Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Machine Learning Modeling of GPS Features with Applications to UAV Location Spoofing Detection and Classification

Mohammad Nayfeh[1], Yuchen Li[1], Khair Al Shamaileh[1,*], Vijay Devabhaktuni[2], Naima Kaabouch[3]

[1] *Department of Electrical and Computer Engineering, Purdue University Northwest, Hammond, IN 46375, USA*
[2] *Department of Electrical and Computer Engineering, The University of Maine, Orono, ME 04469, USA*
[3] *School of Electrical Engineering and Computer Science, The University of North Dakota, Grand Forks, ND 58202, USA*

ARTICLE INFO

Article history:
Received 8 September 2022
Revised 15 December 2022
Accepted 27 December 2022
Available online 28 December 2022

Keywords:
Global positioning system (GPS)
machine learning (ML)
software-defined radio (SDR)
spoofing detection
unmanned aerial vehicles (UAVs)

ABSTRACT

In this paper, machine learning (ML) modeling is proposed for the detection and classification of global positioning system (GPS) spoofing in unmanned aerial vehicles (UAVs). Three testing scenarios are implemented in an outdoor yet controlled setup to investigate static and dynamic attacks. In these scenarios, authentic sets of GPS signal features are collected, followed by other sets obtained while the UAV is under spoofing attacks launched with a software-defined radio (SDR) transceiver module. All sets are standardized, analyzed for correlation, and reduced according to feature importance prior to their exploitation in training, validating, and testing different multiclass ML classifiers. The resulting performance evaluation of these classifiers shows a detection rate (DR), misdetection rate (MDR), and false alarm rate (FAR) better than 92%, 13%, and 4%, respectively, together with a sub-millisecond detection time. Hence, the proposed modeling facilitates accurate real-time GPS spoofing detection and classification for UAV applications.

© 2022 Elsevier Ltd. All rights reserved.

1. Introduction

The use of unmanned aerial vehicles (UAVs) has increased in the past few years in many applications, such as remote sensing, agriculture, search and rescue missions, 3D mapping, costal engineering, and disaster management (Shakhatreh et al., 2019, Naidoo et al., 2011, Nex and Remondino, 2014, Radoglou-Grammatikis et al., 2020, Drummond et al., 2015, Erdelj and Natalizio, 2016). To this end, the UAV market has grown significantly and is expected to reach USD 100B by 2030 (Unmanned Aerial Vehicle (UAV) Drones Market Size 2022-2030). This growth is attributed to the rapid development in emerging navigation and control technologies that enable swarm networking and collision avoidance protocols (Shakhatreh et al., 2019). However, UAV-specific solutions for cybersecurity vulnerabilities are not adequately addressed. These vulnerabilities potentially lead to damaging private properties and infrastructure, not to mention threatening public safety as recently reported in many incidents worldwide (Electromagnetic Interference Behind Darling Harbour Drone Crash, SkyJack Software Finds and Hijacks Drones, HK$1 million in damage caused by GPS jamming that caused 46 drones to plummet during Hong Kong show, Drone Crash Due To GPS Interference, Drones crash during light display at lantern festival). Therefore, it is paramount to develop detection and mitigation methodologies for the different types of cyberattacks against UAVs. Machine learning (ML) modeling for real-time detection and classification of global positioning system (GPS) spoofing is of a special interest to this work. GPS spoofing mitigation, on the other hand, is addressed with other techniques including inter-vehicle ranging and data sharing, inconsistency evaluations of GPS statistical properties, spatial processing of the GPS angle of arrival, signal strength and noise floor computations, null steering and beam forming, adaptive filtering, and time correlation estimation of the received GPS signals (Carson et al., 2016, Haider and Khalid, 2016, Jahromi et al., 2012, Ahmad et al., 2019, Sathaye and Ranganathan, 2020, Lee et al., 2020).

GPS spoofing attack affects the location awareness of a UAV by broadcasting a fake GPS signal that outpowers authentic transmissions to enforce a different position that serves the adversary interest (da Silva, 2017). Various detection solutions to this attack were proposed, such as evaluating the autocorrelation of the received signals or calculating position tolerances via the GPS and inertial measurement unit (IMU) modules (Khan et al., 2020, Wang et al., 2020). However, these methods introduce a high computational complexity and require high-precision sensors. Other solutions utilized antenna arrays for enabling a multi-stage spatial processing of the received signal to calculate the correlation in posi-

[*] Corresponding author
E-mail address: kalshama@pnw.edu (K.A. Shamaileh).

https://doi.org/10.1016/j.cose.2022.103085
0167-4048/© 2022 Elsevier Ltd. All rights reserved.

tion error (Broumandan and Curran, 2017, Rothmaier et al., 2021, Jansen et al., 2016). Nevertheless, these solutions introduce additional hardware components and may not be suitable for UAV applications. A vision-based solution that exploits a camera and the IMU module to measure the UAV velocity and compare it to that obtained by the GPS receiver was proposed in (Qiao et al., 2017). Although this solution has demonstrated an excellent performance in detecting spoofing, it imposes greater challenges to the onboard processor due to the sophisticated processing of images. Spoofing detection via analyzing the civilian and encrypted GPS signals were explored in (O'Hanlon et al., 2013). However, this approach requires information from another reference GPS receiver outside the attack range, which can be impractical, particularly in the scenario where the location of the attacker is unknown. The use of the automatic gain control (AGC) unit in the GPS receiver as a solution for detecting spoofing attacks was studied in (Akos, 2012). This solution, however, is affected by the noise level at the GPS frequency band, the environmental conditions, and the quality of the AGC unit.

The aforementioned literature survey suggests the need for developing real-time UAV-specific GPS spoofing detection and classification techniques with reasonable computational resources and minimal to no modifications to the already-existing hardware. These techniques must consider testing setups that pair both simulations and measurements to represent realistic attack scenarios. Therefore. this work explores, and ultimately validates, a solution for GPS spoofing detection and classification utilizing ML. The solution presented herein differs from other reported solutions in the following facets:

I Unlike (Carson et al., 2016, Sathaye and Ranganathan, 2020, Broumandan and Curran, 2017, Jansen et al., 2016, Zhang et al., 2012, Wang et al., 2018), the proposed approach neither requires new sensors nor modifications to the existing hardware (e.g., receiver circuitry). Rather, it can be integrated with standard receivers and ubiquitous modules.

