IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Securing Autonomous Vehicles Against GPS Spoofing Attacks: A Deep Learning Approach

MALIHA SHABBIR[1], MOHSIN KAMAL[2], (SENIOR MEMBER, IEEE), ZAHID ULLAH[3] AND MAQSOOD MUHAMMAD KHAN[4]

[1]Department of Electrical Engineering, National University of Computer and Emerging Sciences, Lahore, Pakistan
[2]School of Electrical Engineering and Computer Science, National University of Science and Technology, Islamabad, Pakistan
[3]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, 20133 Milano, Italy
[4]Department of Electrical Engineering, National University of Computer and Emerging Sciences, Peshawar, Pakistan

Corresponding author: Zahid Ullah (e-mail: zahid.ullah@polimi.it).

ABSTRACT With the rapid advancement of technology and multimedia systems, ensuring security has become a critical concern. Connected and Autonomous Vehicles (CAVs) are vulnerable to various hacking techniques, including jamming and spoofing. Global Positioning System (GPS) location spoofing poses a significant threat to CAVs, compromising their security and potentially endangering pedestrians and drivers. To address this issue, this research proposes a novel methodology that uses deep learning (DL) algorithms, such as Convolutional Neural Networks (CNN), and machine learning (ML) algorithms, such as Support Vector Machine (SVM), to protect CAVs from GPS location spoofing attacks. The proposed solution is validated using real-time simulations in the CARLA simulator, and extensive analysis of different learning algorithms is conducted to identify the most suitable approach across three distinct trajectories. Training and testing data include GPS coordinates, spoofed coordinates, and localization algorithm values. The proposed machine learning algorithm achieved 99% and 96% accuracy for the best and worst case scenarios, respectively. In case of deep learning, it achieved as high as 99% for best and 82% for the worst case scenario.

INDEX TERMS Connected and Autonomous Vehicles, Convolutional Neural Networks, Security, GPS Spoofing, Support Vector Machine, CARLA

I. INTRODUCTION

THE autopilot system such as autonomous vehicles or drones are frequently used for surveillance systems, secure communication and packet delivery. Relying on GPS measurements aided by precise high definition maps, autonomous vehicles choose shortest and optimized path from starting point to destination [1]. This is mandatory for such vehicles in order to operate autonomously as well as correctly without any sort of human intervention [2]. Thus the reliability and secure operation of GPS sensor is crucial factor for the wider acceptance of such vehicles. During any unforeseen condition, the communication signals that are exchanged between the autonomous vehicles and the ground stations can be lost or corrupted by incorporating some cyber-attacks such as spoofing or jamming [3]. Jamming attacks refers to the fully blockage of the GPS operation via the disruptive signal transmission on the same frequency as that of GPS signals [4]. On contrary to this, spoofing attack refers to deceiving the user by transmitting the signals possessing same characteristics just like the legitimate GPS satellite signals [5].

To resist such cyber-attacks, it is crucial that the autonomous vehicle architecture to be robust. Autonomous vehicles can be attacked in two forms which includes Denial of Service attack (DoS) and integrity attack [6]. False data injection and spoofing comes under the category of integrity attack while black and gray hole attack and jamming comes under the category of DoS attack. Global Naviagtion Satellite System (GNSS) spoofing involves manipulating signals to misguide receivers, potentially causing dangerous consequences. Despite increased interest in GNSS spoofing, there is a lack of Commercial off the Shelf (COTS) receivers capable of countering advanced attacks. Addressing this gap is crucial to ensure the security and reliability of GNSS systems [7]. We can categorize the GPS spoofing into two major classes, refined receiver based spoofers, GPS signal simulator and receiver based spoofers [6]. In first category, it is supposed that position and velocity of the victim receiver are known precisely and such spoofing is quite impossible to detect using the traditional anti-spoofing techniques. In the second category, the simulators used to send the GPS signals which are concatenated with radio signals in order to produce duplicate GPS signal [8].

For interference detection, signal classification, multipath detection and data quality assurance, machine and deep learning is being utilized in GNSS [8]. Various machine learning as well as deep learning based algorithms are also developed for the detection of GPS spoofing attack. Most widely used algorithms for GPS spoofing attack detection are decision trees, support vector machines and neural networks [8]. Monitoring of cross correlation of multiple GNSS measurements and observables can be used for the detection of potentially spoofed signals [9]. The stability and accuracy of the GNSS absolute solutions in case of autonomous vehicles can be significantly improved using the multi-layer recurrent neural networks in combination with long-short term memory (LSTM) algorithms [10]. The deep learning methods can be used for the vehicle position prediction based on the multi-sensors data which includes GNSS, without the redesigning of the analytical model of every individual sensors on the autonomous vehicle [11].

The key contributions of this paper are outlined as follows:

- CARLA[1] is used in this research to acquire real-time sensor values, specifically yaw rate ($\phi_{.}$), steering angle (α), wheel speed (v) and GPS receiver data. These sensor values serve as crucial inputs for training and evaluating the GPS spoofing attack detection model enabling realistic simulation environment.
- Novel sensor fusion method is developed for integrating data from diverse sensors, such as yaw rate ($\phi_{.}$), steering angle (α) and GPS. This sensor fusion approach enhances the accuracy and reliability of GPS spoofing detection by incorporating multiple sensor modalities thus leading to improved detection performance.
- GPS location spoofing attack detection solution is proposed based on machine and deep learning algorithms. Leveraging the CARLA dataset, the detection system employs state-of-art techniques, including anomaly detection and pattern recognition to differentiate between genuine and spoofed signals. The proposed solution is evaluated in terms of precision, recall, F1 score and accuracy through multiple scenarios using realistic data.

The rest of the paper is organized as follows. Section II reviews the related work on GPS location spoofing attacks and detection techniques. Section III presents the proposed methodology, explaining the algorithm and framework for GPS spoofing detection. Section IV describes the experimental setup, including the utilization of the CARLA dataset. It also presents the results and analysis, discussing the performance of machine learning and deep learning algorithms. Section V concludes the paper and suggests future research directions.

II. RELATED WORK

The most common approaches used for the detection of GPS location spoofing attacks includes signal processing and data driven techniques. However, solutions based on signal processing requires prior knowledge of the expected signal properties, making them vulnerable to attacks that exploit such assumptions and also require specialized equipment. Data driven approach employs machine or deep learning algorithms for pattern detection and anomalies in large datasets and no specialized equipment is required.

The vulnerability of CAVs to GPS location spoofing attacks is explored in [1]. It proposed a data-driven approach based on machine learning to detect these attacks, using only normal location data for training. The solution is tested and evaluated using realistic data and demonstrates over 98% accuracy in detecting attacks.

The vulnerability of Unmanned Aerial Vehicles (UAVs) to GPS signal spoofing attacks is discussed in [3]. The article proposed a machine learning-based solution using SVMs to detect counterfeit GPS signals. Experimental analyses demonstrated the effectiveness of the model in accurately identifying spoofed signals, surpassing existing techniques. The proposed solution achieved 96% accuracy in detecting GPS spoofing attacks.

The use of machine learning in GNSS applications is explored in [8]. A systematic review of literature is presented, encompassing various applications of machine learning in GNSS, including signal acquisition, classification, prediction, and anomaly detection. The article also addresses challenges and potential future applications of machine learning in GNSS. Highlighted applications include earthquake warning systems, hurricane tracking, ice detection and thickness estimation, as well as soil moisture estimation. The conclusion drawn from the review is that machine learning has the potential to enhance the accuracy and reliability of GNSS applications, while also paving the way for further research and exploration of new possibilities in the field [16].

The proposed paper [9] introduces a machine learning-based method for detecting potentially spoofed GNSS signals. The approach involves monitoring the cross-correlation of multiple GNSS observables and measurements. To validate the approach, both synthetic and real-world spoofing datasets were utilized. The results demonstrated the effectiveness of monitoring cross-correlation among significant GNSS observables and measurements in detecting spoofing signals. SVM classification was employed for the spoofing detection, achieving an impressive accuracy rate of 97.8%.

The open service (OS) signals of any GNSS core constellation were vulnerable to manipulation, presenting a significant risk for Safety-of-Life (SoL) applications. Two categories of data manipulation, namely spoofing and meaconing, were

[1] https://carla.org/

TABLE 1: Comparison of Related Work

Algorithms	Ref [1]	Ref [2]	Ref [3]	Ref [9]	Ref [12]	Ref [13]	Ref [14]	Ref [15]	Proposed
Machine Learning	✓	✗	✓	✓	✗	✓	✓	✓	✓
Deep Learning	✗	✗	✗	✗	✓	✗	✗	✗	✓

identified [4]. Spoofing involved generating and transmitting manipulated false GNSS signals, while meaconing consisted of recording and rebroadcasting authentic signals with a controlled delay. The threat of GNSS signal spoofing escalated with advancements in digital signal processing and the hardware implementations of Software Defined Radio (SDR) GNSS-spoofing transceivers. In response, the authors [17] proposed the GNSS signal post-correlation method along with machine learning algorithms to detect the presence of spoofing signals. Previous researchers had successfully employed SVM-based approaches, achieving success rates ranging from 94 to 95%.

