2022 IEEE International Conference on Electro Information Technology (eIT) | 978-1-6654-8009-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/eIT53891.2022.9813819

ADS-B Message Injection Attack on UAVs: Assessment of SVM-based Detection Techniques

Hadjar Ould Slimane[1], Selma Benouadah[1], K. Al Shamaileh[2], V. Devabhaktuni[3], and N. Kaabouch[1]
[1]School of Electrical Engineering and Computer Science, University of North Dakota, Grand Forks, ND 58202, USA.
[2]Electrical and Computer Engineering Department, Purdue University Northwest, Hammond 46323, IN, USA.
[3]Electrical and Computer Engineering Department, The University of Maine, Orono 04469, ME, USA.

***Abstract*—As more aircraft are using the Automatic Dependent Surveillance-Broadcast (ADS-B) devices for navigation and surveillance, the risks of injection attacks are highly increasing. The exchanged ADS-B messages are neither encrypted nor authenticated while containing valuable operational information, which imposes high risk on the safety of the airspace. For this reason, we propose in this paper an SVM-based ADS-B message injection attack detection technique for UAV onboard implementation. First, we simulated several message injection attacks on real raw ADS-B data. Then, three Support Vector Machine (SVM) models were examined in terms of two types of assessment criteria, detection efficiency and model performance. The results show that the C-SVM model is the best fit for our application, with an accuracy of 95.32%.**

***Index Terms*—ADS-B, UAV, injection attacks, machine learning, SVM, detection techniques, wireless networks.**

I. Introduction

Automatic Dependent Surveillance-Broadcast (ADS-B) is a communication and surveillance technology designed to improve the reliability and efficiency of air navigation as an extension and, eventually, a substitute to RADARs. The U.S. Federal Aviation Administrations (FAA) [1] and the European Aviation Safety Agency (EASA) [2] have mandated that aircraft operating in their controlled airspace, including Unmanned Aerial Vehicles (UAVs), must be equipped with ADS-B devices beginning in January 2020 and June 2020, respectively.

ADS-B devices enable aircraft to send their coordinates, velocity, and other information to nearby aircraft and ground receivers every second. These continuous broadcasts greatly improve navigation safety and efficiency while reducing the risk of mid-air collisions. Two main operations characterize the ADS-B devices: ADS-B IN and ADS-B OUT. ADS-B IN devices receive ADS-B messages, whilst ADS-B OUT devices broadcast aircraft information. The ADS-B OUT is the one required by the two mandates previously mentioned [3].

ADS-B devices operate on two main frequency bands: 1090 MHz and 978 MHz. The datalink using the first band, referred to as 1090ES, carries traffic information, whereas the 978 MHz link handles Universal Access transceiver (UAT) broadcasts, which consists of aircraft traffic information along with other details such as weather. The 1090ES datalink is the most used ADS-B link and the one we are considering for this study [3].

Although ADS-B has brought many benefits to air traffic control, security was not a key issue in its design. As a matter of fact, ADS-B packets sent through the open 1090ES datalink are neither encrypted nor authenticated, making this technology vulnerable to a variety of cybersecurity attacks, such as message injection, eavesdropping, and jamming [4] [5]. Such malicious attacks can have serious consequences, such as increasing the risk of aircraft collisions. Therefore, efficient solutions must be developed to mitigate and reduce ADS-B vulnerabilities. In this work, we focus on ADS-B message injection attack detection on UAV networks since it imposes high-risk on-air navigation. Furthermore, the number and complexity of UAVs are increasing at a faster rate than ever before. According to FAA [6], they now account for the greatest number of aircraft, with over 800,000 registered drones.

In the literature, several strategies for detecting ADS-B injection attacks have been presented. These methods can be classified into five categories, namely, traffic modeling, group validation, physical layer fingerprint, data fusing, and machine learning.

