. Mendes, N. Ivaki, and M. Henrique, "Effects of GPS spoofing on unmanned aerial vehicles." In 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 155-160. IEEE, 2018.

[33] A. Chakrabarty, A. M. Robert, X. Bouyssounouse, and R. Hunt. "An integrated system for autonomous search and track with a small unmanned aerial vehicle." In AIAA Information Systems-AIAA Infotech Aerospace, pp. 0671, 2017.

[34] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, “Security analysis of drones systems: Attacks, limitations, and recommendations,” Internet of Things, p. 100218, 2020.

[35] Z. Feng, N. Guan, Lv, M., Liu, W., Deng, Q., Liu, X., & Yi, W, "Efficient drone hijacking detection using onboard motion sensors." Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1414-1419. IEEE, 2017.

[36] Y. Qiao, Z. Yuxing, and X. Du, "A vision-based GPS-spoofing detection method for small UAVs." International Conference on Computational Intelligence and Security (CIS), pp. 312-316. IEEE, 2017.

[37] Y. Hu, B. Shaofeng, C. Kejin, and Bing Ji, "GNSS spoofing detection based on new signal quality assessment model." GPS Solutions 22, no. 1,2018.

[38] M. Varshosaz, A. Afary, B. Mojaradi, M. Saadatseresht, P. E. Ghanbari, "Spoofing detection of civilian UAVs using visual odometry." ISPRS International Journal of Geo-Information 9, no. 1, 2020.

[39] T. T Khoei, S. Ismail, and N. Kaabouch, "Dynamic Selection Techniques for Detecting GPS Spoofing Attacks on UAVs." Sensors 22, no. 2, p. 663, 2022.

[40] G. Liu, R. Zhang, C. Wang, and L. Liu, “Synchronization-free GPS spoofing detection with crowdsourced air traffic control data,” International Conference on Mobile Data Management (MDM). IEEE, pp. 260–268, 2019.

[41] Z. Feng, M. L. Nan Guan, L. Wenchen, D. Qingxu, L. Xue, and Y. Wang, "Efficient drone hijacking detection using two-step GA-XGBoost." Journal of Systems Architecture 103, 101694, 2020.

[42] Q.Sun, M. Xinyu, G. Zhihao, W. Jin, and G. Demin, "Spoofing Attack Detection Using Machine Learning in Cross-Technology Communication." Security and Communication Networks, 2021.

[43] S. Semanjski, I. Semanjski, W. D. Wilde, and A. Muls, "Use of supervised machine learning for GNSS signal spoofing detection with validation on real-world meaconing and spoofing data—part i." Sensors 20, no. 4,1171, 2020.

[44] Xue, Nian, Liang Niu, Xianbin Hong, Zhen Li, Larissa Hoffaeller, and Christina Pöpper. "Deepsim: Gps spoofing detection on UAVs using satellite imagery matching." In Annual computer security applications conference, pp. 304-319. 2020.

[45] L. A. van Mastrigt, A. J. van der Wal and P. J. Oonincx, "Exploiting the Doppler effect in GPS to monitor signal integrity and to detect spoofing," *2015 International Association of Institutes of Navigation World Congress (IAIN)*, 2015, pp. 1-8

[46] A. M. Pushpa. "Detecting signal spoofing and jamming attacks in UAV networks using a lightweight IDS." In 2019 international conference on computer, information, and telecommunication systems (CITS), pp. 1-5. IEEE, 2019.

[47] S. P. Arteaga, L. A. M. Hernández, G. S. Pérez, A. L. S. Orozco, and L. J. G. Villalba. "Analysis of the GPS spoofing vulnerability in the drone 3DR solo." IEEE Access 7, pp. 51782-51789, 2019.

[48] D. Mendes, N. Ivaki, and M. Henrique, "Effects of GPS spoofing on unmanned aerial vehicles." In 2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 155-160. IEEE, 2018.

[49] F. S. Betti, M. Conti, C, M. Pinotti, and G. Rigoni. "UAVs path deviation attacks: Survey and research challenges." In 2020 IEEE International Conference on Sensing, Communication, and Networking (SECON Workshops), pp. 1-6. IEEE, 2020.

[50] E.Gentilho, P. R. Scalassara, and T. Abrão. "Direction-of-arrival estimation methods: A performance-complexity tradeoff perspective." Journal of Signal Processing Systems 92, no. 2, 239-256, 2020.

[51] M. Aljehani, I. Masahiro, W. Akira, Y, Taketoshi, O. Fumiya, and I. Hidemasa. "UAV communication system integrated into network traversal with mobility." SN Applied Sciences 2, no. 6, 1-20, 2020.

[52] S. Park, T. K. Hyeong, S. Lee, H. Joo, and H. Kim. "Survey on anti-drone

systems: Components, designs, and challenges." IEEE Access 9, PP. 42635-42659, 2019.

[53] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. G. Rodriguez, and J. Yuan. "Survey on UAV cellular communications: Practical aspects, standardization advancements, regulation, and security challenges." IEEE Communications surveys & tutorials 21, no. 4, 2019, PP. 3417-3442.

[54] G. Lykou, D. Moustakas, and D. Gritzalis. "Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies." Sensors 20, no. 12, 2020, 3537.

[55] G. Aissou, S. Benouadah, H. El Alami, and N. Kaabouch, “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), 2022.