In [12], the vulnerability of UAVs to GPS spoofing attacks, which involve attackers disguising themselves as genuine GPS signals to manipulate the navigation and positioning of UAVs, was discussed. The article proposed a novel GPS spoofing attack detection algorithm utilizing LSTM. The algorithm aimed to predict the flight paths of UAVs and identify deviations from these paths as potential GPS spoofing attacks. The article asserted that this algorithm outperformed existing detection methods in terms of efficiency and adaptability. To evaluate the algorithm's performance, it was tested in a simulation environment. The results demonstrated its effectiveness in detecting GPS spoofing attacks, with a detection ratio of 78%. Additionally, the computation time required for the algorithm ranged from 3 to 5 seconds.

Machine learning-based methodology for the automatic and accurate detection of amplitude ionospheric scintillation events, which induce fluctuations in satellite broadcast signals is explored in [18]. The approach utilized common GNSS stand-alone receivers observables and achieved a high detection accuracy of 98% without prefiltering or excluding low-elevation angle data. It outperformed traditional scintillation detection techniques by reducing false alarms and missed detections. The authors also provided an overview of scintillation effects on GNSS signals and analyzed machine learning algorithms, models, and metrics for performance evaluation. Decision trees were highlighted as robust, nonlinear learners with the ability to avoid overfitting through pruning or ensembling techniques. However, it was acknowledged that individual decision trees could be prone to overfitting if they memorized the training data by excessively branching.

Utilization of deep learning models to enhance the modeling of multipath propagation effects on GNSS correlation outputs is discussed in [19]. A DNN structure was proposed as a substitute for standard correlation schemes to effectively model multipath channels. The proposed solution could be seamlessly integrated into acquisition and tracking receiver blocks, exhibiting promising performances in time-delay tracking. The analysis of our proposed model along with the previous research is shown in Table 1.

III. PROPOSED METHODOLOGY

In this paper, we employed various machine learning and deep learning algorithms to propose a mechanism for detecting authentic and spoofed GPS location. In machine learning algorithms, SVM proved valuable for this task. However, tuning the algorithm and selecting the appropriate kernel for SVM are critical factors. On the other hand, deep learning algorithms require large amounts of data, significant computational resources, and extensive hyper-parameter tuning. They may not perform well on small datasets. The performance of both machine learning and deep learning algorithms can vary depending on the characteristics of the data.

A. DATA ACQUISITION AND SYSTEM MODEL

The proposed system model is shown in Figure 1. The proposed methodology involves the acquisition of data from CAV equipped with a GPS receiver and a specialized device with Software Defined Radio (SDR) hardware and software. The CAV moves on a road network, and its true location p_k and velocity u_k are represented as

$$\mathbf{p}_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix}, \tag{1}$$

$$\mathbf{u}_k = \begin{bmatrix} x'_k \\ y'_k \end{bmatrix}, \tag{2}$$

where x_k and y_k represents the x and y coordinates of CAV's location at time k respectively. In the same manner x'_k and y'_k represent the horizontal and vertical components of CAV's velocity at time k, respectively. The GPS receiver processes satellite positioning signals to output the GPS location of the CAV as

$$\mathbf{p}_k^G = \begin{bmatrix} x_k^G \\ y_k^G \end{bmatrix}, \tag{3}$$

where p_k^G is a two-dimensional vector that represents the CAV's GPS location at time k with x_k^G represents the CAV's latitude and y_k^G represents the CAV's longitude. The underlying assumption in this work is that a user defined constant bias is introduced by the attacker in the GPS location Values. The GPS location of the CAV under attack is modeled as a Gaussian random variable as

$$\mathbf{p}_k^G \sim \mathcal{N}(\mathbf{p}_k + \mathbf{B}_A, \mathbf{\Sigma}_k^G), \tag{4}$$

$$\mathbf{\Sigma}_k^G = \mathrm{diag}(2\sigma_k^G), \tag{5}$$

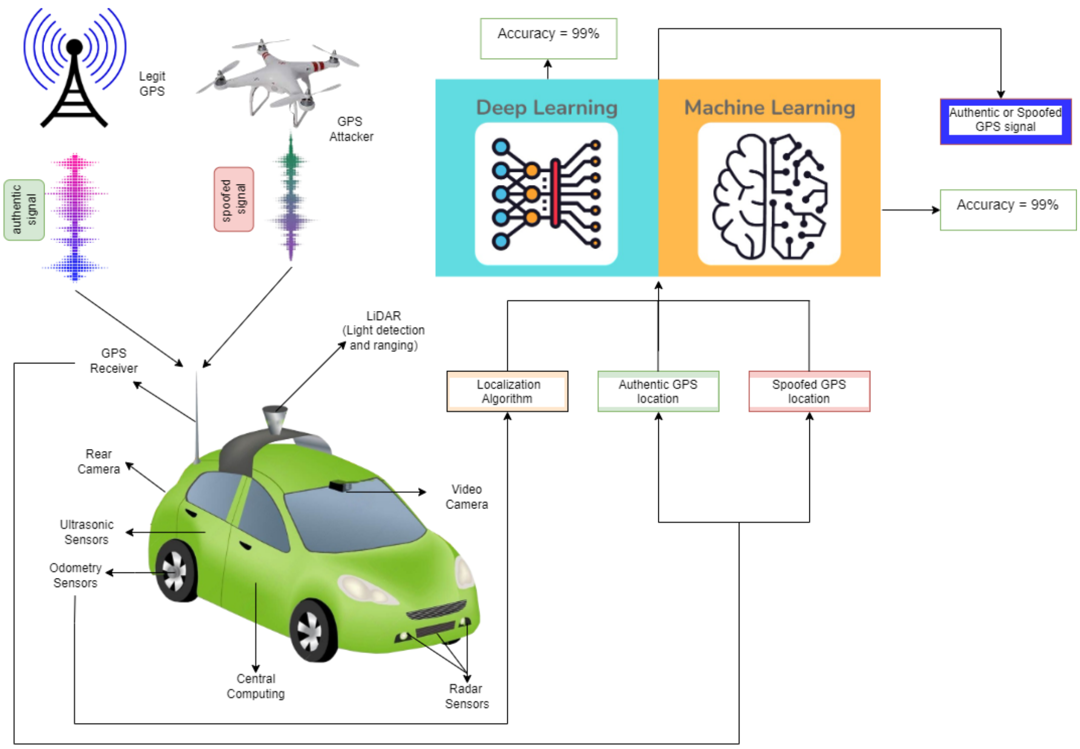


FIGURE 1: GPS location spoofing attack scenario on CAVs

where

$$\mathbf{B}_A = \begin{bmatrix} b_x \\ b_y \end{bmatrix}. \tag{6}$$

B_A represents the attack vector with b_x and b_y as the attack biases, indicating the magnitude of the attack in meters. In case of attack free scenario, $B_A = 0$. p_k^G represents the GPS location of the CAV at time k. p_k is the true location of the CAV at time k, B_A is the attack vector, $\mathbf{\Sigma}_k^G$ is the covariance matrix and $\boldsymbol{\sigma}_{\boldsymbol{k}}^{\boldsymbol{G}}$ is the standard deviation of the GPS location.

A specialized device, equipped with SDR software and hardware, is integrated into the CAV to monitor signals from the surrounding connected vehicles and wireless network infrastructure. This device operates autonomously, without relying on GPS measurements. At the heart of its functionality lies the Localization Algorithm (LA), specifically designed to estimate the precise location of the CAV based on these signals; see [20] for more details of localization. By using the characteristics of these signals, such as signal strength, time-of-arrival, or signal propagation patterns, the algorithm outputs the estimated location p_L^k of CAV's which is denoted as

$$\mathbf{p}_k^L = \begin{bmatrix} x_k^L \\ y_k^L \end{bmatrix}, \tag{7}$$

where p_k^L represents the CAV's location estimated by the localization algorithm at time k. p_k^L is a two-dimensional vector that represents the CAV's location, with x_k^L representing the CAV's localized x-coordinates and y_k^L representing the CAV's localized y-coordinates. The localization algorithm uses the information from on-board sensors and more precisely the yaw rate ($\boldsymbol{\phi}_.$), steering angle ($\boldsymbol{\alpha}$) and wheel speed (v) measurements for the estimated location of CAV's. The LA measurements are modelled as Gaussian random variable and represented as

$$\mathbf{p}_k^L \sim \mathcal{N}(\mathbf{p}_k, \mathbf{\Sigma}_k^L), \tag{8}$$

$$\mathbf{\Sigma}_k^L = \text{diag}(2\sigma_k^L), \tag{9}$$

where σ_k^L is the standard deviations of the LA measurement and $\mathbf{\Sigma}_k^L$ is the covariance matrix.