II The datasets used for training and validating the ML classifiers are collected from testing setups that characterize authentic and spoofed autonomous flights. In such setups, the flights are created with a commercial flight controller and software interface, and attacks are launched with a software-defined radio (SDR) module. On the contrary, other reported approaches based their solutions on either simulated attacks or by launching attacks in a confined lab environment that fails in capturing authentic GPS signals (Wang et al., 2020, Jansen et al., 2016, O'Hanlon et al., 2013, Zhang et al., 2012, Wang et al., 2018, Xue et al., 2020, Jiang et al., 2021, Dang et al., 2022).

III ML techniques for spoofing detection were discussed in (Xue et al., 2020, Jiang et al., 2021, Dang et al., 2022). However, these techniques are computationally expensive because of utilizing deep learning or image-based classification, leading to increased training and detection time. On the other hand, the classifiers developed in this work are trained and tested considering minimal use of computational resources for enabling real-time spoofing detection.

IV All collected datasets and trained classifiers are made accessible to the research community. The datasets convey features that are extracted from a commercial GPS module. As a result, these datasets and classifiers can be utilized in promoting the cybersecurity in other research fields, such as transportation services, autonomous vehicles, and robotics.

The remaining of this paper is organized as follows: Section 2 discusses the underlined experimental setup, which elaborates on the spoofing attacks and the collection of signal features. Section 3 discusses the classifiers development, which entails the processing of the collected datasets, details the ML training process, and provides a performance evaluation for the different detection and classification models. Finally, Section 4 concludes this study and provides insight into future work.

Table 1
Properties of each square in the experimental setup.

Side length a_i (m)	$a_1 = 20$	$a_2 = 35$	$a_3 = 50$	$a_4 = 70$
Altitude h (m)	10	5	3	5
Ground speed v (m/sec)	1.00	1.50	2.20	5.00
Spoofer antenna gain (dB)	72	76	80	85

2. Experimental Setup

The experimental setup for collecting signal features is illustrated in Figure 1. This setup consists of four square-like flightpaths, each with a square side length of a_1= 20 m, $a_2 = 35$ m, $a_3 = 50$ m, and $a_4 = 70$ m. All squares are centered at the adversary (i.e., spoofer) and are assigned with different UAV flight altitudes and velocities, as detailed in Table 1. The attacks are launched at an open-source UAV from COEX, which is equipped with a u-blox M8 GPS receiver and a PX4 flight controller that enables the logging of several GPS features during flight. Mission plans are created with QGroundControl software tool, which allows for monitoring and controlling the UAV. Attack files are created with gps-sdr-sim, which uses the satellite ephemeris data and the coordinates of the fake location to generate a file with the bit stream of the attack. It is noteworthy to point out that ephemeris data is available at (Daily GPS Broadcast Ephemeris Files). Finally, the attacks are launched via a universal software radio peripheral B-210 SDR from National Instruments, which is interfaced with GNURadio installed on Linux virtual machine. A safe zone is designated for the testing area to avoid disturbing other surrounding electronics. This is achieved by adjusting the SDR (i.e., spoofer) transmitter gain while observing the GPS reception with a handheld GPS receiver.

Two spoofing attack types are investigated in this work: static and dynamic. In the static attack featured in Figure 1(a), the adversary transmits a fake GPS signal with a single location that differs from the correct one. This attack spoofs the UAV by enforcing a lock to a fixed position even though the UAV is in motion. On the other hand, the dynamic attack is created by launching spoofed GPS transmissions with moving location coordinates. This attack enforces the UAV to travel a flightpath designed by the adversary. These attack types are considered for the square with $a_1 = 20$ m. The configuration of the static attack entails collecting authentic and spoofed GPS feature samples. The authentic samples are collected while the UAV is hovering at the center of each square side. Then, attacks are launched to create a fake location at each of the four square corners, followed by collecting samples for the same set of features. The configuration of the dynamic attack (i.e., Dynamic 1) illustrated in Figure 1(b) involves placing the UAV at one of the square corners. Authentic features are collected while the UAV is executing a predefined mission along the perimeter of the square. Then, a spoofed signal with the same flightpath information is launched, and samples for the same set of features are collected. After this attack is completed, another variation of the dynamic attack (i.e., Dynamic 2) is also launched, where the spoofed flightpath conveys a midflight deviation, as shown in Figure 1(c). This deviation causes the UAV to change (i.e., correct) its course during mission execution. The same setups for these attacks are repeated for squares $a_{2,3,4}$. For each square, the transmitter (i.e., spoofer) gain is selected such that the signal power is at the threshold of spoofing the onboard GPS receiver as summarized in Table 1. Also, the collection of feature samples is performed over multiple days to diversify the data samples with different satellite constellations. It is paramount to point out that the

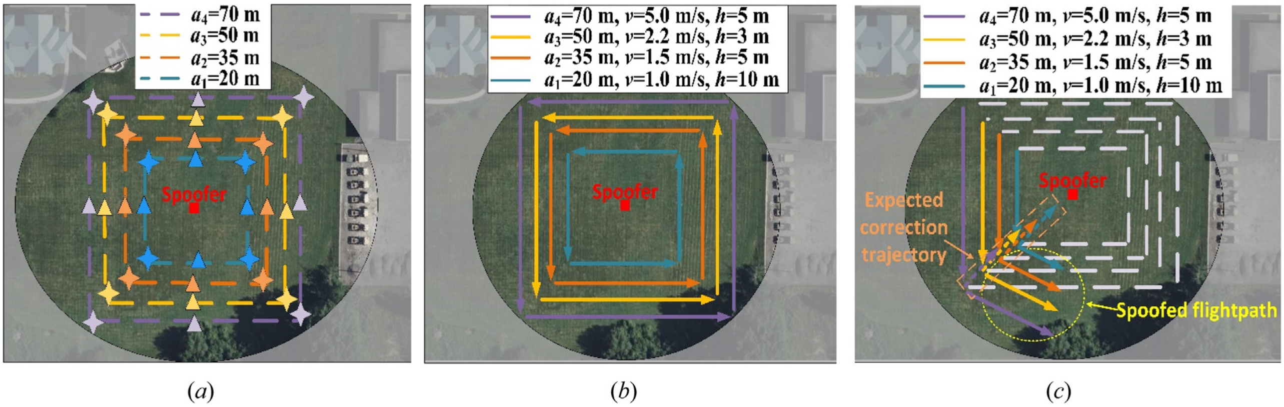
a_4=70 m
a_3=50 m
a_2=35 m
a_1=20 m
Spoofer
a_4=70 m, v=5.0 m/s, h=5 m
a_3=50 m, v=2.2 m/s, h=3 m
a_2=35 m, v=1.5 m/s, h=5 m
a_1=20 m, v=1.0 m/s, h=10 m
Spoofer
a_4=70 m, v=5.0 m/s, h=5 m
a_3=50 m, v=2.2 m/s, h=3 m
a_2=35 m, v=1.5 m/s, h=5 m
a_1=20 m, v=1.0 m/s, h=10 m
Spoofer
Expected correction trajectory
Spoofed flightpath
(a)
(b)
(c)

Figure 1. Experimental setup: (a) static attack. Triangles and stars represent the authentic and spoofed UAV positions, respectively, (b) dynamic attack 1 flightpaths, and (c) dynamic attack 2 flightpaths. Greyed area depicts the safe zone.