In the first category, traffic modeling, a traffic pattern is predicted based on historical data and then compared to incoming data to detect irregularities [7]. This method, however, necessitates prior information, as any change in the target or environment would result in a significant drop in performance. It is also vulnerable to the frog-boiling attacks, which consist of the gradual injection of false data with no abrupt changes.

On the other hand, the group verification category requires multiple devices, aircraft, or ground stations, to compare the broadcasted location [8]. Those devices can also calculate the location using Time Difference of Arrival (TDoA), Frequency Difference of Arrival (FDoA), or Angle of Arrival (AoA) and compare it with the demodulated location [9]. However, this category suffers from certain limits since it is affected by the airspace density and signal delays, in addition to the problem of multipath signal propagation that causes errors in the computed TDoA. Also, the high loss of ADS-B packets and a large number of required receivers limit the feasibility of implementing this method.

Physical layer information leverages the unique physical characteristics of the channel in order to detect attacks [10] [11]. However, attackers can mimic these features and deceive the detection system with sufficient knowledge. In the data fusing technique, the primary surveillance radar (PSR), secondary surveillance radar (SSR), and wide-area multilateration (WAM)

978-1-6654-8009-3/22/$31.00 ©2022 IEEE

are utilized to confirm the authenticity of the received ADS-B data. The discrepancy in precision and sampling frequency between both systems is still a challenge for this approach. Finally, machine learning algorithms have also been proposed to detect cyberattacks.

In addition to the constraints mentioned above, UAVs are subject to the size, weight, and power (SWaP) limitations, which restrict processing capacity and thus the computational complexity of the employed detection techniques.

In this paper, we compare the performance of three machine learning algorithms, linear Support Vector Machine (SVM), c-SVM, and nu-SVM. This performance was performed using the following evaluation metrics: probabilities of detection, midsection, false alarm, and accuracy in addition to the time of training, time of detection, memory consumption of training and of detection.

The main contributions of this paper are:

- Generation of three different ADS-B message injection attacks based on real data.
- Identification of the most significant features for detecting ADS-B message injection attacks.
- Performance comparison of SVM models using specific metrics.

In the following sections, we explain the methodologies used in this work. Then, we present the research results and conduct a comprehensive discussion. At last, we end with a general conclusion.

II. Methodology

In this section, we discuss the acquisition of raw ADS-B data and the simulation of injection attacks. Then, we briefly describe all key steps in developing a machine learning model, namely data preprocessing techniques, feature extraction and selection, machine learning models, hyperparameters tuning, and evaluation metrics. The pipeline of the injection detection process is illustrated in Fig.1.

A. *OpenSky Raw Data*

ADS-B messages are acquired from the OpenSky network [12] which makes air traffic data available to the general public, notably ADS-B communications. Then, we reduced the volume of messages to a one-hour period. In order to minimize the traveled distance while still gathering the most samples, we selected a radius of 50 km around John F. Kennedy International Airport in New York City due to the large amount of air traffic data within that range.

ADS-B signals are broadcast by the airplane transponder and received by all nearby receivers within the transmission range. Since the radius in our dataset is relatively narrow, every aircraft within it will be able to receive ADS-B messages. In our study, we selected one aircraft as the receiver and potential target for the conducted injection attacks. Fig. 2 shows the aircraft location in the received ADS-B messages at an instant of time, while Table I describes the data fields of the dataset.