There are three major steps involved in our proposed solution; see [2] for detailed overview of such algorithm. The first step is of prediction in which the reading from on-board sensors specifically yaw rate ($\boldsymbol{\phi}_.$), steering angle ($\boldsymbol{\alpha}$) and wheel speed (v) are used for the CAV's location prediction represented as

$$\hat{\mathbf{p}}_{k+1} = \begin{bmatrix} \hat{x}_{k+1} \\ \hat{y}_{k+1} \end{bmatrix}, \tag{10}$$

at time $k+1$, where CAV's previously refined location is given as

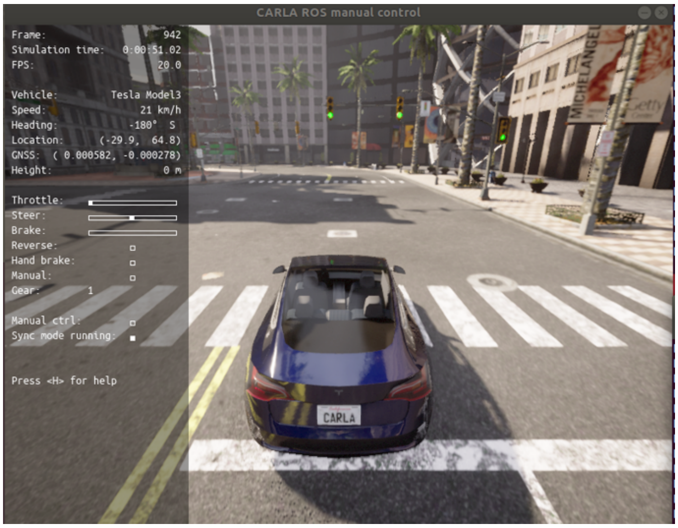
$$\hat{\mathbf{p}}_k = \begin{bmatrix} \hat{x}_k \\ \hat{y}_k \end{bmatrix}. \tag{11}$$

In the second step, CAV's location measurements i.e. p_{k+1}^L which are independent from GPS values, are used to update the values obtained from the first step by means of Bayesian filtering and output the values of refined location estimate $\hat{p}_{k+1}$. In the last step, the GPS location measurements from the GPS receiver of the CAV (authentic and spoofed) i.e. p^G_{k+1}& $\hat{p}^G_{k+1}$ respectively, along with the values obtained in the second step i.e. $\hat{p}^L_{k+1}$ are used with their corresponding labels (0 for spoofed and 1 for authentic) for the training and testing purposes of our proposed machine and deep learning model for the detection of location spoofing. The entire process is represented in Algorithm 1.

Algorithm 1: Data Acquisition from CARLA Simulator: Implementing Spoofing and Localization Algorithm

```
Input  : $SensorsData, GPSReceiverData$
Output: $x^G_k, y^G_k, \dot{x}^G_k, \dot{y}^G_k, \hat{x}^L_k, \hat{y}^L_k$
$p^G_k \leftarrow$ Fetch GPS Receiver Data
ApplyFixedBiasAttack($p^G_k$, $B_A$):
$\dot{p}^G_k \leftarrow p^G_k + B_A$
LocalizationAlgorithm($v_k, \delta_k, \omega_k$):
Set the initial state of the vehicle: $x_{\text{real}}, y_{\text{real}}$
for each movement step i from 1 to movementSteps
do
    Generate random values for $v_k, \delta_k, \omega_k$
    Update the predicted state using the vehicle
      dynamics equations:
    $x_{\text{pred}} = x_{\text{real}} + v_k \cdot \cos(\theta_{\text{real}}) \cdot \Delta t$
    $y_{\text{pred}} = y_{\text{real}} + v_k \cdot \sin(\theta_{\text{real}}) \cdot \Delta t$
    Update the process covariance matrix based on the
      predicted state and noise covariance
    Obtain the current sensor measurements:
      $x_{\text{measured}}, y_{\text{measured}}$
    Generate random values for measurement noise:
      $n_x, n_y$
    Calculate the innovation or measurement residual:
    $\delta x = x_{\text{measured}} - x_{\text{pred}} + n_x$
    $\delta y = y_{\text{measured}} - y_{\text{pred}} + n_y$
    Calculate the innovation covariance matrix:
    $S = H_k \cdot P_{\text{pred}} \cdot H_k^T + R_{\text{GPS}}$
    Calculate the Kalman gain:
    $K = P_{\text{pred}} \cdot H_k^T \cdot S^{-1}$
    Update the state estimate:
    $\hat{x}^L_k = x_{\text{pred}} + K[0] \cdot \delta x$
    $\hat{y}^L_k = y_{\text{pred}} + K[1] \cdot \delta y$
    Update the error covariance matrix:
    $E_{\text{est}} = (I - K \cdot H_k) \cdot P_{\text{pred}}$
    Update the real state variables:
    $x_{\text{real}} = \hat{x}^L_k, y_{\text{real}} = \hat{y}^L_k$
end
```

We conducted an analysis of our machine and deep learning models using three distinct datasets obtained from the CARLA simulator as shown in Figure 2. The first dataset consisted of 1246 samples, the second dataset comprised 2397 samples, and the third dataset contained 5777 samples. Each dataset contains values of p^G_{k+1}, $\dot{p}^G_{k+1}$, $\hat{p}^L_{k+1}$ and corresponding labels i.e. 0 for spoofed and 1 for authentic data. The sampling rate we used for these dataset is 40 Hz i.e. the simulator updates the real world state 40 times per second.



```
Frame:                     942
Simulation time:   0:00:51.02
FPS:                      20.0

Vehicle:          Tesla Model3
Speed:                 21 km/h
Heading:              -180°  S
Location:         (-29.9, 64.8)
GNSS:  ( 0.000582, -0.000278)
Height:                    0 m

Throttle:
Steer:
Brake:
Reverse:            □
Hand brake:         □
Manual:             □
Gear:       1

Manual ctrl:        □
Sync mode running:  ■

Press <H> for help
```

FIGURE 2: Data fetching from CARLA simulator

Additionally, for each dataset, we examined the performance of our algorithms across three bias values i.e. B_A, which correspond to the detection accuracy of GPS spoofing within specific distance thresholds. b_x & b_y were set at $3, 5$ & 9 meters for each dataset.

B. MACHINE AND DEEP LEARNING ALGORITHMS

1) Support Vector Machine

SVM is a powerful machine learning classifier used to classify future predictions into different classes. As a supervised learning algorithm, SVM requires a portion of the dataset for training in order to make predictions on new data. In this work, SVM is employed to effectively discriminate between genuine and spoofed instances. To perform the classification task, we take into account the values x^G_k, y^G_k, $\dot{x}^G_k$, $\dot{y}^G_k$, $\hat{x}^L_k$ and $\hat{y}^L_k$ as input, which determine the dimensionality of the dataset and can be represented as:

$$X = x^G_k, y^G_k, \dot{x}^G_k, \dot{y}^G_k, \hat{x}^L_k, \hat{y}^L_k \tag{12}$$

SVM aim to find an optimal hyperplane that separates the features into different classes with maximum margin. In a 2D dataset, a line (support vector) can accomplish data classification with maximum margins. Hyperplane can be expressed as:

$$Cx + w = 0 \tag{13}$$

To find the optimal hyperplane, we minimize the equation:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i(w^T\Phi(x_i) + b)), \tag{14}$$

subjected to the constraints:

$$y_i(w^T\Phi(x_i) + b) \geq 1 - \xi_i, \tag{15}$$

where $\xi_i \geq 0$. $||w||$ represents the Euclidean norm of the weight vector w, C is the regularization parameter, y_i is the class label for sample x_i, $\Phi(x_i)$ is the feature vector transformed using the kernel function, w^T denotes the transpose of the weight vector w, b is the bias term, and ξ_i is the slack variable. Given a test sample x, we compute the feature vector $\Phi(x)$ and use the trained SVM classifier to make predictions with the expression:

$$y_{\text{pred}} = \text{sign}(w^T\Phi(x) + b) \tag{16}$$

The sign function returns -1 for negative inputs and $+1$ for positive inputs. X_i represents the feature vector which are $x_k^G, y_k^G, \dot{x}_k^G, \dot{y}_k^G, \hat{x}_k^L, \hat{y}_k^L$, and y_i is the label, which is 0 in case of authentic signal or 1 in case of spoofed signal. It can also be expressed as:

$$\begin{aligned} y_{\text{pred}} &= \text{sign}(w^T\Phi(x) + b), \\ &= \begin{cases} 0, & \text{spoofed} \\ 1, & \text{genuine} \end{cases} \end{aligned} \tag{17}$$

SVM also employ different kernels for classification purposes, such as polynomial, RBF, linear, or sigmoid. In this work, all these kernels are utilized in SVM implementation as depicted in Algorithm 2, and the results are reported in Section IV. The hyperparameters used in our proposed SVM algorithm are depicted in Table 2.

```
Algorithm 2: GPS Location Spoofing Attack Detection using SVM
Input  : $x_k^G, y_k^G, \dot{x}_k^G, \dot{y}_k^G, \hat{x}_k^L, \hat{y}_k^L$
Output: accuracy, precision, recall, F1 score, training
        time, prediction time, learning curve data
    Split data into training and testing sets: $X_{\text{train}}$,
      $X_{\text{test}}$, $Y_{\text{train}}$, $Y_{\text{test}}$;
    Define hyperparameters for tuning: $C$, kernel, $\gamma$;
    Perform grid search to find best hyperparameters
      using $X_{\text{train}}$ and $Y_{\text{train}}$;
    Obtain best hyperparameters: $C_{\text{best}}$, kernel$_{\text{best}}$,
      $\gamma_{\text{best}}$;
    Create SVM classifier with best hyperparameters:
      $svm\_model$;
    Start timer;
    Train $svm\_model$ on $X_{\text{train}}$ and $Y_{\text{train}}$;
    Stop timer and calculate training time;
    Start timer;
    Make predictions on $X_{\text{test}}$ using $svm\_model$;
    Stop timer and calculate prediction time;
    Calculate accuracy, precision, recall, and F1 score
      using $Y_{\text{test}}$ and predicted labels;
    Obtain learning curves using $svm\_model$;
    Calculate mean and standard deviation of training
      and test scores
```

TABLE 2: Hyperparameters of ML and DL Algorithm

ML Algorithm Hyperparameters	
C_{best}	0.05
γ_{best}	0.07
$Kernel_{best}$	Linear
DL Algorithm Hyperparameters	
$num\_filters_{best}$	32
$kernel\_size_{best}$	$3 * 3$
$pooling\_size$	$2 * 2$
$hidden\_units$	128
$learning\_rate$	0.009
$batch\_size$	32
num_epochs	20