[56] G. Aissou, H. O. Slimane, S. Benouadah, and N. Kaabouch. "Tree-based supervised machine learning models for detecting GPS spoofing attacks on UAS," 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 0649-0653. IEEE, 2021.

[57] S. Ismail, T. T. Khoei, R. Marsh, and N. Kaabouch, “A comparative study of machine learning models for cyber-attacks detection in wireless sensor networks,” in 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2021, pp. 1–4.

[58] H. Schütze, C. D. Manning, and R. Prabhakar, “Introduction to information retrieval,” Cambridge: Cambridge University Press, Vol. 39, 2008.

[59] S. Semanjski, A. Muls, I. Semanjski, and W. De Wilde, “Use and validation of supervised machine learning approach for detection of GNSS signal spoofing,” in International Conference on Localization and GNSS (ICL-GNSS). IEEE, 2019, pp. 1–6.

[60] P. Mehta et al., “A high-bias, low-variance introduction to Machine Learning for physicists,” Phys. Rep., vol. 810, pp. 1–124, 2019.

[61] N. S. Harzevili, and S. H. Alizadeh, "Mixture of latent multinomial naive Bayes classifier." Applied Soft Computing 69, 516-527, 2018. Berrar, Daniel. "Bayes’ theorem and naive Bayes classifier." Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics 403, 2018.

[62] R. Malhotra, and M. Cherukuri, "Software Defect Categorization based on Maintenance Effort and Change Impact using Multinomial Naïve Bayes Algorithm." In 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), pp. 1068-1073. IEEE, 2020.

[63] H. Amir, H. J. Mieke, A. E. Schulte, S. Y. Goudriaan, and A. Foo, “Optimized Naive-Bayes and decision tree approach for fMRI smoking cessation classification,” Complexity, 2018.

[64] D. R. Jason, S. Lawrence, T. Jaime, and D. R. Karger. "Tackling the poor assumptions of naive Bayes text classifiers." In Proceedings of the 20th international conference on machine learning (ICML-03), pp. 616-623. 2003.

[65] G. Xiao, Q. Cheng, and C. Zhang, "Detecting travel modes using rule-based classification system and Gaussian process classifier." IEEE Access 7,116741-116752, 2019.

[66] Griffis, Joseph C., Jane B. Allendorfer, and Jerzy P. Szaflarski. "Voxel-based Gaussian naïve Bayes classification of ischemic stroke lesions in individual T1-weighted MRI scans." Journal of neuroscience methods 257, 2016, pp. 97-108.

[67] J. Yu, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimizing XGBoost training parameters." IEEE Access 7, pp. 118310-118321, 2019.

[68] McCallum, Andrew, and Kamal Nigam. "A comparison of event models for naive Bayes text classification." In AAAI-98 workshop on learning for text categorization, vol. 752, no. 1, pp. 41-48. 1998.

[69] M. H. D. M. Ribeiro and L. dos Santos Coelho, “Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series,” Applied Soft Computing, vol. 86, p. 105837, 2020.

[70] S. Maryam, G. Martínez-Muñoz, and A. Suárez., “Building heterogeneous ensembles by pooling homogeneous ensembles,” International Journal of Machine Learning and Cybernetics, vol. 13, no. 2, pp. 551-558., 2022.

[71] Breiman, Leo. "Bagging predictors." Machine learning 24, no. 2 (1996): 123-140.

[72] J. H. Friedman, "Stochastic gradient boosting." Computational statistics & data analysis 38, no. 4 2002, pp. 367-378.

[73] D. H. Wolpert, "Stacked generalization." Neural networks 5, no. 2, 1992, pp. 241-259

[74] H. Dalkilic, H. Yildirim, S. N, Yesilyurt, and S. Pijush. "Daily Flow Modeling

with Random Forest and K-Nearest Neighbor Methods." Erzincan University Journal of Science and Technology 14, no. 3, 2021, pp. 914-925.

[75] H. Mahmudul, Md M. Islam, Md I. I. Zarif, and M. M. A. Hashem, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches." Internet of Things 7, 100059, 2019.

[76] O. Nachum, Y. Chow, B. Dai, and L. Li. "Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections." Advances in Neural Information Processing Systems 32, 2019.

[77] J. Yu, G. Tong, H. Yin, and N. Xiong, "A pedestrian detection method based on genetic algorithm for optimizing XGBoost training parameters." IEEE Access 7, pp. 118310-118321, 2019.

[78] T. T. Khoei, M. C. Labuhn, C. Toro, W. C. Hu, and N. Kaabouch, "A Stacking-based Ensemble Learning Model with Genetic Algorithm For detecting Early Stages of Alzheimer's Disease." IEEE International Conference on Electro Information Technology (EIT), pp. 215-222. IEEE, 2021.

[79] W. A. Setyo, and A. A. Supianto, "Hyperparameter optimization using a genetic algorithm on machine learning methods for online news popularity prediction." International Journal of Advanced Computer Science and Applications 9, no. 12, 263-267, 2018.

[80] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab., "Security analysis of drones systems: Attacks, limitations, and recommendations," Internet of Things, vol. 11, p. 100218, 2020.

[81] J. Price, Y. Li, K. A. Shamaileh, Q. Niyaz, N. Kaabouch, and V. Devabhaktuni, "Real-time classification of jamming attacks against UAVs via on-board software-defined radio and machine learning-based receiver module," in 2022 IEEE International Conference on Electro Information Technology (eIT), 2022.