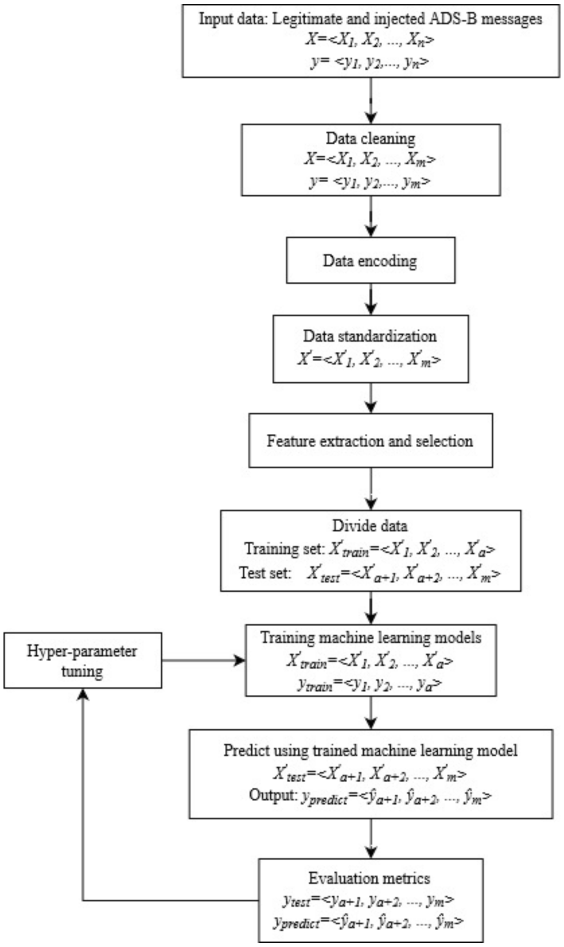
Input data: Legitimate and injected ADS-B messages
X=<X_1, X_2, ..., X_n>
y= <y_1, y_2,..., y_n>
Data cleaning
X=<X_1, X_2, ..., X_m>
y= <y_1, y_2,..., y_m>
Data encoding
Data standardization
X'=<X'_1, X'_2, ..., X'_m>
Feature extraction and selection
Divide data
Training set: X'_{train}=<X'_1, X'_2, ..., X'_a>
Test set: X'_{test}=<X'_{a+1}, X'_{a+2}, ..., X'_m>
Hyper-parameter tuning
Training machine learning models
X'_{train}=<X'_1, X'_2, ..., X'_a>
y_{train}=<y_1, y_2, ..., y_a>
Predict using trained machine learning model
X'_{test}=<X'_{a+1}, X'_{a+2}, ..., X'_m>
Output: $y_{predict}$=<$\hat{y}_{a+1}$, $\hat{y}_{a+2}$, ..., $\hat{y}_m$>
Evaluation metrics
y_{test}=<y_{a+1}, y_{a+2}, ..., y_m>
$y_{predict}$=<$\hat{y}_{a+1}$, $\hat{y}_{a+2}$, ..., $\hat{y}_m$>

Fig. 1. Process of building machine learning models

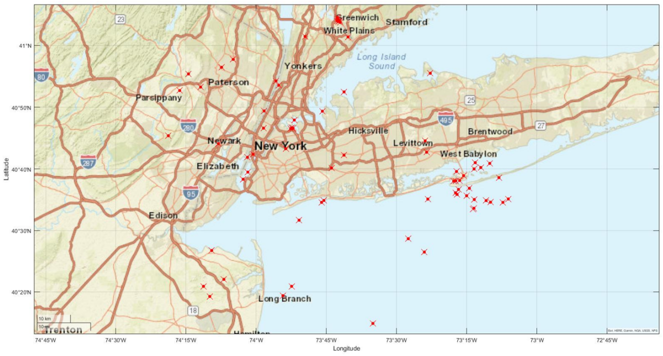

Fig. 2. Example of aircraft positions in the received ADS-B messages

B. *ADS-B Data Injection Attacks*

Due to the restrictions of broadcasting fraudulent ADS-B messages, our injected ADS-B messages were simulated based on the original messages. There are three forms of ADS-B message injection attacks that were simulated: path modification, ghost aircraft injection, and velocity drift. In the path modification attack, some segments of the traveled itinerary are modified by shifting the heading of the aircraft while taking into consideration the physical constraints (consistency

