Instance-based Supervised Machine Learning Models for Detecting GPS Spoofing Attacks on UAS

Ghilas Aissou, Selma Benouadah, Hassan El Alami, and Naima Kaabouch
School of Electrical Engineering and Computer Science, University of North Dakota
Grand Forks, ND 58202 USA

2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC) | 978-1-6654-8303-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCWC54503.2022.9720888

***Abstract*—Unmanned Aerial Systems (UAS) heavily depend on the Global Positioning System (GPS) for navigation. However, the unencrypted civilian GPS signals are subject to different types of threats, including GPS spoofing attacks. In this paper, we evaluate five instance-based learning models for GPS spoofing detection in UAS, namely K Nearest Neighbor, Radius Neighbor, Linear Support Vector Machine (SVM), C-SVM, and Nu-SVM. We used software-defined radio units to collect and extract features from satellite signals. Then, we simulated three types of GPS spoofing attacks specifically the simplistic, intermediate, and sophisticated attacks. The evaluation results show that Nu-SVM outperforms the other instance learning classifiers in terms of accuracy, probability of detection, probability of false alarm, and probability of misdetection. In addition, the model shows good computational performance regarding memory usage and processing time in the detection phase.**

***Index Terms*—UAS, GPS Spoofing attacks, Detection Technique, Machine Learning, Support Vector Machine, K-Nearest Neighbor.**

I. Introduction

Unmanned Aerial Systems (UAS) rely heavily on Global Positioning System (GPS) devices for navigation. However, the unencrypted nature of civilian GPS signals raises serious security concerns. GPS devices are vulnerable to various types of cyber-attacks, including GPS spoofing and mimicking, which are the most serious threat to UAS [1]. This attack can be used to hijack a UAS, steal the onboard technology and data stored inside; worse, the attacker may seek to crash the drone into populated areas, threatening human lives.

A number of research studies have been proposed to detect GPS spoofing attacks on UAS. For instance, the authors of [2] proposed a detection technique based on signal quality monitoring. It continuously checks for abnormal peaks in the received signals. These peaks result from overlapping spoofing signals with the authentic GPS signals. However, when spoofing signals and genuine signals are aligned, there are no significant spikes; hence the attacks cannot be detected, which is the case of synchronized spoofing attacks. The authors of [3] proposed a method that checks the signal power level since the received power of the broadcasted GPS signal is very low, and higher signal power may be a sign of a spoofing attack. This technique can easily detect high power signals, but cannot detect GPS spoofing signals of low power. In the same context, other approaches, such as those described in [4], monitor other information in the receiver signal, such as abrupt changes in the pseudo ranges measurement and time information known as time-of-week (TOW). These techniques are only reliable if the spoofer introduces a large time delay or position drift in the spoofing signal as in the case of simplistic spoofing attacks where the attacker is not aware of the actual position of the receiver. Other proposed detection methods require additional hardware, such as multiple antenna [5] or molecular cameras and vision sensors [6]. However, these techniques are not practical due to the size, weight, and power limitations of UAS.

Others papers focused on applying artificial intelligence methods to detect GPS spoofing attacks, including machine learning (ML) and deep learning. For example, the authors of [7] proposed a technique based on a neural network to detect spoofing and meaconing attacks and a Bayesian inference subsystem to evaluate the severity of the attack. In [8], the authors proposed a GPS replay attack detection system using an artificial neural network. They showed the impact of different features on the detection performance. The best performance was obtained using five features, satellite vehicle number, pseudo-range, carrier phase, Doppler shift, and signal-to-noise ratio. In [9], the authors used the C-Support Vector Machine (C-SVM) to detect GPS spoofing. The model was trained using an unbalanced simulation dataset to modify the receiver's clock drift and time derivative of the clock offset. The authors of [10] proposed a model that examines the received signal strength indicator and the arrival time of air traffic control messages regularly broadcasted by aerial vehicles. They adapted the K-Nearest Neighbors (KNN) classifier to estimate the location of the vehicle and the Extreme Gradient Boost model (XGBoost) for attack detection. In [11] and [12], the authors compared the performance of several ML algorithms in detecting GPS spoofing attacks. In [11], the authors conducted a K-fold analysis to select the best ML algorithm among Support Vector Machine (SVM), Random Forest (RF), decision trees, Naïve Bayesian, and Linear regression. Based on their results, SVM (with the polynomial kernel) outperforms the other methods. While in [12], the authors implemented Radial Basis Function and linear kernel SVM, Ada Boost, decision trees, Nearest Neighbors, and RF models. Their results show that the algorithms based on decision trees give better detection rates.