2) Convolution Neural Network

```
Algorithm 3: GPS Location Spoofing Attack Detection using CNN
Input  : $x_k^G, y_k^G, \dot{x}_k^G, \dot{y}_k^G, \hat{x}_k^L, \hat{y}_k^L$
Output: Accuracy, precision, recall, F1 score, training
        time, prediction time, learning curve data
    Split data into training and testing sets: $X_{\text{train}}$,
      $X_{\text{test}}$, $Y_{\text{train}}$, $Y_{\text{test}}$;
    Define hyperparameters for tuning: num_filters,
      kernel_size, pooling_size, hidden_units,
      learning_rate, batch_size, num_epochs;
    Perform grid search to find best hyperparameters
      using $X_{\text{train}}$ and $y_{\text{train}}$;
    Obtain best hyperparameters: num_filters$_{\text{best}}$,
      kernel_size$_{\text{best}}$, pooling_size$_{\text{best}}$, hidden_units$_{\text{best}}$,
      learning_rate$_{\text{best}}$, batch_size$_{\text{best}}$, num_epochs$_{\text{best}}$;
    Create CNN model with best hyperparameters:
      $cnn\_model$;
    Start timer;
    Train $cnn\_model$ on $X_{\text{train}}$ and $Y_{\text{train}}$;
    Stop timer and calculate training time;
    Start timer;
    Make predictions on $X_{\text{test}}$ using $cnn\_model$;
    Stop timer and calculate prediction time;
    Calculate accuracy, precision, recall, and F1 score
      using $y_{\text{test}}$ and predicted labels;
    Generate learning curves using $cnn\_model$;
    Calculate mean and standard deviation of training
      and test scores
```

By leveraging the power of convolutional layers, pooling layers, and fully connected layers, CNNs can extract intricate spatial features from p_k^G, enabling accurate discrimination between genuine and spoofed GPS signals. Given a labeled dataset D comprising of $x_k^G, y_k^G, \dot{x}_k^G, \dot{y}_k^G, \hat{x}_k^L, \hat{y}_k^L$ as input features and y_i is the corresponding class label (0 for no spoofing and 1 for spoofing), CNNs aim to learn a discriminative mapping function between the inputs and the class label. The output feature map of a convolutional layer is computed as:

$$F_{\text{out}} = \text{conv}(F_{\text{in}}, W) + b, \tag{18}$$

where F_{in} is the input feature map, W is the filter weights, and b is the bias term. The max pooling operation, which down samples the feature maps, is represented as:

$$P_{\text{out}}(i,j) = \max_{m,n \in \text{pooling region}} P_{\text{in}}(m,n), \tag{19}$$

where P_{in} is the input feature map and P_{out} is the output feature map after pooling. The softmax activation function, applied in the final layer, converts the logits into a probability distribution over the classes:

$$P(y=i|x) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}, \tag{20}$$

where z_i is the logarithm of the odds for the event corresponding to the class i, and K is the total number of classes. The acquired data, consisting of p_k^G and $\hat{p}_k^L$, are utilized to train machine and deep learning algorithms for detecting GPS spoofing. The model aim to distinguish between legitimate GPS measurements and spoofed GPS measurements by learning patterns and characteristics from the collected data as depicted in Algorithm 3. The hyperparameters used in our proposed CNN algorithm are depicted in Table 2.

The training process involves feeding the x_k^G, y_k^G, $\dot{x}_k^G$, $\dot{y}_k^G$, $\hat{x}_k^L$, $\hat{y}_k^L$ values with their corresponding labels (i.e., 0 or 1) into the model. We evaluate the performance of our proposed model by analyzing the detection results using a confusion matrix. The confusion matrix categorizes the results into four categories: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). To assess the effectiveness of our attack detection solution, we rely on several metrics derived from the confusion matrix, including Precision (P), Recall (R), and F1 Score. P represents the proportion of correctly identified attacks among all the detected attacks, including false positives. R measures the proportion of correctly identified attacks among all the true attacks, accounting for missed detections (i.e., false negatives). Furthermore, the F1 Score provides a balanced measure by taking into account both P and R. It is a weighted average of P and R, serving as a metric for accuracy on the given dataset. The F1 Score provides insights into the overall performance of our attack detection solution, considering both the ability to correctly identify attacks and minimize FP and FN.

IV. EXPERIMENTAL RESULTS

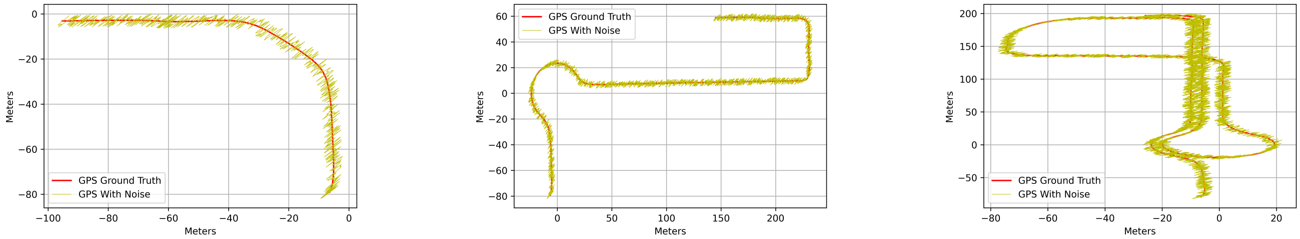
A comprehensive evaluation of our machine and deep learning model for each combination of bias value and dataset is presented in this section. The simulator environment of CARLA for dataset generation is presented in Figure 2. The dataset fetched from CARLA simulator comprises of time, compass, accelerometer readings (x, y, z), gyroscope readings (x, y, z), geolocation coordinates (x, y, z), GNSS latitude, GNSS longitude, GNSS altitude, control gear, control brake, yaw rate ($\phi.$), steering angle (α) and wheel speed (v). Figure 3 illustrates the trajectories along with GPS noise measurements, providing a visual representation of the movement patterns of the vehicle. The figure represents the values of p_k^G, showcasing the variability and noise inherent in the measurements. Figure 4 showcases the values of p_k^L along with the values of p_k^G of dataset 1,2 and 3 respectively. Figure 5 plots the values of p_k^G along with $\hat{p}_k^G$ of dataset 1,2 and 3 respectively. This figure highlights the impact of GPS spoofing, where attackers manipulate GPS signals to deceive the localization algorithm and create a false trajectory. In Figure 6, the box plots showcase the distribution and statistical summary of three distinct datasets. Each box represents the interquartile range (IQR), encompassing the middle fifty percent of the data, with the median line demarcating the center. These box plots offer a visual means of comparing the data distributions and uncovering any discernible dissimilarities or resemblances present in the three datasets. The box plot analysis of dataset 1 reveals that the spoofing attack is more pronounced in the variables x_k^G and $\dot{x}_k^G$ as compared to y_k^G and $\dot{y}_k^G$. This indicates that the spoofing attack has a stronger impact on the GPS coordinates related to the x-axis and its velocity. On the other hand, when considering the comparison between x_k^G and $\hat{x}_k^L$, there is significantly less difference observed. This suggests that the localization algorithm employed shows higher accuracy, as the difference between the estimated localization x_k^L and the actual GPS location x_k^G is relatively small.

The experimental results demonstrate the effectiveness of machine and deep learning algorithms in accurately distinguishing between genuine and spoofed GPS locations. The evaluation metrics, such as Accuracy (A), P, R and F1-score, provide quantitative insights into the performance of the models.

A. MACHINE LEARNING MODEL

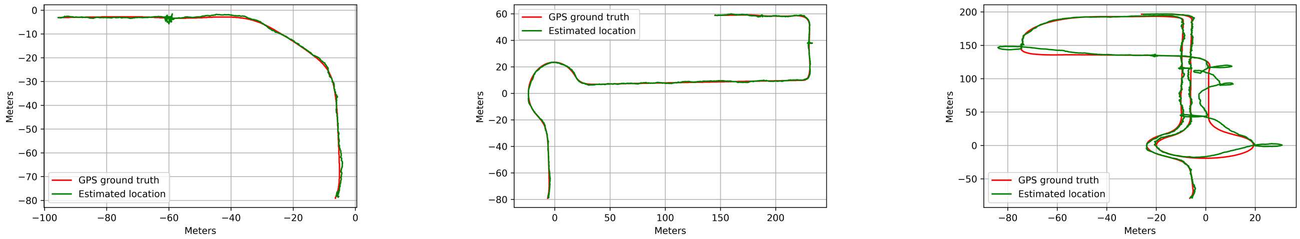
The proposed methodology incorporates K-fold cross-validation to evaluate the performance of the model reliably. The dataset D_A, D_B and D_C comprises of the values of x_k^G, y_k^G, $\dot{x}_k^G$, $\dot{y}_k^G$, $\hat{x}_k^L$ and $\hat{y}_k^L$ along with their labels as 0 or 1, which consists of 1246, 2397 and 5777 samples respectively. We split D_A, D_B and D_C into K where $K = 20$ non-overlapping subsets: $D_1, D_2, ..., D_{20}$. For each iteration of K-fold cross-validation, we select one subset as the test set and use the remaining $K-1$ subsets as the training set. The index of the current iteration is denoted as i, where $1 \leq i \leq K$ and $K = 20$. The training set for iteration i is represented as $D_{i\text{train}}$, and the corresponding test set is $D_{i\text{test}}$.