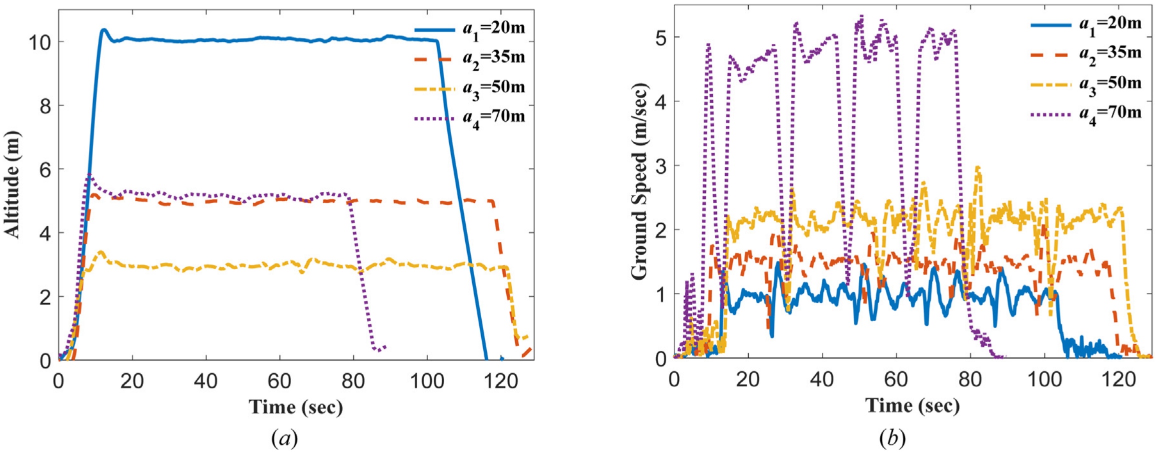

Figure 2. Logged authentic flightpaths of a single flight for each of the four squares: (*a*) altitude above ground level, and (*b*) ground speed. The datasets in (Resources) are for multiple flights with altitudes above sea level.

spoofed static location coordinates are fed as longitude, latitude, and altitude; whereas the spoofed dynamic coordinates are fed as user motion files obtained with SatGen 3 software tool that allows for generating National Marine Educators Association file with custom-velocity moving coordinates. It also enables timing the attack, which allows for choosing the exact date and time for attack initiation. Figure 2 shows the logged altitudes and ground speeds of the four squares during authentic flightpaths, which indicate small discrepancies compared to the predefined counterparts due to GPS measurement tolerances. Moreover, deceleration in ground speed in the case of square a_4 occurs at the corners to allow changing the yaw angle. Recorded videos demonstrating an example of each attack type (i.e., Static, Dynamic 1, Dynamic 2) are provided in (Resources). The extracted signal features from the drone's onboard GPS module during the experimental setup are given in Table 2. A total of 19166, 6923, 7503, and 3914 samples are extracted for Authentic, Static, Dynamic 1, and Dynamic 2 setups, respectively (i.e., 37,506 overall samples). Table 3 shows the resulting distribution of these samples for the four squares, which suggests a high degree of balance, leading to avoiding under- or oversampling. These collected samples can be found at (Resources).

3. Classifiers Development

After the successful extraction of GPS features and collection of data samples necessary for the detection and classification of the attacks, it is empirical that the resulting datasets be preprocessed prior to training and evaluating the classifiers. The preprocessing of the collected data is presented in Section 3.1, whereas the ML training and performance evaluation of the resulting classifiers are detailed in Sections 3.2 and 3.3, respectively.

3.1. Preprocessing of Collected Datasets

The processing of collected data is performed by eliminating the features with redundant values. These features are *fix_type* (value = 3), *jamming_state* (value = 0), *vel_ned_valid* (value = True), *timestamp_time_relative* (value = 0), *heading* (value = NaN), *heading_offset* (value = 0), and *selected* (value = 0). In addition, *timestamp* is eliminated since it stores the startup time of the system, and therefore is not specific to or affected by the attack. Then, the correlations between the remaining 19 features are calculated with the Spearman correlation algorithm, which assumes nonlinearity among features, and are presented in Figure 3 (Hauke and Kossowski, 2011). A correlation of $|c| > 0.8$ is used for considering a pair as highly correlated, leading to identifying (*alt, alt_ellipsoid*), (*eph, epv*), and (*vel, c_variance*) as correlated pairs. Once these pairs are specified, elimination based on feature importance is carried out. To this end, the relative importance of all features is computed according to their mean decrease in impurity as depicted in Figure 4. As a result, *alt_ellipsoid* and *eph* are discarded, leading to a dataset of 17 features. However, the *vel* feature is not eliminated from the feature set due to its contribution in improving the classification accuracy (i.e., misdetection rate) of the static and dynamic spoofing attacks. Finally, a standard scaling of the samples is carried out such that $x_{ij}' = (x_{ij} - \mu_j)/\sigma_j$, where

Table 2
Summary of the extracted features from the GPS module.