However, some of these techniques use large variations of GPS signal features, which makes them ineffective in the case of intelligent GPS spoofers capable of monitoring the physical properties of the GPS signals. In addition, most of the existing works do not consider the Size, Weight and Power constraints of UAS while designing and developing these methods. A

978-1-6654-8303-2/22/$31.00 ©2022 IEEE

TABLE I. Comparison of GPS spoofing detection techniques.

Existing technique	Approach	Features used	Power constraint consideration	Memory constraint consideration	Technique limitations
[2]	Signal quality monitoring	• Correlators amplitude	No	No	High misdetection rate
[3]	Received signal power level check	• Power level	No	No	Hardware modification High cost
[4]	Auxiliary peak tracking for duplicate correlation peaks and decoded information monitoring	• Duplicate correlation peaks • Time of week information • Almanac and Ephemeris Data	No	Yes	Undetected generated GPS signal
[5]	Signal discrimination based on the angle of arrival estimation using a multi-antenna scheme	• Angle of arrival	No	No	Hardware modification High cost
[6]	UAS position validation using relative positioning estimation using Visual Odometry	• Sub-trajectories of UAV	Yes	No	Hardware modification High cost
[7][8]	A supervised machine learning detection technique based on ANN	• Satellite vehicle number • Signal-to-noise ratio • Pseudo range • Doppler shift • Carrier phase shift	No	No	High processing time
[9]	Support vector machine algorithm	• Lock time • Carrier to noise ratio • Pseudorange	No	No	Dataset contains only time falsification attacks.
[10]	Use of machine learning to estimate the real position of the UAS using some characteristics of the air traffic messages	• Air traffic messages	Yes	No	Detection system is based on the assumption of available air traffic control messages
[11]	Performance comparison of several machine learning models including SVM, RF, DT, Naïve Bayes, and Linear Regression in detecting GPS spoofing attacks based on accuracy, precision, recall, and F-score	• Differences between signal period expressed in seconds or percentage • Differences between the signal peaks expressed in dB	No	No	Used small dataset
[12]	Evaluation of several machine learning models' performances according to the allowed noise limit	• Estimated residual noise	No	No	Assumes that the attacker is not able to null the authentic signal

comparison of someexisting techniques is summarized in Table I.

In previous work, the team investigated the performance of tree-based ML models, including, Random Forest, Gradient Boost, XGBoost and LightGBM in detecting sophisticated spoofing attacks [13]. In this paper, we compare the performance of five instance-based ML models, namely KNN, Radius Neighbor, SVM, C-SVM, and Nu-SVM in detecting different types of spoofing attacks, simplistic, intermediate, and sophisticated. We used a dataset consisting of 10,055 samples with 13 extracted features. The performance evaluation is conducted in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, processing time, memory size, and detection time per sample. To guarantee optimal results, we used a hyperparameter tuning technique to identify the best parameters for each model.

Concisely, the main contributions of this paper are:

- Real-time feature extraction from collected legitimate GPS signals at different GPS receiver stages.
- Assessment and simulation of three types of spoofing attacks according to attack signatures.
- Identification of relevant features for GPS spoofing detection using a feature selection technique.
- Performance investigation of instance-based models, namely Linear SVM, Nu-SVM, C-SVM, KNN, and Radius Neighbor in terms of accuracy, probability of detection, probability of misdetection, probability of false alarm, processing time, memory size, and detection time per sample.

The remainder of this paper is organized as follows: Section II discusses the materials and methods used in this work. Section III represents and discusses the results. The conclusion

is discussed in Section IV.

II. MATERIALS AND METHODS

In this section, we describe the experimental design used in collecting satellite GPS signals, the feature extraction process, and the GPS spoofing attack simulation. In addition, we present the data preprocessing techniques used to refine and filter the dataset for the Machine learning models.

Experiment Design

The GPS signals were acquired using a Universal Software Radio Peripheral (USRP), a front-end active GPS antenna, and an I5-4300U laptop with 8G RAM running Ubuntu 16.04.7 LTS version, as described in [13]. An open-source Global Navigation Satellite Systems Software-Defined Receiver was utilized in the implementation (GNSS-SDR).