The model is trained on the training set $D_{i\text{train}}$ and then evaluated on the test set $D_{i\text{test}}$. The performance metrics, such as A, P, R, and F1 score, are calculated based on the predictions of the model on $D_{i\text{test}}$. To assess the performance of the proposed model across different iterations, the K-fold cross-validation process is repeated multiple times, varying the subsets used for training and testing. This helps to mitigate the impact of random variations in the dataset splits. By applying K-fold cross-validation and calculating these performance metrics, we obtain a robust evaluation of the proposed model's effectiveness in detecting the nature of the signal (spoofed or authentic). Figure 7 shows the results of



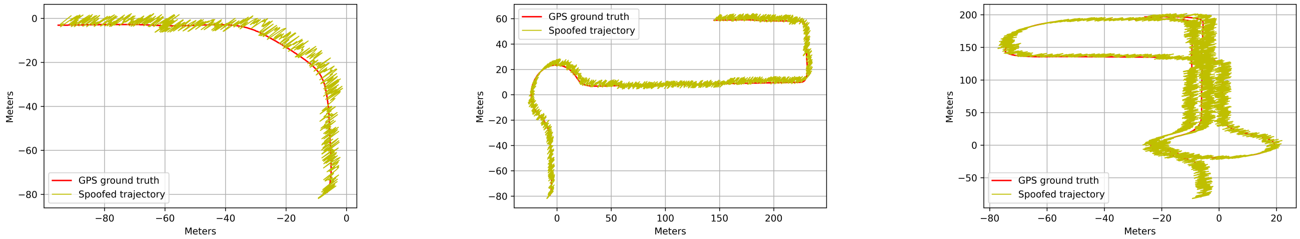
(a) (b) (c)

FIGURE 3: GPS ground truth with noise for dataset 1,2 and 3 respectively



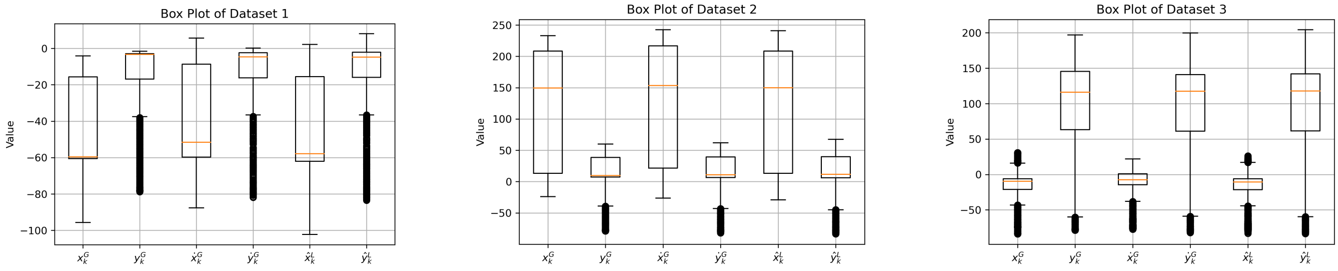
(a) (b) (c)

FIGURE 4: GPS ground truth and estimated location for dataset 1,2 and 3 respectively



(a) (b) (c)

FIGURE 5: GPS ground truth and Spoofed location for dataset 1,2 and 3 respectively



(a) (b) (c)

FIGURE 6: Box plot for dataset 1,2 and 3 respectively

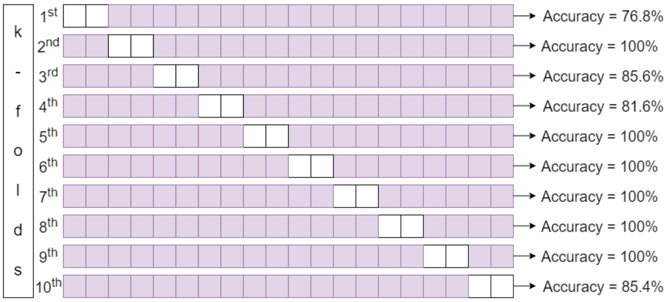


FIGURE 7: K-fold experimentation when 10% of the dataset is chosen in each iteration as the training fold

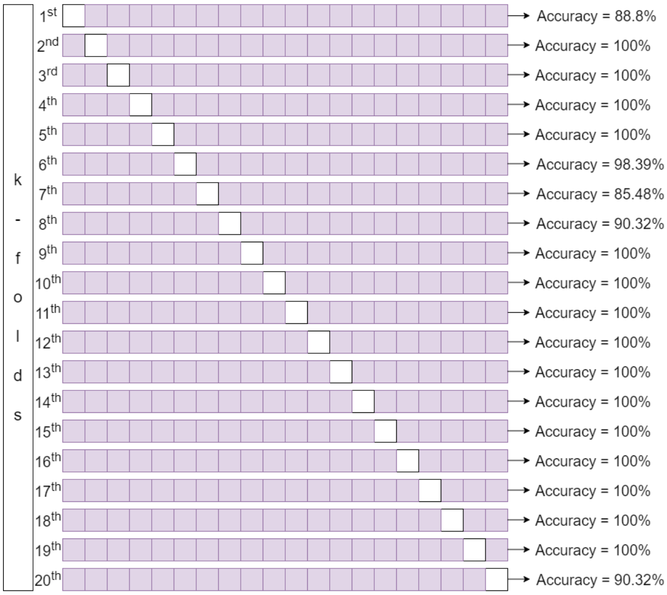


FIGURE 8: K-fold experimentation when 5% of the dataset is chosen in each iteration as the training fold

K-folds experimentation when 10% of the dataset is chosen in each iteration as the training fold and Figure 8 shows the same results in case of 5% training set selection in each fold. The variations in A observed across folds during K-fold cross validation is attributed to the potential over-fitting or under-fitting of the model. To mitigate this issue, the value of K was chosen appropriately, specifically $K = 20$ for our GPS location spoofing attack detection. This choice ensures that the data is sufficiently diversified and reduces the risk of over-fitting or under-fitting, thereby enhancing the reliability of the model's performance evaluation.

Figure 9 depicts the learning curve of the SVM model. It illustrates the relationship between the training set size and the model's training and validation accuracy or loss. The learning curve demonstrates a consistent and promising trend. Starting from an initial accuracy of 0.980, the curve exhibits a steady increase, eventually converging towards a near perfect accuracy of 1. This trend indicates that as the model is exposed to additional training examples, it learns from the data and refines its predictions, leading to higher accuracy. The learning curve's upward trajectory indicates that the model is effectively capturing the underlying patterns in the training, testing and cross validation datasets and successfully generalizing its knowledge to achieve near perfect accuracy showcasing its potential for accurate GPS location spoofing attack detection. The graph clearly demonstrates that convergence takes place for both the training and testing sets at approximately 950 training examples.

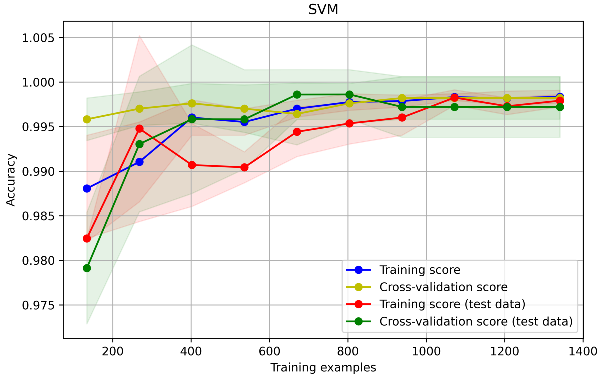


FIGURE 9: Learning curve for SVM

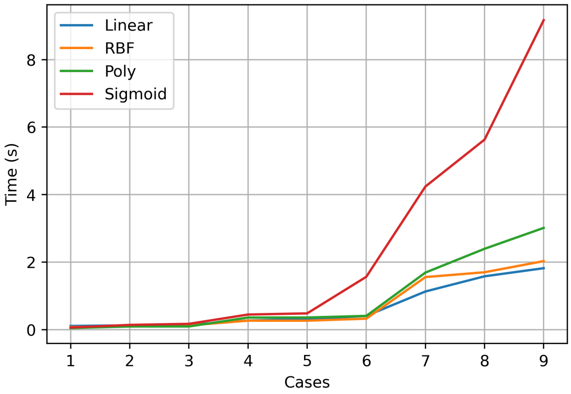


FIGURE 10: Computational time for SVM kernel

Among different SVM kernels, the linear kernel exhibits the best computational time performance as depicted in Figure 10. It is the most efficient and fastest. The RBF (Radial Basis Function) kernel requires more computational time. The polynomial kernel falls in between, while the sigmoid kernel has the highest computational time, making it the least efficient option. For optimal computational efficiency, the linear kernel is recommended for GPS location spoofing attack detection.