TABLE I. Data Fields

Parameters	Type/Values	Description
time	Integer/ No null values	Time in Unix timestamp (seconds) since the last position report.
icao24	string/ No null values	The International Civil Aviation Organization (ICAO)24 unique address of the transponder in hex string representation (24-bit).
lat	Float/ Can be null	Latitude in decimal degrees (WGS84 coordinates).
lon	Float/ Can be null	Longitude in decimal degrees (WGS84 coordinates).
velocity	Float/ Can be null	Velocity over ground in m/s.
callsign	String/ Can be null	Callsign of the vehicle.
onground	Boolean/ No null values	Its value is true if the aircraft is on ground otherwise it is false.
spi	Boolean/ No null values	Special Purpose Indicator (SPI) pulse is used by air traffic controllers to confirm the identity of certain aircraft.
squawk	Integer/ Can be null	Transponder code used for identification and emergencies.
baroaltitude	Integer/ Can be null	Barometric altitude in meters.
geoaltitude	Integer/ Can be null	Geometric altitude in meters.
RSS	Float/ Can be null	Strength of the signal at the ADS-B receiver's antenna in dB. (calculated after the attack simulation)
Doppler Shift	Float/ Can be null	Change in frequency effect due to the transmitter and receiver movement in Hertz. (calculated after the attack simulation)
Label	Float/ Can be null	Indicates if the message is legitimate or injected.

of the traveled distance with the reported velocity). In the ghost injection attack, numerous fake aircraft are inserted in a small radius from the target. To construct the ADS-B messages sent from fake aircraft, data from previous routes are used. Finally, the velocity drift attacks consist of a gradual drift applied to the velocity of legitimate aircraft.

Our dataset consists of 22,315 instances with equally distributed two classes: 11,158 authentic messages and 11,157 attack messages.

TABLE II. Class Distribution

Class	Legitimate messages	Injected messages
Number of instances	11,158	11,157
Percentage	50.00%	50.00%

C. Data Preprocessing

Before feeding data to the machine learning model, it has to be presented in a proper format to obtain accurate results. If the data is not well preprocessed before using the machine learning model, the results might be misleading even with high accuracy levels. In the following sections, we will describe the applied data cleaning, encoding, and standardization techniques.

D. Data Cleaning

ADS-B communications have a high loss rate due to message collisions and the distance between aircraft, resulting in a large number of null values in the dataset, which affects model learning. In addition, as a result of the multipath effect [13], the number of duplicate messages may arise. Therefore, we removed all null cells and duplicated rows from the dataset. While inspecting our dataset, we notice a large number of null values in some features more than others. Moreover, some features (i.e., squawk, spi, and callsign) do not contribute to the detection of injections attacks; therefore, they are removed from the dataset.

E. Data Encoding

The majority of ML models process the dataset features numerically. Since some ADS-B attributes are non-numerical, we must convert them appropriately so that the models can extract the necessary information. ADS-B features that need data encoding are icao24 and onground. icao24 is the identifier of the aircraft represented in hexadecimal which can be converted to base-10. The onground feature is Boolean, which has two values, True and False, that can be encoded as 1 and 0, respectively.

F. Data Standardization

Feature standardization should be the first step before SVM machine learning models. These models limit the magnitude of the coefficients associated with each feature, which depends on the value of the feature. In our case, the features' values have varying ranges. Therefore, we standardize the data using Equation 1 in order for the model to handle the features equally.

$$X' = \frac{X - \mu}{\sigma} \tag{1}$$

Where X' is the standardized feature value, X is the initial feature , μ is the mean, and σ is the standard deviation.

G. Feature Extraction and Selection

1) Feature Extraction: To improve the dataset, two new physical parameters, Received Signal Strength (RSS) and Doppler shift, were included as characteristics. To calculate the RSS [14] and Doppler shift, the distance from the receiver in the normal and attack cases was calculated using the aircraft and attacker's locations, respectively, by applying the following formulas:

$$RSS = \frac{P_T G_T G_R \lambda^2}{(4\pi d)^2} \tag{2}$$

Where P_T is the transmission power, G_T is the transmitter antenna gain, G_R is the receiver antenna gain, λ is the wavelength, and d is the distance between the transmitter and receiver.