GPS module working flow

As shown in Fig. 1, the GPS receiver architecture module consists essentially of four processing blocks: Signal conditioner, acquisition, and tracking blocks, observables block, and the position velocity and timing block. The signal conditioner is in charge of adjusting the data type to be used by the host computer. The acquisition block is responsible for declaring the presence of the GPS signal; it performs a three-dimensional search for a particular C/A code identified by the Pseudo Random Code (PRN), known as the ID of the satellite. Once the GPS signal is declared available by the acquisition block, a tracking loop block observes the changes in the signal's synchronization parameters including the code phase, the Doppler shift, and the carrier phase. This process is done independently in each of the eight channels. The synchronization data are then forwarded to the observables block to compute the pseudo-range and validate the carrier phase and the Doppler shift. Lastly, the PVT block calculates the position and determines the velocity using the multilateration technique.

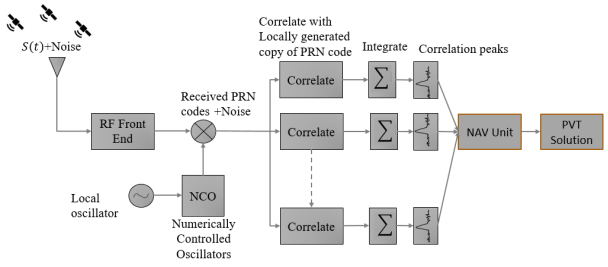

Fig. 1. Multi-channel GPS receiver architecture

A. Feature Extraction

As shown in Table II, thirteen features are extracted in each parallel channel from different blocks of the GPS receiver, starting from the pre-correlation stage going through the delay-locked loop and phase-locked loop correlation loops and ending with the post-correlation stage. The process starts by estimating the received signal's carrier to noise ratio ($C/N_\circ$). The correlator outputs are extracted at the same time that the Doppler shift is estimated since the output of the correlators widely varies in response to the satellite and receiver motion. The rest of the features are extracted at the observables block.

B. GPS Spoofing Attack Simulation

After collecting authentic GPS data, we used the MATLAB software to populate the dataset with three major types of spoofing attacks, simplistic spoofing attacks, intermediate spoofing attacks, and sophisticated spoofing attacks. We outlined the normal ranges of each feature extracted from the real GPS signals and used the signature of each type of attack to simulate each instance of the spoofed data. The normal data ranges are validated using two metrics, the standard deviation (STD) when the data is stationary (1) and the instantaneous rate of change (2).

$$STD = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \tag{1}$$

Where x_i is the value of the feature at sample i, μ is the mean, N is the total number of samples.

$$Instantaneous\ rate\ of\ change = \frac{x_{i+1} - x_i}{n_{i+1} - n_i} \tag{2}$$

Where $n_{i+1} - n_i$ is the distance between two samples which is equal to 1.

In a simplistic spoofing attack, the counterfeit signal is not synchronized with the authentic signal, resulting in an unaligned DO with the real signal. The PD is derived from DO estimation, hence the outrange value of the DO will affect the PD measurement too. Another attack signature present in simplistic attacks is the C/N_0 exceeding the threshold due to the spoofing signal sent at a higher power level than the authentic signal [14].

During an intermediate spoofing attack, the spoofer generates fake signals using a GPS signal simulator resulting in unremarkable DO variations. However, the CP accumulated cycles will be affected by the carrier signal generated by the GPS signal simulator. Another feature that the attacker manipulates in this type of attack is the navigation data such as TOW which will rise time error drift [4] [15].

In sophisticated attack, multiple synchronized transmitters are used to transmit jamming and spoofing signals at different arrival angles. Such an attack overpasses even multi-antenna-based detection techniques. However, observing distortions in the correlation loop can be a reliable solution to detect these attacks. Nevertheless, this technique is sometimes misleading in detecting multipath signals as a spoofing attack. In this work, we inserted distortions in the correlation peaks of multiple channels, emulating a real case spoofing attack using multiple transmitters [16].