The analysis of the accuracy achieved by the SVM kernel when applied to three distinct trajectories is shown in Table 3. Linear kernel achieves accuracy values ranging from 0.96 to 1 across the different cases. The highest accuracy is observed

TABLE 3: Accuracy pertaining to SVM kernel for three different trajectories

Cases	Samples	b (m)	SVM Kernel Accuracy			
			Linear	RBF	Poly	Sigmoid
Case 1	1246	3	0.96	0.90	0.95	0.54
Case 2		5	0.98	0.97	0.98	0.58
Case 3		9	0.99	0.98	0.99	0.78
Case 4	2397	3	0.96	0.96	0.78	0.46
Case 5		5	0.98	0.96	0.80	0.49
Case 6		9	0.99	0.97	0.83	0.55
Case 7	5777	3	0.96	0.85	0.82	0.61
Case 8		5	0.98	0.86	0.90	0.61
Case 9		9	0.99	0.93	0.95	0.68

in case 6 and case 9 which is 0.99. The accuracy of RBF kernel ranges from 0.85 to 0.98. The highest accuracy is observed in case 9 i.e. 0.93, while the lowest is in case 7 i.e. 0.85. For polynomial kernel, the accuracy ranges from 0.78 to 0.95. Case 9 has the highest accuracy of 0.95, and case 4 has the lowest accuracy of 0.78. In case of sigmoid kernel, the accuracy values vary from 0.46 to 0.68. The highest accuracy is observed in case 9 which is 0.68, while the lowest is in case 4 i.e. 0.46.

The results of all SVM kernel are demonstrated in Table 4. The comparative analysis of SVM models reveals that the linear kernel outperforms other kernel functions in terms of P, R and F1 score and generalization, making it the most suitable choice for GPS location spoofing attack detection. The RBF kernel demonstrates competitive performance. However, it falls slightly behind the linear kernel in accuracy and computational efficiency. The polynomial kernel proves effective in handling nonlinear relationships and intricate patterns, particularly in datasets with polynomial characteristics. It requires careful hyper parameter tuning and is computationally demanding for large datasets. The sigmoid kernel shows moderate performance, being capable of handling certain non-linearities but struggling with complex and high-dimensional datasets. Parameter sensitivity and careful tuning are necessary for optimal results.

Table 5 illustrates the confusion matrix of the SVM algorithm with different kernels. The confusion matrix provides a detailed breakdown of the model's predictions, showing the TP (correctly classified spoofed signal, TN (correctly classified spoofed signal), FP (misclassified authentic signal) and FN values (misclassified spoofed signal). The SVM linear kernel achieves perfect accuracy, almost correctly classifying all authentic and spoofed signals in the dataset. It has the highest number of TP and TN indicating excellent performance. The SVM RBF and polynomial kernels also show high accuracy, with a majority of authentic and spoofed signals being correctly classified. However, they have a slightly higher number of FN and FP compared to the linear kernel. The SVM sigmoid kernel demonstrates relatively lower accuracy compared to other kernels with a higher number of misclassifications for both authentic and spoofed signals. Based on these observations, the SVM linear kernel performs the best among the evaluated kernels, achieving the highest accuracy and lowest misclassification rate for authentic and spoofed GPS location.

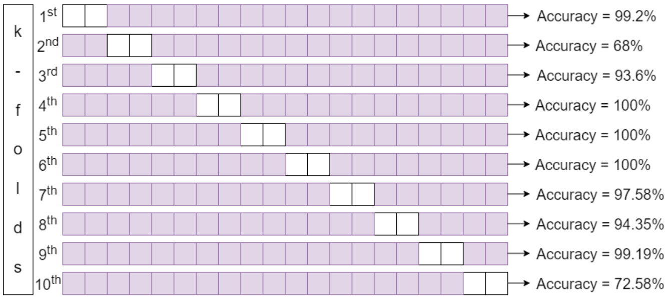


FIGURE 11: K-fold experimentation when 10% of the dataset is chosen in each iteration as the training fold

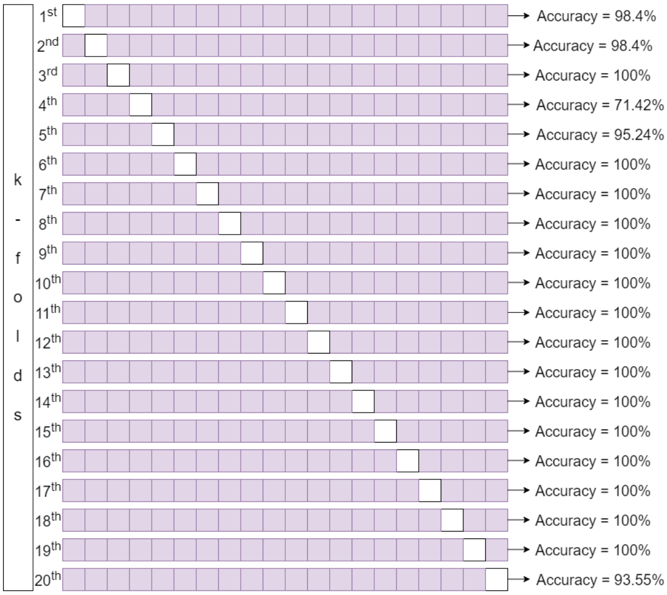


FIGURE 12: K-fold experimentation when 5% of the dataset is chosen in each iteration as the training fold

B. DEEP LEARNING ALGORITHMS

The experimentaional results in case when 10% and 5% of dataset is chosen for each iteration as training fold are shown in Figure 11 and 12, respectively. The experimental results demonstrates that employing a 5% training fold in each iteration of the GPS location spoofing attack detection model leads to higher accuracy compared to using a 10% training fold. This finding suggests a reduction in over-fitting, indicating that the model is better able to generalize to unseen data. Additionally, utilizing a smaller training fold enables a better balance between bias and variance resulting in improved accuracy. These results highlight the importance of considering the appropriate training fold size to mitigate over-fitting and achieve optimal performance in GPS location spoofing attack detection.

The impact of increasing epochs on the accuracy of CNN is shown in Figure 13. By systematically increasing the num-

TABLE 4: Analysis of proposed work corresponding to SVM with different Kernels

Cases	SVM Kernel											
	Linear			RBF			Poly			Sigmoid		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Case 1	0.98	0.97	0.98	0.85	0.93	0.90	0.97	0.92	0.95	0.57	0.37	0.48
Case 2	0.99	0.97	0.98	0.93	0.95	0.97	0.97	0.93	0.98	0.62	0.39	0.48
Case 3	0.99	0.98	0.99	0.98	0.97	0.96	0.99	0.97	0.98	0.45	0.45	0.57
Case 4	0.91	0.95	0.98	0.89	0.92	0.95	0.88	0.65	0.75	0.45	0.26	0.32
Case 5	0.96	0.97	0.98	0.92	0.94	0.96	0.91	0.72	0.82	0.50	0.32	0.39
Case 6	0.99	0.98	0.99	0.97	0.96	0.98	0.94	0.74	0.84	0.58	0.44	0.50
Case 7	0.97	0.99	0.97	0.83	0.86	0.85	0.85	0.78	0.81	0.64	0.46	0.54
Case 8	0.98	0.99	0.98	0.84	0.89	0.87	0.92	0.88	0.90	0.68	0.47	0.58
Case 9	0.99	0.99	0.99	0.95	0.90	0.93	0.98	0.91	0.94	0.71	0.49	0.65

TABLE 5: Confusion Matrix of SVM Algorithm

Total No. of Test Samples	Predicted Authentic Signal	Predicted Spoofed Signal
	Linear	
Actual Authentic Signal	880	3
Actual Spoofed Signal	2	849
	RBF	
Actual Authentic Signal	843	40
Actual Spoofed Signal	82	768
	Poly	
Actual Authentic Signal	867	16
Actual Spoofed Signal	77	773
	Sigmoid	
Actual Authentic Signal	659	224
Actual Spoofed Signal	453	397

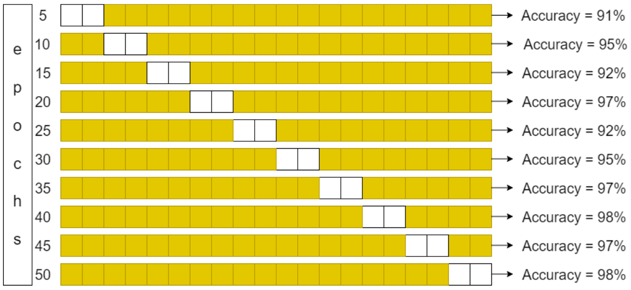


FIGURE 13: The impact of increasing epochs on CNN

ber of epochs during training, the analysis aims to uncover any patterns or trends in the model's performance metrics. The results shed light on the relationship between epoch count and metrics such as accuracy, loss, and convergence rate, providing insights into the optimal number of epochs for achieving optimal model performance. The findings contribute to the understanding of the training dynamics and help in fine-tuning the training process to maximize the deep learning model's predictive capabilities.

As epochs increase, evaluation metrics such as A, P, R, and F1 score tend to exhibit certain trends as shown in Figure 14. Initially, as epochs increase, we observed improvements in the model's evaluation metrics. This indicates that the model is learning and refining its predictions by iteratively adjusting the weights and biases during training but with the increased number of epochs, computational time also increased which is not suitable for our GPS location spoofing attack detection problem. The results in Figure 15 focuses on the impact of increasing epochs on both computational time and accuracy in CNN algorithm. As the number of epochs increases, the computational time required for training the model also increases due to the extended duration of forward and backward passes through the neural network. Therefore, a trade-off needs to be considered between computational time and accuracy when determining the optimal number of epochs for a deep learning model. It is crucial to strike a balance where the model achieves satisfactory accuracy without significantly increasing computational time. Proper model evaluation and monitoring techniques, such as early stopping, can help identify the point of optimal performance to mitigate the risk of over-fitting and unnecessary computational burden. For our GPS location spoofing attack detection model, we conducted the training process using 20 epochs. This choice aimed to achieve a reasonable level of accuracy while managing computational time effectively.