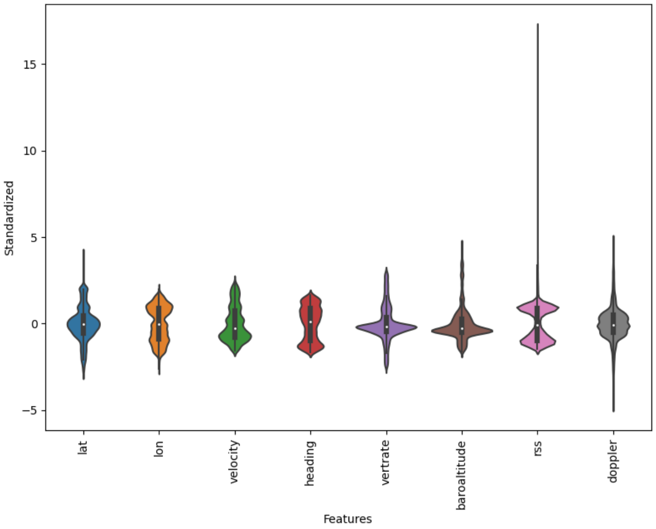

Fig. 3. Features after standardization

$$Doppler = f_R - f_0 \tag{3}$$

And

$$f_R = (\frac{c + v_R}{c + v_T}) \times f_0 \tag{4}$$

Where f_R is the frequency of the received signal, c is the speed of light, f_0 is the frequency of the transmitted signal, v_R is the receiver velocity, and v_T is the transmitter velocity.

2) Feature Selection: Feature selection is a critical step for machine learning models. This step assists us in identifying irrelevant features in the dataset and reducing computational complexity. In our work, we employed Kendall's Tau correlation to remove the correlated features.

a) Kendall's Tau Correlation: For a dataset with n observations, a pair of observations (x_i, y_i) and (x_j, y_j) is said to be concordant (n_c) if $x_i - x_j$ and $y_i - y_j$ have the same sign and discordant (n_d) if they have opposite signs. Kendall's Tau correlation is a powerful non-parametric analysis that determines the strength and direction of the relationship between two variables using the following formula [15]:

$$\tau = \frac{n_c - n_d}{n_c + n_d} = \frac{n_c - n_d}{n(n-1)/2} \tag{5}$$

Where τ achieves the maximum values of 1 when all pairs $n(n-1)/2$ are concordant and the minimum value -1 when all pairs are discordant.

H. Machine Learning Models

This paper compares the performance of three SVM models in detecting ADS-B injection attacks. SVM models are the most prevalent and popular machine learning models for classification and regression problems. These models can achieve high accuracy while preventing overfitting.

SVM [16] is an instance-based supervised ML model that can classify non-linear and linear data. This model is based on a complex algorithm that, unlike the other instance-based methods, uses a subset of training points, called support vectors, in the decision function, which highly optimizes the memory.

This model maps each data instance into an n-dimensional feature space and seeks the optimum hyperplane that divides the data into two classes with maximum marginal distance from both classes and minimum classification error.

The group of training instances used for the prediction process is selected using a kernel function. Linear SVM is a faster SVM implementation that employs a linear kernel by default. The basic optimization function of SVM is represented by the formula (6) [16].

$$min\frac{1}{2}||w||^2 + C\sum_{i=0}^{m} \xi_i \tag{6}$$

$$y_i(f(x_i)) \geq 1 - \xi_i \tag{7}$$

$$f(x) = (w^T\varphi(x) + b) \tag{8}$$

Where $f(x)$ is the decision function, w and b are its coefficients, φ is a non-linear function mapping the input features, $C(> 0)$ is the tradeoff between the distance of the separating margin and the training error, and ξ_i is the training error.