C. Data Preprocessing

The generated dataset contains 10,055 samples that include 55% legitimate signals and 45% simulated attacks. Since the normal range of each feature differs from one another, we used the Min-Max technique for data normalization. This difference affects the performance of some learning models for example

instance-based models. Equation (3) represents the formula applied to rescale all the features values in a [0,1] range.

$$\hat{X}[:,i] = \frac{X[:,i] - Min(X[:,i])}{Max(X[:,i]) - Min(X[:,i])} \tag{3}$$

Where $\hat{X}$ is the normalized value of the feature, X is the original value, $Max(X[:,i])$ is the maximum weight of the feature while $Min(X[:,i])$ is its minimum weight.

Furthermore, to improve learning accuracy and decrease both overfitting and training time, we applied a feature selection technique namely the Spearman correlation method to remove correlated features of the GPS dataset. The Spearman correlation coefficient is known as a filter-based method., which measures the monotonic relationship between the features. This coefficient is defined as:

$$S(f_i, f_j) = 1 - 6\sum_{l} \frac{(f_i - f_j)^2}{N(N^2-1)} \tag{4}$$

Where S is the Spearman rank correlation coefficient, N is the length of the vector while f represents the rank of the feature, i is used for feature ranking. The relationship of the two features is scored between $\pm$ 1. When the result is around 1 or -1, the features have a positive or negative high correlation, respectively. In general, ML algorithms are based on the assumption that the data does not follow time-series distribution. With this assumption, most ML models expect a static relationship in data distribution. Several parameters are constant in stationary data, like mean, median, and variance. In our dataset, we found the TOW information and CP following a non-stationary relationship that can affect the performance of the proposed ML models [18]. For this reason, we used the differentiation method given by:

$$\Delta x_i = x_{i+1} - x_i \tag{5}$$

Where Δx_i is the first order differentiation, x_i is a data sample in the non-stationary data.

D. ML Models

In this work, we applied supervised learning models, known for their high performance in anomaly detection, in order to detect spoofing attacks. In addition to their ability to overcome the computational power limitation in UAS, the models can be trained in a non-restricted power consumption computer first and then uploaded to the UAS for detection which further reduces the processing power.

SVM, known for its capabilities to handle a large number of features, shows a great benefit in fault detection and fault classification. According to [19], SVM-based classifiers also minimize the misclassification risk compared to other traditional classifiers. In this work, three SVM versions are chosen, the classical C-SVM, the linear-SVM, and the Nu-SVM. The Nu-SVM model is quite similar to C-SVM but has the advantage of controlling the number of support vectors using the "nu" parameter, hence we may expect better performance results compared with the standard C-SVM.

In addition to the SVM models, the KNN and the Radius Neighbor, which is an extension of the latter, are implemented for GPS spoofing detection. These two models are called lazy learners. They have no training period and record all the instances in memory making them more appropriate in prediction problems with a limited number of features. This work compares the SVM-based classifiers, the C-SVM, Nu-SVM, linear SVM with KNN, and Radius Neighbor models.

ML models depend on several parameters, referred to as hyperparameters, that need to be well selected to achieve the best performance. Hyperparameter tuning can be time-consuming if done manually, especially when the ML model has many parameters [20]. In this work, we used the grid search technique to obtain the best results. This technique is an exhaustive search based on a defined subset of the tuning parameters.

III. Results and Discussion

In this work, we used four evaluation metrics to compare the selected ML models, the probability of detection, the probability of false alarm, the probability of misdetection, and accuracy. These metrics are defined as follows:

$$Probability\,of\,detection = \frac{T_p}{T_p + F_N} * 100 \tag{6}$$

$$Probability\,of\,false\,alarm = \frac{F_p}{T_N + F_p} * 100 \tag{7}$$

$$Probability\,of\,misdetection = \frac{F_N}{T_p + F_N} * 100 \tag{8}$$

$$Accuracy = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} * 100 \tag{9}$$

Where T_p is the number of correct predicted malicious flows, T_N is defined as the number of predicted normal flows, F_p presents the number of incorrect predicted malicious flows, and F_N shows the number of incorrect predicted normal flows. In addition, we used the processing time and memory usage to assess the performance of each model in both training and detection phase.

The results of the Spearman correlation coefficients show two pairs of features highly correlated with a score greater than 0.9, [RX, TWO] and [TCD, DO]. We deleted the RX time from the first pair and TCD from the second pair as explained in [13]. Furthermore, an additional data preprocessing step was performed to convert the two features, TOW and CP, that follow a non-stationary relationship into stationary distribution using the first-order differencing as discussed in [13].