Extracted Feature	Short Description	Unit
timestamp	Time since system starts	μ-seconds
lat	Latitude in 1E−7	Degrees
lon	Longitude in 1E−7	Degrees
alt	Altitude in 1E−3 above sea level	Millimeters
alt_ellipsoid	Altitude in 1E−3 above ellipsoid	Millimeters
s_variance_m_s	GPS speed accuracy estimate	m/s
c_variance_rad	GPS course accuracy estimate	Radians
fix_type	The type of the GNSS fix	
	0-1: no fix	
	2: 2D fix	–
	3: 3D fix	–
	4: Radio Technical Commission for Maritime Services code differential	–
	5: Real-time kinematic, float	–
	6: Real-time kinematic, fixed	–
	8: Extrapolated	–
eph	GPS horizontal position accuracy	Meters
epv	GPS vertical position accuracy	Meters
hdop	Horizontal dilution of precision	–
vdop	Vertical dilution of precision	–
noise_per_ms	GPS noise per millisecond	dB
jamming_indicator	Indication of jamming occurrence	–
jamming_state	Indication of jamming detection by receiver	
	0: Unknown	–
	1: OK	–
	2: Warning	–
	3: Critical	–
vel_m_s	GPS ground speed	m/s
vel_n_m_s	GPS North velocity	m/s
vel_e_m_s	GPS East velocity	m/s
vel_d_m_s	GPS Down velocity	m/s
cog_rad	Course over ground (movement direction, not heading)	Radians
vel_ned_valid	True if north-east-down (NED) coordinates velocity is valid	–
timestamp_time_relative	Timestamp + timestamp_time_relative	μ-seconds
	Time of the UTC timestamp since system start	
time_utc_usec	UTC timestamp	μ-seconds
satellites_used	Number of satellites used	–
heading	Heading angle of XYZ body frame relative to NED	Radians
	NaN: Not available	
	Updated: Used for dual antenna GPS	
heading_offset	Heading offset of dual antenna array in body frame	Radian
	NaN : Not applicable	
	Value: $[-\pi, \pi]$	
selected	GPS device selection (if multiple receivers connected)	
	0: GPS1	–
	1: GPS2	–
	2: GPS3	–
	3: Blending multiple receivers	–

Table 3
Distribution of the collected data samples.

	Clean Samples	Attack Samples
Static		
20m	1703	1716
35m	1793	1771
50m	1744	1720
70m	1685	1716
Dynamic 1		
20m	1826	1835
35m	2120	1955
50m	1980	1890
70m	2000	1823
Dynamic 2		
20m	994	898
35m	1046	915
50m	1069	905
70m	1206	1196
Total Samples	19166	18340

$x_{ij}{}'$ is the scaled i^{th} sample of the j^{th} feature, and μ_j and σ_j are the mean and standard deviation of the sample values within the j^{th} feature, respectively.

3.2. ML Training Process

Once data processing is completed, two datasets are created from the collected samples. The first dataset is referred to as “Dataset 1: location-dependent” and conveys all 17 features for ML training and testing; whereas the second dataset is referred to as “Dataset 2: location-independent”, which excludes *lat, lon*, and *alt* (i.e., 14 features for ML training and testing). The overarching goal for each of these two datasets is to evaluate the performance difference that location-specific features impose. This differentiation also facilitates ML modeling for fixed-route applications (e.g., public transportations). Several three-class ML classifiers are trained, tested, and evaluated. These three classes are Clean, Static, and Dynamic, which denote no attack, the presence of a static attack, and the presence of a dynamic attack, respectively. The developed classifiers are random forest (RF), K-nearest neighbor (KNN), multilayer perceptron (MLP), logistic regression (LR), decision tree (DT), support vector machine (SVM), and naïve Bayes (NB).

Each of these classifiers represents a particular category in ML modeling, which are ensemble-based, instance-based, regularization-based, tree-based, neural network-based, and

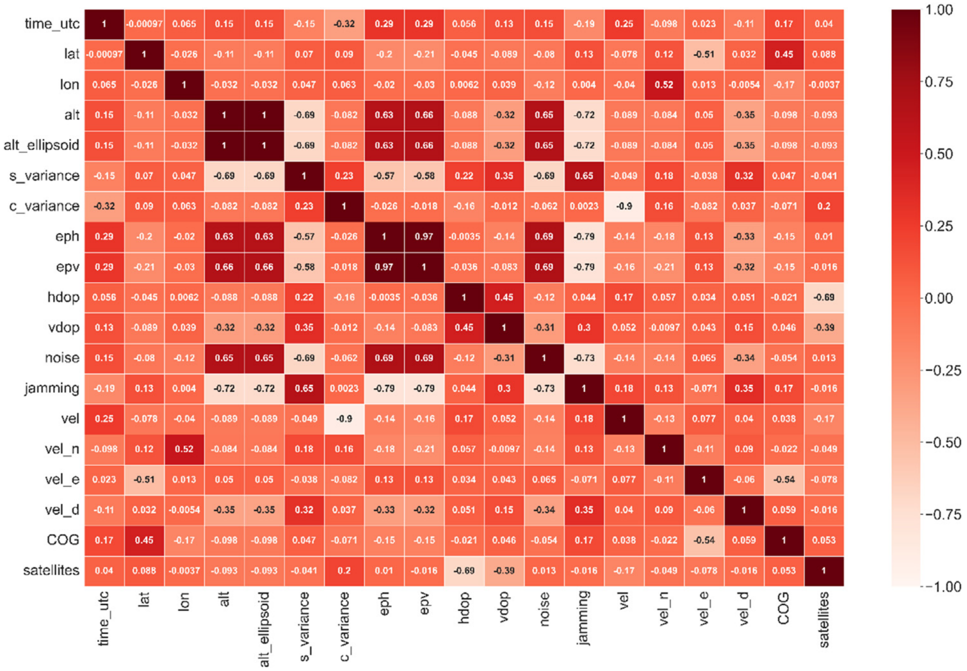

Figure 3. Resulting Spearman correlation of features according to the training and validation datasets in (Resources).

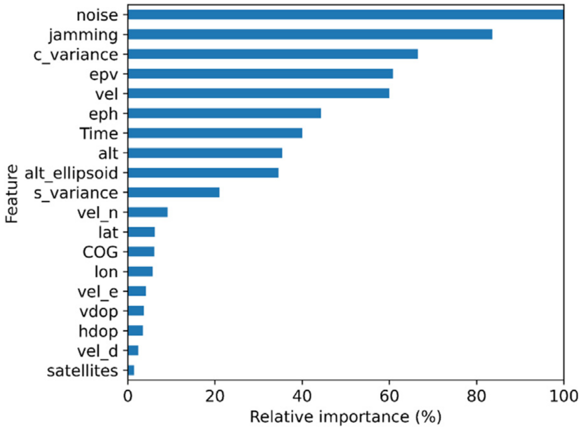

Figure 4. Relative importance of features according to the training and validation datasets in (Resources).

Bayesian-based modeling. The hyperparameters of all classifiers are optimized using randomized search algorithm, which facilitates the optimum configuration for each classifier model for a given dataset. Candidate hyperparameters are fed to the algorithm in form of user-defined range, and the resulting optimized hyperparameters are provided in Table 4. The classifiers are trained, validated, and tested with Datasets 1 and 2 considering their corresponding optimum hyperparameters.