C-SVM [17] is a variant of SVM that aims to find the optimal margin for the support vectors to produce a better outcome. A parameter C was introduced to adjust this margin in order to balance the misclassification between the two datasets. Nu-SVM [18], like C-SVM, introduces a new hyperparameter, nu, to control the number of support vectors in the SVM basic algorithm. The fundamental difference between nu and C hyperparameters is that nu has a limited and smaller range of values.

I. Hyperparameter Tuning

Hyperparameters are configuration arguments that guide the learning process for a machine learning model for a specific dataset. Hyperparameter tuning or hyperparameter optimization is a technique used to select an optimal set of hyperparameters that achieve the best performance for an ML model and a given dataset. The optimization procedure begins with the definition of a search space containing various model configurations with different hyperparameter values.

The goal of the optimization is to find the best combination of hyperparameter values for the model's performance. A number of optimization methods have been proposed, including Random Search (RS) used in this work. Random Search is a simple and widely used hyperparameter tuning method that finds the optimal parameters configuration by randomly sampling ML configuration models in a bounded search space.

This method was specifically chosen due to its high-performance results with SVM models [19]. In this paper, the authors investigated the use of the Random Search method for adjusting SVM hyperparameters. They performed various experiments with different datasets and compared the results to other optimization meta-heuristics like genetic algorithms. The experiments show that the simple Random Search method leads to SVM models with predictive accuracy similar to the meta-heuristics results.

J. Evaluation Metrics

We selected eight evaluation metrics to assess the machine learning models for an onboard implementation on UAVs. They can be categorized into two types: detection efficiency and model performance, which can vary depending on the data size and the used model.

1) Detection Efficiency:

a) Probability of Detection (PoD): It denotes the probability of accurately classifying injected messages divided by the total number of injected messages.

b) Probability of Misdetection (PoM): It shows the proportion of injected messages that were categorized as genuine messages over the total number of injected messages.

c) Probability of False Alarm (PoFA): It gives the percentage of genuine messages that were incorrectly classified over the total number of legitimate messages.

d) Accuracy (Acc): It is defined as the percentage of correctly classified messages over the total number of messages.

2) Model Performance:

a) Time of Training (T_t): The machine learning model's execution time throughout the training phase.

b) Time of Detection (T_d): The elapsed time to detect the attacks.

c) Memory Usage in Training (Mem_t): The machine learning model's memory usage during the training period.

d) Memory Usage in Detection (Mem_d): The memory consumption during the detection phase.

III. Results and Discussion

In this study, 70% of the data was trained and the remaining 30% tested using a 10-fold cross-validation. After conducting the feature extraction and selection, the baroaltitude and the geoaltitude are found to be highly correlated, which is fairly expected. We selected the baroaltitude since it had fewer missing values and is considered to be more precise. Therefore, nine features that mainly reflect the status of the aircraft are used: latitude, longitude, baroaltitude, velocity, heading, vertical rate, onground, RSS, and Doppler shift.

TABLE III. BEST HYPER-PARAMETERS OF MODELS

ML Model	**Hyperparameters**
Linear SVM	Penalty = 'l2', loss = 'hinge', C = 10.
C-SVM	Kernel = 'rbf', gamma = 'auto', C = 100.
nu-SVM	Nu = 0.2, kernel = 'poly', gamma = 'auto', degree = 8.

The best hyper-parameters after applying the random search technique are given in Table III. Table IV shows the obtained results of the SVM models, while Figs. 4 and 5 illustrate the probability of detection and accuracy, and the probabilities of misdetection and false alarm, respectively.