The ML classification models are trained and tested with a labeled dataset of 10,055 samples with 55% of legitimated samples and 45% of simulated attacks. We used 10-fold cross-validation to train 70% of the dataset and test 30% of the remaining data. We customize the learning models according to the dataset using the Grid search technique for hyperparameter optimization. As illustrated in Table III, the SVM-based classifiers, the C-SVM, Nu-SVM, linear SVM, KNN, and Radius

TABLE II. Extracted Features

Feature name	Feature description	Extraction stage
Carrier to noise ratio in dB ($C/N_\circ$)	• The variation in the $C/N_\circ$ is mainly due to the input signal power. • During a spoofing attack, the attacker sends the fake signal to overcome the legitimate signal with a higher power level. Consequently, the value of $C/N_\circ$ will exceed the predefined normal ranges which is estimated to be in $\pm$ 2.8dB range [15].	Pre-correlation
Magnitude of the Prompt Correlator (PC)	• The local correlators equalize the Early (EC) and Late (LC) correlators in the tracking loops to identify the authentic GPS signal code phase. • PC will be at a half distance between the EC and LC correlations at t=0 [16].	During correlation
Magnitude of the Early Correlator (EC)	• EC is at 1⁄2 chip spacing before the prompt correlator.	During correlation
Magnitude of the Late Correlator (LC)	• LC is at 1⁄2 chip spacing after the prompt correlator.	During correlation
Prompt in phase correlator (PIP)	• PIP is the in-phase component of the prompt correlator amplitude.	During correlation
Prompt Quadrature component (PQP)	• PQP is the quadrature component of the prompt correlator amplitude. • PC can be expressed in terms of PIP and PQP components as $PC=\sqrt{PIP^2 + PQP^2}$.	During correlation
Carrier Phase Cycles (CP)	• CP is the beat frequency difference between the received carrier and a receiver-generated carrier replica. • CP is given in a number of accumulated cycles. • If the distance between the correlation peaks of the fake signal and the real signal is greater than 1.5 chips, the fake signal is considered to be severely delayed thus abnormal accumulation can be observed in the CP cycles [17].	Post-correlation
Time of the week in second (TOW)	• It is defined as the time of transmission of the navigation messages expressed in seconds [4]. • TOW is one of the key elements that an attacker can manipulate during a spoofing attack [14].	Post-correlation
Receiver Time (RX)	• It is defined as the receiver time of reception after the start of TOW.	Post-correlation
Pseudo-range in meter (PD)	• PD is the difference between the transmission and reception time which is expressed in meter representing the distance between the receiver and the satellite. • PD rate of change is deduced according to the working frequency of our GPS module. • Any abnormal changes out of the predefined range can be considered as a spoofing attack attempt.	Post-correlation
Carrier Doppler in HZ (DO)	• The Doppler effect or the Doppler shift is the change in frequency for a GPS receiver moving relative to its source. • The Doppler shift values are in a range of ±5kHz according to the configuration of the receiver, and the change rate is estimated to be equal to ±20Hz [14].	Post-correlation
Carrier Doppler in Tracking loop (TCD)	• It is the Doppler shift measured during the correlation stage. • TCD is used to define the upper and lower bounds of the accepted DO.	During correlation
Satellite vehicle number (PRN)	• It is the satellite identification number; the constellation requires a minimum of 24 operational satellites. The actual active number of satellites is 31.	Post-correlation

Neighbor models have specific parameter settings. The best parameters setting are introduced in the training process to carry out optimal learning performances.

Fig. 2, Fig. 3, and Table IV illustrate the results of the ML models for GPS spoofing attack detection. Fig. 2 shows the accuracy and the probability of detection results of the selected ML models. It can be observed that the Nu-SVM narrowly outperformed the C-SVM with an accuracy of 92.78% and a probability of detection of 91.26% because the Nu-SVM uses the "nu" parameter, which controls the number of support vectors. The Nu-SVM and C-SVM outperform the linear SVM and the distance-based models since they use 5_{th} degree polynomial kernels to resize the data into higher dimensional relationships. The linear SVM has the worst performance in terms of the two metrics as it is based on linear kernel function kernel which results in a rough estimation.