3.3. Performance Evaluation of the Developed Classifiers

The following metrics are used for evaluating the performance of the adopted classifiers:

$$\text{DetectionRate(DR)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1} - \text{score(FS)} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

$$\text{FalseAlarmRate(FAR)} = \frac{FP}{FP + TN} \tag{5}$$

$$\text{MisdetectionRate(MDR)} = \frac{FN}{TP + FN} \tag{6}$$

In (1)-(6), *TP*, *TN*, *FP*, and *FN* represent the positive samples predicted as positive (i.e., true positive), negative samples predicted as negative (i.e., true negative), negative samples predicted as positive (i.e., false positive), and positive samples predicted as negative (i.e., false negative), respectively. The detection rate (DR) calculates the percentage of the correctly predicted samples in the dataset. The precision measures the classifier performance in classing negative samples as negatives and positive samples as positives. The recall measures the ability of the classifier to correctly predict all positive samples. The F1-score (FS) calculates the harmonic mean of the precision and recall. The false alarm rate (FAR) measures the probability of false detection. Finally, the misdetection rate (MDR) measures the probability of not detecting an attack.

Figure 5 summarizes the approach for detecting and classifying the underlined spoofing attacks, which entails preparing and launching the attacks, extracting GPS features, collecting and processing samples, and developing multiclass classifiers. Table 5 illustrates the resulting classifiers evaluation scores. Training, validation, and testing are performed on a 64-bit Windows 10 machine with AMD Ryzen 7 3700X CPU @ 3.6 GHz and 32 GB of DDR4-3600 MHz memory. The reported scores are averaged over ten independent runs, each of which the samples in the datasets are shuffled and split into 70% in training and 30% in validation (i.e., 10-fold cross validation). Testing is carried out by introducing the classifiers to subsets that are not used during the training and validation stages. To this end, the datasets from squares $a_{1,3,4}$ are exploited for training and validating the classifiers; whereas

Table 4
Optimized hyperparameters for each of the three-class ML classifiers.

Category	Classifier	Dataset 1: Location-dependent	Dataset 2: Location-independent
Ensemble	**RF**	Quality of split criterion: Entropy	Quality of split criterion: Entropy
		Maximum tree depth: 394	Maximum tree depth: 394
		Minimum number of samples at a leaf node: 39	Minimum number of samples at a leaf node: 39
		Minimum number of samples to split a node: 489	Minimum number of samples to split a node: 489
		Number of trees: 757	Number of trees: 757
		Cost-Complexity pruning parameter: 0.00114063	Cost-Complexity pruning parameter: 0.00114063
Instance	**KNN**	Leaf size: 728	Leaf size: 728
		Number of neighbors: 538	Number of neighbors: 755
		Weight function: Distance	Weight function: Distance
		Nearest neighbor computation algorithm: Ball Tree	Nearest neighbor computation algorithm: K-D Tree
		Distance metric: Manhattan	Distance metric: Manhattan
		Power parameter for the distance metric: 6	Power parameter for the distance metric: 6
	SVM	Norm used in penalty: L1	Norm used in penalty: L2
		Loss function: Squared Hinge	Loss function: Squared Hinge
		Dual optimization algorithm: False	Dual optimization algorithm: False
		Maximum number of iterations: 777	Maximum number of iterations: 1159
		Regularization parameter: 3.50658	Regularization parameter: 6.29736
Regularization	**LR**	Optimization algorithm: Stochastic avg. gradient descent	Optimization algorithm: Newton's method
		Norm used in penalty: L2	Norm used in penalty: None
		Regularization parameter: 7.210172	Regularization parameter: 4.4248
		Maximum number of iterations: 459	Maximum number of iterations: 904
Tree	**DT**	Quality of split criterion: entropy	Quality of split criterion: entropy
		Maximum tree depth: 394	Maximum tree depth: 394
		Minimum number of samples at a leaf node: 39	Minimum number of samples at a leaf node: 39
		Minimum number of samples to split a node: 489	Minimum number of samples to split a node: 489
		Node split strategy: Best	Node split strategy: Best
		Cost-Complexity pruning parameter: 0.00140636	Cost-Complexity pruning parameter: 0.3489472
Neural network	**MLP**	Optimization algorithm: Limited-memory Broyden–Fletcher–Goldfarb–Shanno	Optimization algorithm: Limited-memory Broyden–Fletcher–Goldfarb–Shanno
		Hidden layers and neurons: two with 221 & 170 neurons each	Hidden layers and neurons: one with 602 neurons
		Activation function: Logistic	Activation function: Logistic
		Maximum number of iterations: 954	Maximum number of iterations: 596
		L2 regularization term strength: 0.001761192	L2 regularization term strength: 0.533722
		Early stopping: True	Early stopping: True
Bayesian	**Gaussian NB**	Smoothing parameter for calculation stability: 0.433139	Smoothing parameter for calculation stability: 1.724043e-6

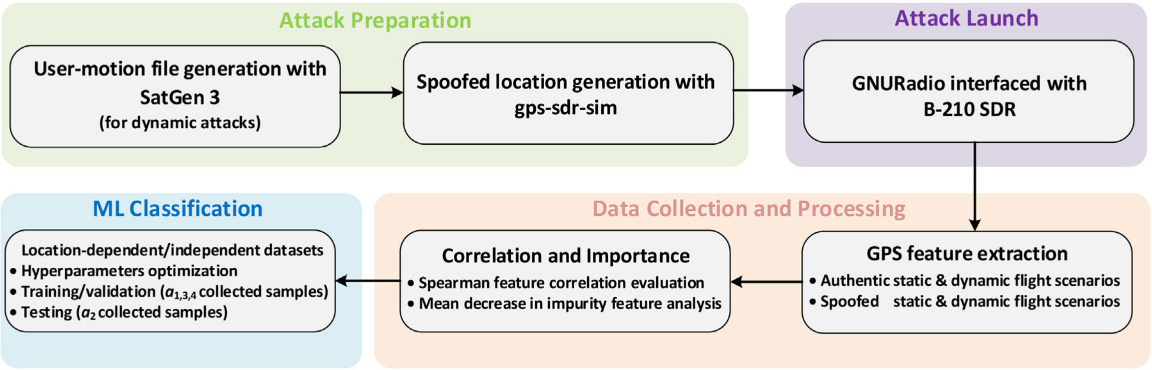
Attack Preparation
User-motion file generation with SatGen 3 (for dynamic attacks)
Spoofed location generation with gps-sdr-sim
Attack Launch
GNURadio interfaced with B-210 SDR
Data Collection and Processing
GPS feature extraction
• Authentic static & dynamic flight scenarios
• Spoofed static & dynamic flight scenarios
Correlation and Importance
• Spearman feature correlation evaluation
• Mean decrease in impurity feature analysis
ML Classification
Location-dependent/independent datasets
• Hyperparameters optimization
• Training/validation ($a_{1,3,4}$ collected samples)
• Testing (a_2 collected samples)

Figure 5. Flowgraph summarizing the development of the classifiers for GPS spoofing attack detection and classification.