TABLE IV. EVALUATION METRICS RESULTS

Model	PoD %	PoM %	PoFA %	Acc %	T_t (s)	T_d (s)	Mem_t (MiB)	Mem_d (MiB)
Linear SVM	82.45	17.55	7.51	87.05	**0.4532**	**0.0045**	1.3320	**0.0391**
C-SVM	**92.92**	**7.07**	**1.86**	**95.32**	3.2914	0.6242	**0.9844**	0.6211
nu-SVM	91.37	8.63	5.31	92.87	19.4205	6.2886	1.8164	0.6602

From Fig.4, we can observe that both C-SVM and nu-SVM show high results but C-SVM gives better results. These two models, C-SVM and nu-SVM, have an accuracy of 95.32% and 92.87%, respectively. On the other hand, linear-SVM has considerably low accuracy and probability of detection compared to the two other models. Fig.5 shows that C-SVM significantly outperforms the other models in terms of both probabilities of misdetection (7.07%) and false alarm (1.86%).

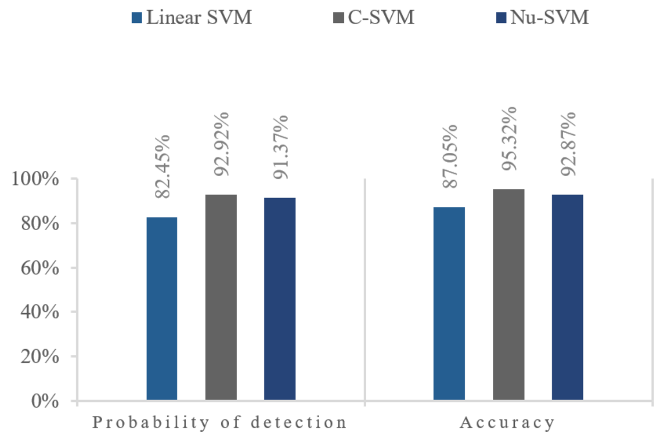

Fig. 4. Probability of detection and accuracy

As shown in Table IV, overall, all models have low memory consumption during the training and detection. However, nu-SVM is slow compared to the two other models; it takes this model 19.42s for the training phase and 6.29s for detection of attacks. Linear-SVM on the other hand is noticeably faster and lighter in both training (0.45s) and detection (0.0045s) phases. Meanwhile, C-SVM shows some reasonably good results in terms of the required training and detection times.

While choosing the best model, it is crucial to examine the costs of different metrics in the context of the situation at hand. In our case, a high probability of misdetection enables fraudulent messages to pass as genuine messages, but a high probability of false alarm may result in wrongly notifying the user of a possible attack only. In this case, the probability of misdetection is more important than the probability of false alarm.

Despite the expense of the detection time and memory, we would prefer to have higher accuracy and a lower misdetection probability than having a faster but less accurate detection model. This trade-off may be preferred since a relatively slower response is better than no or false response.

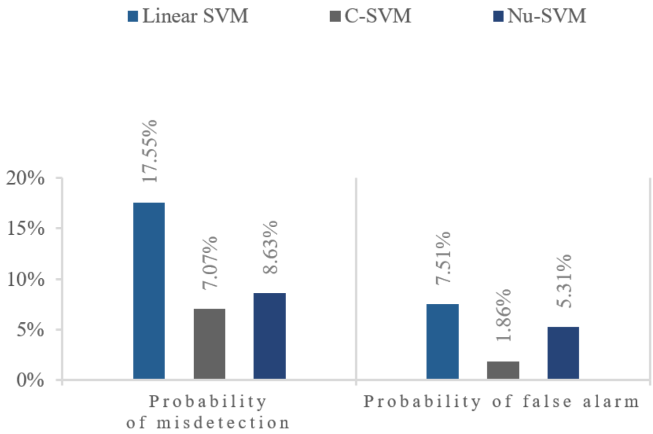

Fig. 5. Probabilities of misdetection and false alarm

Therefore, we can say that C-SVM is the best model for detecting ADS-B message injection attacks on UAVs. It has the highest accuracy and detection probability and the lowest misdetection and false alarm rate, while it maintains relatively good results in the time and memory metrics.