In terms of probability of misdetection and probability of false alarm, the C-SVM and Nu-SVM show good performance results, with a probability of misdetection of 11.13% and 8.73% respectively as illustrated in Fig. 3. This misdetection

TABLE III. Hyperparameters Tuning Results

Classifier	Best parameters
C-SVM	kernel='poly', degree=5, gamma=0.9, C=10
Nu-SVM	nu=0.1, kernel= 'poly', degree=5, gamma= 0.1, tol=$1e^{-7}$
Linear SVM	C= 11, class_weight='balanced', dual= True, fit_intercept= True, loss= 'hinge', max_iter= 600, penalty= 'l2', tol= $1e^{5}$
Radius Neighbor	algorithm= 'auto', leaf_size= 50, metric= 'manhattan', radius= 2.1318, weights= 'distance'
KNN	n_neighbors= 16, metric= 'manhattan', weights= 'distance'

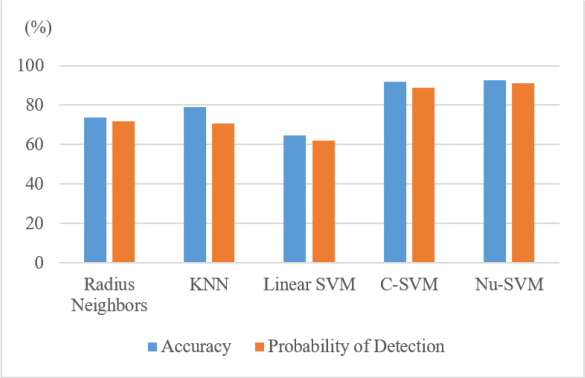

Fig. 2. Accuracy and probability of detection

rate is the result of undetected sophisticated attacks affecting slightly the correlator amplitude, which is naturally varying in response to satellite and receiver motion. Only a severe distortion in the correlators' function or a large quadrature accumulation shift is detected correctly as a result of a spoofing attack. However, the Radius Neighbor, Linear SVM, and KNN have very low-performance results with a probability of miss detection of 58.29%, 37.91%, and 29.24% respectively. This high misdetection probability means a failure of the system to identify the attacks especially the intermediate and sophisticated attacks.

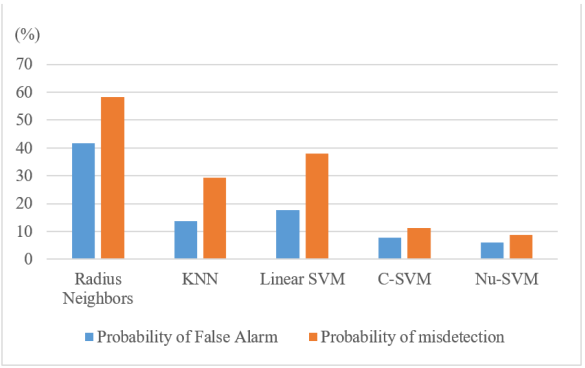

Fig. 3. Probability of misdetection and false alarm

As shown in Table V, SVM models require substantially longer training than distance-based models and this is because KNN and the Radius Neighbor are memory-based models known as lazy learners. These models store the training data

TABLE IV. Evaluation Metrics Results

	Radius Neighbors	KNN	Linear SVM	C-SVM	Nu-SVM
TPR	71.97%	70.75%	62.08%	88.86%	91.26%
FNR	58.29%	29.24%	37.91%	11.13%	8.73%
FPR	41.7%	13.68%	17.61%	7.7%	6.02%
ACC	73.81%	79.08%	64.56%	91.85%	92.78%

to be used during the detection stage without learning from it. As a result, distance-based models require a much longer time and larger memory size than SVM during the detection process. Hence, it can be noticed from the results that distance-based models are at least two times slower than SVM models. In contrast KNN and Radius Neighbor use 0.969 MiB and 2.434 MiB respectively during classification which is at least two times larger than the SVM models.

TABLE V. Performance Metrics Results

		Radius Neighbors	KNN	Linear SVM	C-SVM	Nu-SVM
Processing time (s)	Training	0.02	0.03	0.19	19.38	164.86
	Detection	2.7	0.22	0.04	0.16	0.12
Memory size (MiB)	Training	0.820	0.121	0.199	0.457	0.180
	Detection	2.434	0.969	0.043	0.523	0.508

In summary, Nu-SVM outperforms the other models in terms of evaluation metrics. Considering that the training of supervised learning models can be performed in a nonrestricted power and memory computational platform, we focus on the time and memory size requirement in the detection phase only, therefore Nu-SVM proves to be the most suitable model for UAS applications.