Table 5
Metrics for the three-class GPS spoofing detection and classification models (TT: training time, PT: prediction time).

	Location-dependent							Location-independent						
Model	**DR** (%)	**Precision** (%)	**FS**	**FAR** (%)	**MDR** (%)	**TT** (ms)	**PT** (ms)	**DR** (%)	**Precision** (%)	**FS**	**FAR** (%)	**MDR** (%)	**TT** (ms)	**PT** (ms)
RF	90.89	90.04	0.90	4.29	15.90	7445.01	154.77	89.47	90.23	0.89	4.96	17.99	5659.51	159.38
KNN	87.99	90.53	0.86	5.96	21.42	4.95	3752.68	84.68	88.27	0.81	7.39	27.35	3.31	3568.57
MLP	89.64	90.24	0.89	4.83	17.54	6413.49	32.72	88.77	90.73	0.87	5.32	19.85	31207.65	44.07
LR	90.53	91.13	0.90	4.42	16.09	1768.15	0.26	**96.67**	**96.77**	**0.97**	**1.59**	**4.26**	**653.57**	**0.22**
DT	**92.36**	**93.95**	**0.92**	**3.70**	**12.94**	**35.94**	**0.23**	95.32	94.38	0.95	2.18	6.73	26.34	0.21
SVM	89.84	90.66	0.89	4.84	17.37	925.33	0.29	88.44	89.25	0.88	5.40	17.97	50.99	0.26
NB	91.17	91.10	0.91	4.12	14.01	2.70	0.82	51.66	26.69	0.35	33.3	66.67	2.41	0.66

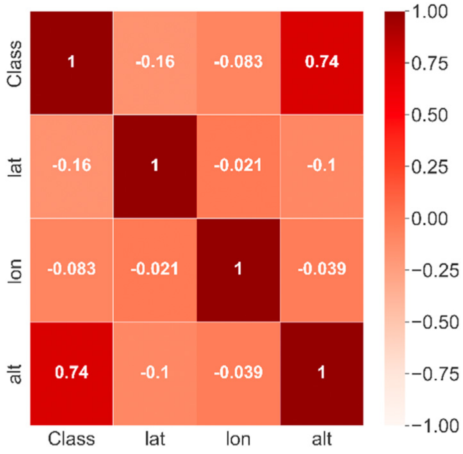

Figure 6. Degree of linear correlation between the location-dependent features and the class.

those from square a_2 are utilized for testing. Results show that the optimum classifier in classifying the location-dependent dataset is DT with a DR of 92.36%, FAR of 3.70%, and MDR of 12.94%. On the other hand, the optimum classifier in classifying the location-independent dataset is LR with a DR, FAR, and MDR of 96.67%, 1.59%, and 4.26%, respectively.

Table 5 also shows that the performance of the LR classifier is improved significantly in classifying the location-independent dataset in comparison to classifying the location-dependent dataset. This improvement is mainly due to eliminating the *lat* and *lon* features, which are not linearly correlated with the class (i.e., Clean, Static, Dynamic), as depicted in Figure 6. This finding is obtained after examining linear correlation between the location-specific features and the class with Pearson algorithm. In addition, Table 5 shows the average training time and prediction time for each classifier. It is noticed that DT and LR algorithms have the optimum prediction time of 0.23 ms and 0.22 ms, respectively. On the other hand, NB has the lowest training time of 2.70 ms considering the location-dependent dataset. This training time reduces to 2.41 ms in the case of eliminating the location-specific features from the dataset at the expense of model accuracy. The aforementioned prediction times account for all samples in the testing dataset (i.e., 9600 samples), leading to a 0.024 µs prediction time per sample. Hence, this prediction rate enables real-time detection and classification. Finally, although the features/samples are reduced in the location-independent dataset, MLP algorithm experiences an increase in training time due to using a different set of hyperparameters as compared to those used for the location-dependent scenario.

Figure 7 presents the confusion matrices of the optimum classifiers for the location-dependent and location-independent datasets. These matrices allow for evaluating the classifiers by illustrating the number of *TP*, *TN*, *FP*, and *FN* samples. For example, the DT confusion matrix shown in Fig. 7(a) indicates that 2,607 dynamic attack samples are correctly classified; while only 21 are misclassified as Clean and 242 are misclassified as Static attack (i.e., total number of samples in the testing dataset labeled as Dynamic is 2,870). Accordingly, both classifiers exhibit low misclassification between classes. The subroutines used for training, validating, and testing the classifier models and computing the corresponding evaluation metrics with the presence of the *vel* feature (i.e., Table 5) and without the presence of this feature can be found in (Resources).

4. Conclusion

In this work, a ML-based approach for real-time detection and classification of GPS spoofing attacks is presented. This approach entails developing different classifiers utilizing realistic datasets obtained from rigorous testing setups of authentic and spoofed flight scenarios. Such classifiers are evaluated with multiple metrics for two dataset types (i.e., with and without location information). DRs of 92.63% and 96.67% are achieved considering the location-dependent and location-independent datasets, respectively. These DRs are assumed to satisfy a multitude of applications since the flight controller used in this research registers five sample sets per second. Therefore, with these DRs, at least four of the five registered sets will be correctly detected and classified. This performance is intertwined with low MDR, FAR, and prediction time, enabling real-time detection and classification. The proposed approach does not require hardware modifications as it classifies spoofing attacks based on the measurements of the commercial GPS receivers irrespective of the UAV architecture, and the classifier routines can be hosted inside the onboard microprocessor, where detection and classification occurs. However, the proposed approach potentially introduces minimal software modifications to allow for forwarding GPS feature-related samples, which are typically stored in the flight controller, to the microprocessor in real-time. Future work includes exploring mitigation techniques as well