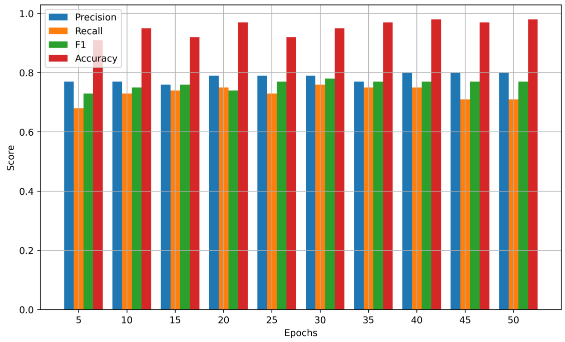


FIGURE 14: The impact of increasing epochs on evaluation metrics

Figure 16 illustrates the learning curve of CNN model. It demonstrates the relationship between the training set size and the model's training and validation accuracy or loss. The

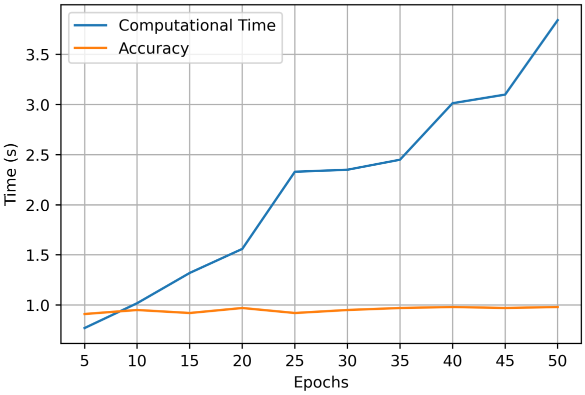


FIGURE 15: The impact of increasing epochs on computational time and accuracy

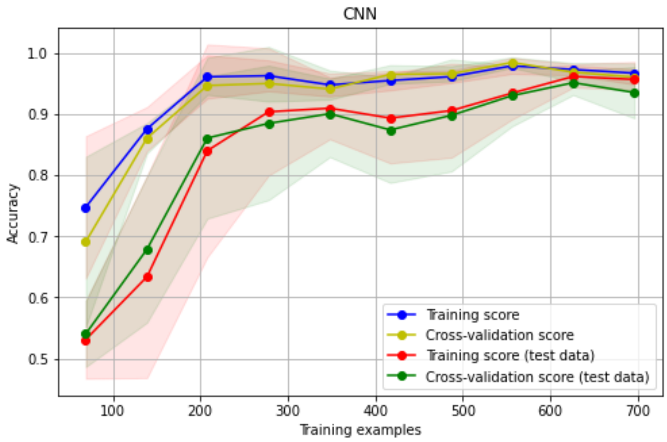


FIGURE 16: Learning curve of CNN

learning curve visually depicts the convergence behavior of the CNN model during the training process. By analyzing the learning curve of the CNN model, we determined that the accuracy remains almost constant after 620 training examples. This information is valuable for fine-tuning the model and improving its performance in GPS location spoofing attack detection.

Figure 17 represents the computational time of the CNN, LSTM and DNN algorithm. The chart displays the time measurements for each model variant: CNN, LSTM, and DNN. The values indicate the duration in seconds for algorithm to process. As observed, the CNN model consistently exhibits the shortest computational time, followed by the LSTM and DNN models.

Table 6 presents the accuracy results obtained from various deep learning algorithms for three different trajectories. The table provides a comparative analysis of the algorithms' performance in terms of accuracy. As observed in Table 6, it is evident that increasing the bias values leads to noticeable improvements in terms of A, P, R, and F1 for each trajectory.

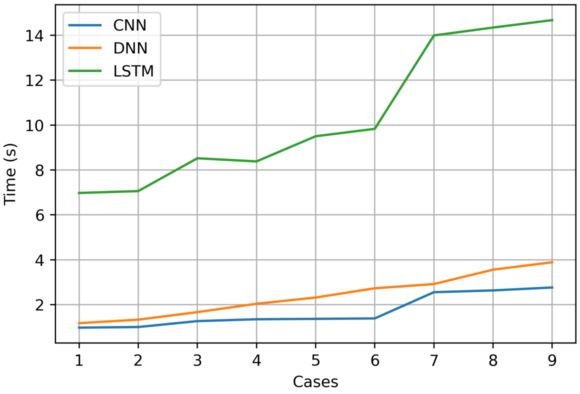


FIGURE 17: Computational time of deep learning models

TABLE 6: Accuracy pertaining to deep learning algorithms for three different trajectories

Cases	Samples	b (m)	DL Algorithm Accuracy		
			CNN	LSTM	DNN
Case 1	1246	3	0.87	0.65	0.58
Case 2		5	0.95	0.68	0.60
Case 3		9	0.99	0.71	0.74
Case 4	2397	3	0.82	0.95	0.76
Case 5		5	0.84	0.95	0.77
Case 6		9	0.99	0.97	0.77
Case 7	5777	3	0.86	0.95	0.63
Case 8		5	0.96	0.95	0.65
Case 9		9	0.99	0.97	0.96

The observed improvement is attributed to the clear distinguishability between authentic and spoofed GPS locations at bias values of 9. However, as the bias values decrease, the proximity between authentic and spoofed GPS signals increases, posing a greater challenge in differentiation. The analysis presented in Table 7 evaluates the proposed work using different deep learning models: CNN, LSTM, and DNN. Performance metrics, including A, P, R, and F1, are evaluated for each model across multiple cases. Consistently, the CNN model outperforms the others across multiple cases, achieving high P, R, and F1. Notably, in Case 2, the CNN model demonstrates superior performance with a P, R and F1 of 0.94, 0.97, and 0.96 respectively. However, the LSTM and DNN models also exhibit competitive performance in specific cases. For instance, in Case 4, the LSTM model achieves a P, R and F1 of 0.78, 0.81, and 0.78 respectively, accurately detecting and classifying GPS spoofing attacks. Similarly, in Case 9, the DNN model achieves P, R and F1 of 0.96, 0.96, and 0.96 respectively, indicating its effectiveness in spoofing detection. The performance of all models may vary across different cases due to varying data characteristics and patterns, impacting detection accuracy. Overall, the results underscore the CNN model's efficacy in achieving higher accuracy and robustness in detecting GPS location spoofing attacks. Meanwhile, the LSTM and DNN models exhibit promising performance in specific scenarios.

TABLE 7: Analysis of proposed work corresponding to different deep learning models

Cases	Deep Learning Models								
	CNN			LSTM			DNN		
	P	R	F1	P	R	F1	P	R	F1
Case 1	0.93	0.80	0.86	0.59	0.83	0.72	0.55	0.78	0.67
Case 2	0.94	0.97	0.96	0.62	0.84	0.74	0.56	0.79	0.71
Case 3	0.99	0.98	0.99	0.65	0.88	0.75	0.72	0.80	0.76
Case 4	0.86	0.93	0.84	0.78	0.81	0.78	0.79	0.55	0.70
Case 5	0.87	0.99	0.86	0.85	0.89	0.82	0.80	0.57	0.72
Case 6	0.99	0.99	0.99	0.95	0.96	0.96	0.95	0.59	0.73
Case 7	0.95	0.75	0.84	0.94	0.65	0.81	0.64	0.89	0.72
Case 8	0.99	0.91	0.95	0.94	0.89	0.92	0.68	0.93	0.74
Case 9	0.99	0.98	0.99	0.99	0.95	0.97	0.96	0.96	0.96

TABLE 8: Confusion Matrix of DL Algorithms

Total No. of Test Samples	Predicted Authentic Signal	Predicted Spoofed Signal
	CNN	
Actual Authentic Signal	879	4
Actual Spoofed Signal	14	836
	LSTM	
Actual Authentic Signal	864	19
Actual Spoofed Signal	44	806
	DNN	
Actual Authentic Signal	846	37
Actual Spoofed Signal	38	812

Table 8 illustrates the confusion matrix for the deep learning models: CNN, LSTM, and DNN. The confusion matrix provides a comprehensive summary of the models' performance, including FN, FP, TN, and TP for each class. By examining the confusion matrix, we can evaluate the accuracy and misclassification patterns of the deep learning models across different classes. Regarding the CNN algorithm, the matrix reveals that out of the total number of test samples, 879 were correctly classified as authentic signals, while only 4 were mistakenly classified as spoofed signals. Similarly, for spoofed signals, 836 were accurately identified, with 14 being misclassified as authentic signals. For the LSTM algorithm, the matrix demonstrates that 864 authentic signals were correctly predicted, while 19 were misclassified as spoofed signals. For the spoofed signals, 806 were correctly identified, and 44 were misclassified as authentic signals. Lastly, for the DNN algorithm, the matrix reveals that 846 authentic signals were correctly predicted, while 37 were misclassified as spoofed signals. Among the spoofed signals, 812 were correctly identified, and 38 were misclassified as authentic signals. Based on these observations, the CNN performs best among LSTM and DNN, achieving the highest accuracy and lowest misclassification rate for authentic and spoofed GPS location.