IV. Conclusion and Future Work

The U.S. Federal Aviation Administrations (FAA) and the European Aviation Safety Agency (EASA) mandates require all aircraft to be equipped with ADS-B systems; nevertheless, this technology lacks the fundamental security aspects. The aim of this work is to develop robust techniques to detect ADS-B message injection attacks. This paper compares the performance of three SVM models in detecting such attacks. A dataset was built using real ADS-B messages from the OpenSky network and messages resulting from three types of ADS-B message injection attacks that were simulated. The performance of the three SVM models was performed in terms of two types of evaluation metrics that are essential for UAVs, detection efficiency and model performance. The obtained findings indicate that C-SVM is the most suitable model for our application due to its high accuracy (95.32%) and fast attacks detection (0.62s).

Acknowledgment

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

References

[1] Federal Aviation Administration, “Automatic dependent surveillance-broadcast ADS-B out performance requirements to support air traffic control ATC service,” *Final Rule, 14 CFR Part 91*, vol. 75, no. 103, 2010.

[2] European Aviation Safety Agency, “Certification considerations for the enhanced ATS in non-radar areas using ADS-B surveillance (ADS-B-NRA) application via 1,090 MHz extended squitter,” *AMC*, vol. 20, no. 24, 2008.

[3] Federal Aviation Administration, “Automatic dependent surveillance-broadcast (ADS-B) flight inspection,” *National Policy*, vol. 8200.45, 2014.

[4] 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.

[5] M. R. Manesh and N. Kaabouch, “Analysis of vulnerabilities, attacks, countermeasures and overall risk of the automatic dependent surveillance-broadcast (ADS-B) system,” *International Journal of Critical Infrastructure Protection*, vol. 19, pp. 16–31, 2017.

[6] Federal Aviation Administration, “Drones by the numbers,” February 2022. [Online]. Available: https://www.faa.gov/uas/resources/by_the_numbers/

[7] J. A. Besada, G. de Miguel, A. M. Bernardos, and J. R. Casar, “Automatic-dependent surveillance–broadcast experimental deployment using system wide information management,” *International Journal of Microwave and Wireless Technologies*, vol. 4, no. 2, pp. 187–198, 2012.

[8] K. Sampigethaya and R. Poovendran, “Security and privacy of future aircraft wireless communications with offboard systems,” pp. 1–6, 2011.

[9] M. Monteiro, A. Barreto, T. Kacem, J. Carvalho, D. Wijesekera, and P. Costa, “Detecting malicious ADS-B broadcasts using wide area multilateration,” pp. 4A3–1, 2015.

[10] M. Strohmeier, V. Lenders, and I. Martinovic, “Intrusion detection for airborne communication using PHY-layer information,” pp. 67–77, 2015.

[11] M. Strohmeier and I. Martinovic, “On passive data link layer fingerprinting of aircraft transponders,” pp. 1–9, 2015.

[12] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, “Bringing up opensky: A large-scale ADS-B sensor network for research,” pp. 83–94, 2014.

[13] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, “Realities and challenges of nextgen air traffic management: the case of ADS-B,” *IEEE Communications Magazine*, vol. 52, no. 5, pp. 111–118, 2014.

[14] S. Kurt and B. Tavli, “Path-loss modeling for wireless sensor networks: A review of models and comparative evaluations.” *IEEE Antennas and Propagation Magazine*, vol. 59, no. 1, pp. 18–37, 2017.

[15] M. G. Kendall, “A new measure of rank correlation,” *Biometrika*, vol. 30, no. 1/2, pp. 81–93, 1938.

[16] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[17] C.-C. Chang and C.-J. Lin, “Training v-support vector regression: theory and algorithms,” *Neural computation*, vol. 14, no. 8, pp. 1959–1977, 2002.

[18] H. Chew, R. Bogner, and C. Lim, “Dual nu-support vector machine with error rate and training size biasing,” vol. 6, no. 1, pp. 4041–4041, 1999.

[19] R. G. Mantovani, A. L. Rossi, J. Vanschoren, B. Bischl, and A. C. De Carvalho, “Effectiveness of random search in SVM hyper-parameter tuning,” pp. 1–8, 2015.