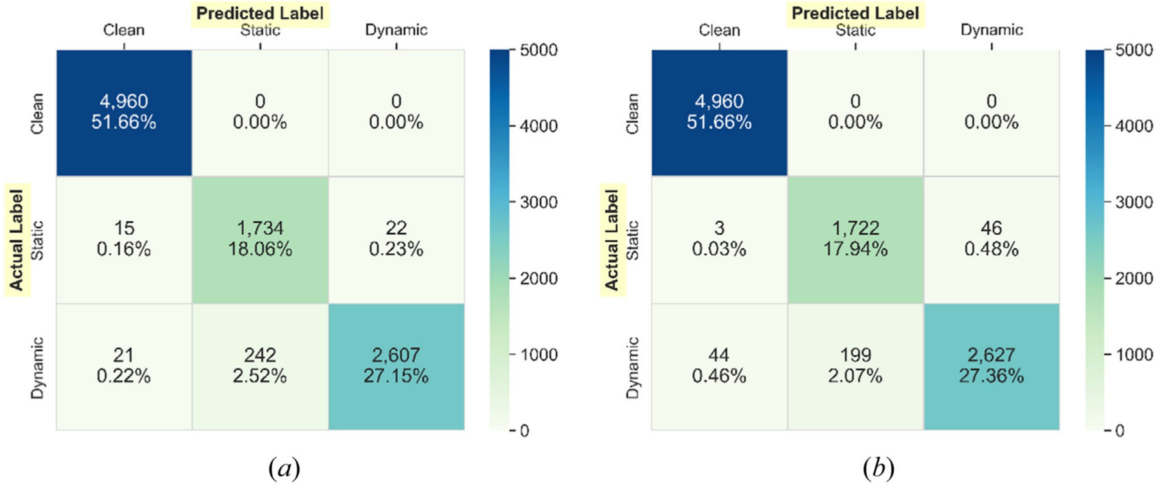

Figure 7. Confusion matrices of the optimum classifiers: (a) DT for location-dependent and (b) LR for location-independent datasets.

as investigating the detection and classification of more sophisticated GPS spoofing attacks.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

Khair Al Shamaileh reports financial support was provided by National Science Foundation.

CRediT authorship contribution statement

Mohammad Nayfeh: Methodology, Software, Investigation, Writing – original draft, Visualization. **Yuchen Li:** Investigation. **Khair Al Shamaileh:** Conceptualization, Writing – original draft, Supervision, Project administration, Funding acquisition. **Vijay Devabhaktuni:** Conceptualization, Funding acquisition. **Naima Kaabouch:** Project administration.

Data availability

https: //github.com/mnayfeh/gps_spoofing_detection

Acknowledgement

This work was supported by the National Science Foundation, Secure and Trustworthy Cyberspace Program, under Award 2006662.

References

Shakhatreh, H., et al., 2019. Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges. IEEE Access 7, 48572–48634.

Naidoo, Y., Stopforth, R., Bright, G., 2011. Development of an UAV for search & rescue applications. In: IEEE Africon '11, Victoria Falls, Zambia, pp. 1–6.

Nex, F., Remondino, F., 2014. UAV for 3D mapping applications: a review. Applied Geomatics 6, 1–15.

Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., Moscholios, I., 2020. A compilation of UAV applications for precision agriculture. Computer Networks 172, 107148.

Drummond, C., Harley, M., Turner, I., Matheen, A., Glamore, W., 2015. UAV applications to coastal engineering. Australasian Coasts & Ports Conference.

Erdelj, M., Natalizio, E., 2016. UAV-assisted disaster management: Applications and open issues. In: *2016 International Conference on Computing, Networking and Communications* (*ICNC*), HI, USA, pp. 1–5.

Unmanned Aerial Vehicle (UAV) Drones Market Size 2022-2030. [Online]. Available: https://www.precedenceresearch.com/unmanned-aerial-vehicle-drones-market

Electromagnetic Interference Behind Darling Harbour Drone Crash. [Online]. Available: https://australianaviation.com.au/2022/06/electromagnetic-interference-behind-darling-harbour-drone-crash/

SkyJack Software Finds and Hijacks Drones. [Online]. Available: https://www.pcmag.com/news/skyjack-software-finds-and-hijacks-drones

HK$1 million in damage caused by GPS jamming that caused 46 drones to plummet during Hong Kong show. [Online]. Available: https://www.scmp.com/news/hong-kong/law-and-crime/article/2170669/hk13-million-damage-caused-gps-jamming-caused-46-drones

Drone Crash Due To GPS Interference in U.K. Raises Safety Questions. [Online]. Available: https://www.forbes.com/sites/davidhambling/2020/08/10/investigation-finds-gps-interference-caused-uk-survey-drone-crash/?sh=57b389bd534a

Drones crash during light display at lantern festival. [Online]. Available: https://www.taipeitimes.com/News/taiwan/archives/2020/02/24/2003731529

Carson, N., Martin, S.M., Starling, J., Bevly, D.M., 2016. GPS spoofing detection and mitigation using cooperative adaptive cruise control system. In: *2016 IEEE Intelligent Vehicles Symposium* (*IV*), Gothenburg, Sweden, pp. 1091–1096.

Haider, Z., Khalid, S., 2016. Survey on effective GPS spoofing countermeasures. In: *2016 Sixth International Conference on Innovative Computing Technology* (*INTECH*), Dublin, Ireland, pp. 573–577.

Jahromi, A., Broumandan, A., Nielsen, J., Lachapelle, G., 2012. GPS spoofer countermeasure effectiveness based on signal strength, noise power, and C/N_0 measurements. International Journal of Satellite Communications and Networking 30 (4), 181–191.

Ahmad, M., Farid, M.A., Ahmed, S., Saeed, K., Asharf, M., Akhtar, U., 2019. Impact and Detection of GPS Spoofing and Countermeasures against Spoofing. In: *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies* (*iCoMET*), Sukkur, Pakistan, pp. 1–8.

H. Sathaye and A. Ranganathan, "SemperFi: a spoofer eliminating standalone GPS receiver," *13th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (*WiSec* '20), NY, USA, pp. 353–355, 2020.

Lee, J., Taha, A.F., Gatsis, N., Akopian, D., 2020. Tuning-Free, Low Memory Robust Estimator to Mitigate GPS Spoofing Attacks. IEEE Control Systems Letters 4 (1), 145–150.

da Silva, D., 2017. GPS jamming and spoofing using software defined radio. Department Of ISTA, University Institute of Lisbon.

Khan, A., Iqbal, N., Khan, A., Khan, M., Ahmad, A., 2020. Detection of intermediate spoofing attack on global navigation satellite system receiver through slope based metrics. The Journal of Navigation 73 (5), 1052–1068.