IV. Conclusion

In this work, we conducted a performance comparison of five instance-based learning models for GPS spoofing attack detection in terms of accuracy, probability of detection, probability of false detection, probability of false alarm, processing time, and memory size. We performed real-time feature extraction from the collected civilian GPS signals. Spoofing attacks were simulated using attack signatures found in the literature. Then, we selected the relevant features for each type of spoofing attack (simple, medium, and sophisticated spoofing attack). The simulation results show that the Nu-SVM outperforms the other learning models in terms of all evaluation metrics, with a probability of misdetection of 8.73%, which is a modest result in the cybersecurity domain. This misdetection rate is due to the sophisticated attacks being fully synchronized and tuned with authentic GPS signals. The Nu-SVM is also found to be faster and requires less memory during the detection phase than the other models.

Acknowledgment

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

REFERENCES

[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] 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. 66 428–66 441, 2018.

[3] M. L. Psiaki, T. E. Humphreys, and B. Stauffer, "Attackers can spoof navigation signals without our knowledge. here's how to fight back GPS lies," *IEEE Spectrum*, vol. 53, no. 8, pp. 26–53, 2016.

[4] A. Ranganathan, H. Ólafsdóttir, and S. Capkun, "Spree: A spoofing resistant GPS receiver," in *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, 2016, pp. 348–360.

[5] J. Magiera, "A multi-antenna scheme for early detection and mitigation of intermediate GNSS spoofing," *Sensors*, vol. 19, no. 10, p. 2411, 2019.

[6] M. Varshosaz, A. Afary, B. Mojaradi, M. Saadatseresht, and E. Ghanbari Parmehr, "Spoofing detection of civilian uavs using visual odometry," *ISPRS International Journal of Geo-Information*, vol. 9, no. 1, p. 6, 2020.

[7] N. Kaabouch, M. R. Manesh, and J. R. Kenney, "Detection of spoofing and meaconing for geolocation positioning system signals," Jul. 16 2020, US Patent App. 16/367,961.

[8] M. R. Manesh, J. Kenney, W. C. Hu, V. K. Devabhaktuni, and N. Kaabouch, "Detection of GPS spoofing attacks on unmanned aerial systems," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–6.

[9] 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 *2019 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2019, pp. 1–6.

[10] G. Liu, R. Zhang, C. Wang, and L. Liu, "Synchronization-free GPS spoofing detection with crowdsourced air traffic control data," in *2019 20th IEEE International Conference on Mobile Data Management (MDM)*. IEEE, 2019, pp. 260–268.

[11] A. Shafique, A. Mehmood, and M. Elhadef, "Detecting signal spoofing attack in uavs using machine learning models," *IEEE Access*, vol. 9, pp. 93 803–93 815, 2021.

[12] F. Gallardo and A. P. Yuste, "Scer spoofing attacks on the galileo open service and machine learning techniques for end-user protection," *IEEE Access*, vol. 8, pp. 85 515–85 532, 2020.

[13] G. Aissou, H. Ould Slimane, S. Benouadah, and N. Kaabouch, "Tree-based supervised machine learning models for detecting GPS spoofing attacks on uas," in *IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*. IEEE, 2021.

[14] A. Jovanovic, C. Botteron, and P.-A. Fariné, "Multi-test detection and protection algorithm against spoofing attacks on GNSS receivers," in *Proceedings of IEEE/ION PLANS 2014*, 2014, pp. 1258–1271.

[15] C. J. Wullems, "A spoofing detection method for civilian L1 GPS and the E1-B galileo safety of life service," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 4, pp. 2849–2864, 2012.

[16] E. Shafiee, M. Mosavi, and M. Moazedi, "Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency gps receivers," *The Journal of Navigation*, vol. 71, no. 1, pp. 169–188, 2018.

[17] B. W. Parkinson and J. J. Spilker, *Global positioning system: theory and applications*. Aiaa, 1996, vol. 1.

[18] M. Sugiyama and M. Kawanabe, *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

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

[20] M. Tayebi and S. El Kafhali, "Hyperparameter optimization using genetic algorithms to detect frauds transactions," in *The International Conference on Artificial Intelligence and Computer Vision*. Springer, 2021, pp. 288–297.