S. Wang, J. Wang, C. Su and X. Ma, "Intelligent detection algorithm against UAVs' GPS spoofing attack," *IEEE 26th International Conference on Parallel and Distributed Systems* (*ICPADS*), Hong Kong, China, pp. 382–389, 2020.

A. Broumandan, and J. Curran, "GNSS spoofing detection in covered spoofing attack using antenna array," *International Technical Symposium on Navigation and Timing* (*ITSNT*), Toulouse, France, pp. 1-9, 2017.

Rothmaier, F., Chen, Y., Lo, S., Walter, T., 2021. GNSS spoofing detection through spatial processing. Navigation 68 (2), 243–258.

Jansen, K., Tippenhauer, N.O., Pöpper, C., 2016. Multi-receiver GPS spoofing detection: error models and realization. In: *32nd Annual Conference on Computer Security Applications* (*ACSAC '16*), NY, USA, pp. 237–250.

Qiao, Y., Zhang, Y., Du, X., 2017. A vision-based GPS-spoofing detection method for Small UAVs. In: 13th *International Conference on Computational Intelligence and Security* (*CIS*), Hong Kong, China, pp. 312–316.

O'Hanlon, B., Psiaki, M., Bhatti, J., Shepard, D., Humphreys, T., 2013. Real-time GPS spoofing detection via correlation of encrypted signals. Navigation 60 (4), 267–278.

Akos, D., 2012. Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (AGC). Navigation 59 (4), 281–290.

Zhang, Z., Trinkle, M., Qian, L., Li, H., 2012. Quickest detection of GPS spoofing attack. In: MILCOM 2012 - 2012 IEEE Military Communications Conference, Orlando, FL, pp. 1–6.

Wang, Q., Lu, Z., Gao, M., Qu, G., 2018. Edge Computing based GPS Spoofing Detection Methods. In: *2018 IEEE 23rd International Conference on Digital Signal Processing* (*DSP*), Shanghai, China, pp. 1–5.

Xue, N., Niu, L., Hong, X., Li, Z., Hoffaeller, L., Pöpper, C., 2020. DeepSIM: GPS spoofing detection on UAVs using satellite imagery matching. In: *Annual Computer Security Applications Conference* (*ACSAC* '20), NY, USA, pp. 304–319.

Jiang, P., Wu, H., Xin, C., 2021. DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network. Digital Communications and Networks.

Dang, Y., Benzaïd, C., Taleb, T., Yang, B., Shen, Y., 2022. Transfer learning based GPS spoofing detection for cellular-connected UAVs. In: *2022 International Wireless Communications and Mobile Computing* (*IWCMC*), Dubrovnik, Croatia, pp. 629–634.

Daily GPS Broadcast Ephemeris Files. [Online]. Available: https://cddis.nasa.gov/archive/gnss/data/daily/

Hauke, J., Kossowski, T., 2011. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. Quaestiones geographicae 30 (2).

Mohammad Nayfeh received the B.Sc. degree in communications and electronics engineering from Jordan University of Science and Technology, in 2017. He is currently pursuing the M.Sc. degree in electrical and computer engineering at Purdue University Northwest. His research interests include machine learning, cybersecurity, and wireless communications.

Yuchen Li received the B.Sc. degree in communications engineering from Tianjin University of Technology, Tianjin, in 2019, and the M.Sc. degree in electrical and computer engineering from Purdue University Northwest, in 2021. His research interests include machine learning, deep learning, cybersecurity, and wireless communications.

Khair Al Shamaileh received the B.Sc. degree in communications and electronics engineering and the M.Sc. degree in wireless communications engineering from Jordan University of Science and Technology, in 2009 and 2011, respectively, and the Ph.D. degree in engineering from The University of Toledo, USA, in 2015. He joined the ECE Department at Purdue University Northwest as an Assistant Professor in 2016 and was promoted to an Associate Professor in 2022. His research interests include physical layer security, microwave modeling, RF circuit design, sensor networks, localization algorithms, cybersecurity for autonomous systems, and artificial intelligence. His research has been sponsored by the National Science Foundation.

Vijay Devabhaktuni received the B.Eng. degree in electrical and electronics engineering, the M.Sc. degree in physics from the Birla Institute of Technology and Science, Pilani, India, in 1996, and the Ph.D. degree in electronics from Carleton University, Ottawa, Canada, in 2003. He held the competitive Natural Sciences and Engineering Research Council of Canada (NSERC) Postdoctoral Fellowship and spent the tenure researching with Dr. J. W. Haslett with the University of Calgary, Calgary, Canada, from 2003 to 2004. In 2005, he taught with Penn State Behrend. From 2005 to 2008, he held the Canada Research Chair of Computer-Aided High-Frequency Modeling and Design with Concordia University, Montreal, Canada. In 2008, he joined the Department of Electrical Engineering and Computer Science, The University of Toledo, Toledo, as an Associate Professor, and was promoted to a Professor, in 2013. In 2018, he joined Purdue University Northwest, Hammond, as the Chair of the Electrical and Computer Engineering Department, and in 2020, he joined The University of Maine as the Chair of the Electrical and Computer Engineering Department. He secured external funding close to $5M in his research areas (sponsoring agencies include AFOSR, AFRL, CFI, NASA, NIST, NSERC, NSF, ONR, and industry partners). He has authored 250 peerreviewed papers. His research interests include applied electromagnetics, biomedical applications of wireless sensor networks, computer-aided design, device modeling, image processing, infrastructure monitoring, neural networks, RF/microwave design, unmanned aerial vehicles, and virtual reality. In Canada and USA, he graduated 75 theses students at the M.S. and Ph.D. levels and won student nominated teaching excellence awards. He served as an Associate Editor for the International Journal of RF and Microwave Computer-Aided Engineering under the Editor-in-Chief Dr. I. Bahl. He is also a Professional Engineer of the Association of Professional Engineers and Geoscientists of Alberta.

Naima Kaabouch is a Professor in the Electrical Engineering Department at the University of North Dakota, USA. She is the Director of two research laboratories located within the College and Engineering & Mines at UND. She got her Ph.D., M.S., and B.S. in Electrical Engineering from the University of Paris 6 and the University of Paris 11, France. Her research interests include signal/image processing, sensing, smart systems, and cognitive radio systems. Examples of her current projects include radio spectrum access and management, payloads and algorithms for space applications, radars to remotely monitor vital signs, cybersecurity, and breast micro calcifications detection.