TABLE 9: Comparison of proposed work with existing ones

Algorithms	Accuracy	Precision	Recall	F1
Proposed	0.99	0.99	0.98	0.99
Ref [2]	0.97	0.95	0.99	0.97
Ref [3]	0.96	0.99	0.93	0.97
Ref [9]	0.97	0.97	0.96	0.96
Ref [12]	0.96	0.96	0.98	0.96
Ref [13]	0.88	0.90	0.92	0.90
Ref [14]	0.87	0.88	0.90	0.86
Ref [15]	0.85	0.81	0.88	0.89

C. COMPARATIVE ANALYSIS

In Table 7, we conducted a comparison between our proposed work and existing studies. In [2], the authors have used bias measurements of 5, 9, and 12 meters, whereas we focused on bias measurements of 3, 5, and 9 meters. Despite the lower bias measurements, higher accuracy was achieved. The proposed algorithm presented in [2] attained an accuracy of 97.57% in the best case, while our approach achieved an accuracy of 99% in both machine learning and deep learning algorithms. Additionally, we performed a similar comparison with the methodologies proposed in papers [3], [9], [12], [13], [14], [15] and the results are included in Table 9.

V. CONCLUSION

In conclusion, this research work addresses the critical security concerns associated with CAVs by proposing a novel methodology that uses DL and ML algorithms. Specifically, CNN and SVM are utilized to protect CAVs from GPS location spoofing attacks. The proposed solution has undergone extensive experimentation and analysis, utilizing real-time simulations in the CARLA simulator. The performance evaluation encompasses different learning algorithms applied to three distinct trajectories, considering metrics such as A, P, R, F1, and computational costs. The results strongly indicate the effectiveness of the proposed approach in mitigating the risks associated with GPS location spoofing attacks on CAVs. By harnessing the power of DL and ML algorithms, the proposed solution demonstrates great potential in fortifying the security of CAVs and reducing potential hazards to pedestrians and drivers. This research makes a significant contribution to the existing knowledge by conducting a comprehensive evaluation and comparison of various learning algorithms in the context of GPS location spoofing detection. The findings highlight the superiority of the proposed methodology over existing solutions, emphasizing the importance of incorporating advanced technologies to safeguard the integrity and security of CAVs. Looking ahead, future research could explore additional ML and DL techniques, as well as real-time implementation and testing on physical CAVs. Continued efforts in this field will play a crucial role in bolstering the security of CAVs, ensuring the safe and reliable deployment of autonomous transportation systems in the future.

REFERENCES

[1] S. Filippou, A. Achilleos, S. Z. Zukhraf, C. Laoudias, K. Malialis, M. K. Michael, and G. Ellinas, "A Machine Learning Approach for Detecting

GPS Location Spoofing Attacks in Autonomous Vehicles," in 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), 2023, pp. 1–7.

[2] M. Kamal, A. Barua, C. Vitale, C. Laoudias, and G. Ellinas, "GPS Location Spoofing Attack Detection for Enhancing the Security of Autonomous Vehicles," in 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), 2021, pp. 1–7.

[3] 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.

[4] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The Security of Autonomous Driving: Threats, Defenses, and Future Directions," Proceedings of the IEEE, vol. 108, no. 2, pp. 357–372, 2020.

[5] G. Vasconcelos, R. S. Miani, V. C. Guizilini, and J. R. Souza, "Evaluation of DOS Attacks on Commercial Wi-Fi-based UAVs," International Journal of Communication Networks and Information Security, vol. 11, no. 1, pp. 212–223, 2019.

[6] E. Ranyal and K. Jain, "Unmanned Aerial Vehicle's Vulnerability to GPS Spoofing: A Review," Journal of the Indian Society of Remote Sensing, vol. 49, no. 3, pp. 585–591, 03 2021. [Online]. Available: https://doi.org/10.1007/s12524-020-01225-1

[7] M. Kamal, C. Kyrkou, N. Piperigkos, A. Papandreou, A. Kloukiniotis, J. Casademont, N. P. Mateu, D. B. Castillo, R. D. Rodriguez, N. G. Durante, P. Hofmann, P. Kapsalas, A. S. Lalos, K. Moustakas, C. Laoudias, T. Theocharides, and G. Ellinas, "A Comprehensive Solution for Securing Connected and Autonomous Vehicles," in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2022, pp. 790–795.

[8] A. Siemuri, K. Selvan, H. Kuusniemi, P. Valisuo, and M. S. Elmusrati, "A Systematic Review of Machine Learning Techniques for GNSS Use Cases," IEEE Transactions on Aerospace and Electronic Systems, vol. 58, no. 6, pp. 5043–5077, 2022.

[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), 2019, pp. 1–6.

[10] H.-U. Kim and T.-S. Bae, "Deep Learning-Based GNSS Network-Based Real-Time Kinematic Improvement for Autonomous Ground Vehicle Navigation," Journal of Sensors, vol. 2019, p. 3737265, 2019.

[11] M. Mendonça and M. C. Santos, "Assessment of a GNSS/INS/Wi-Fi Tight-Integration Method Using Support Vector Machine and Extended Kalman Filter," in Beyond 100: The Next Century in Geodesy, ser. International Association of Geodesy Symposia, J. T. Freymueller and L. Sánchez, Eds. Cham: Springer, 2020, vol. 152.

[12] S. Wang, J. Wang, C. Su, and X. Ma, "Intelligent Detection Algorithm Against UAVs' GPS Spoofing Attack," in 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), 2020, pp. 382–389.

[13] 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.

[14] G. Panice, S. Luongo, G. Gigante, D. Pascarella, C. Di Benedetto, A. Vozella, and A. Pescapè, "A SVM-based detection approach for GPS spoofing attacks to UAV," in 2017 23rd International Conference on Automation and Computing (ICAC), 2017, pp. 1–11.

[15] E. M. d. L. Pinto, R. Lachowski, M. E. Pellenz, M. C. Penna, and R. D. Souza, "A Machine Learning Approach for Detecting Spoofing Attacks in Wireless Sensor Networks," in 2018 IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018, pp. 752–758.

[16] N. Souli, P. Kolios, and G. Ellinas, "Online Relative Positioning of Autonomous Vehicles Using Signals of Opportunity," IEEE Transactions on Intelligent Vehicles, vol. 7, no. 4, pp. 873–885, 2022.

[17] S. Semanjski, I. Semanjski, W. De Wilde, and A. Muls, "Use of Supervised Machine Learning for GNSS Signal Spoofing Detection with Validation on Real-World Meaconing and Spoofing Data—Part I," Sensors, vol. 20, no. 4, p. 1171, 2020.

[18] N. Linty, A. Farasin, A. Favenza, and F. Dovis, "Detection of GNSS Ionospheric Scintillations Based on Machine Learning Decision Tree," IEEE Transactions on Aerospace and Electronic Systems, vol. 55, no. 1, pp. 303–317, 2019.

[19] H. Li, P. Borhani-Darian, P. Wu, and P. Closas, "Deep Learning of GNSS Signal Correlation," in Proc. 33rd Int. Tech. Meeting Satell. Div. Inst. Navigation, 2020, pp. 2836–2847.

[20] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A Survey of Enabling Technologies for Network Localization, Tracking, and Navigation," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 3607–3644, 2018.

MALIHA SHABBIR did her Masters in Electrical Engineering from National University of Computers and Emerging Sciences (FAST-NUCES), Lahore, Pakistan, in 2021. She completed her Bachelors in Electrical Engineering from University of Engineering and Technology (UET), Lahore, Pakistan, in 2016. Currently, she is working as research assistant under the supervision of Dr. Mohsin Kamal in FAST-NUCES, Lahore, Pakistan. Her research interests include Cyber Security, Blockchain and Speech Processing and Analysis.

MOHSIN KAMAL (Senior Member, IEEE) did his Postdoc from KIOS Research and Innovation Center of Excellence, the University of Cyprus in 2022, where he was designated as a Research Associate under the supervision of Professor Georgios Ellinas. He completed his Ph.D. degree in Electrical Engineering from National University of Computer and Emerging Sciences (FAST-NUCES), Peshawar, Pakistan in 2020. He received his M.S. degree in Electrical Engineering from Blekinge Institute of Technology, Karlskrona, Sweden, in 2012. Currently, he is designated as Associate Professor in School of Electrical Engineering and Computer Science (SEECS), National University of Science and Technology (NUST). Previously, he was employed as an Associate Professor in the department of Electrical Engineering at FAST-NUCES, Lahore campus. From February 2022 to August 2022, he was designated as the Incharge of Electrical Engineering Department at FAST-NUCES, Peshawar campus. From March 2013 to February 2021, he worked as an Assistant Professor at FAST-NUCES, Peshawar, Pakistan. He is a PhD approved supervisor and has supervised MS and BS students for their thesis and final year projects, respectively. He is the reviewer of many renowned Q1 journals and also serves as a Technical Program Committee (TPC) member of national and international conferences. His research interests include the development of lightweight solutions for various IoT applications, Cyber Physical Systems, Blockchain, Cyber Security, Wireless Sensor Networks, Cooperative Communication and Cognitive Radio Networks.

ZAHID ULLAH received his BS and MS Electrical Engineering from UET Peshawar and COMSATS University Islamabad, Abbottabad Campus, Abbottabad, Pakistan, in 2014 and 2017, respectively. He is pursuing a Ph.D. in Electrical Engineering with Politecnico di Milano, Italy. He has worked as a Lecturer in UMT Lahore, Pakistan (2017 to 2020). He published various articles in reputed journals and IEEE conference proceedings. His research interests include Smart Grid, Energy Management, Renewable Energy Systems, ICTs for Power Systems, V2G, and Machine and Deep Learning.

MAQSOOD MUHAMMAD KHAN is an Assistant Professor in the department of Electrical Engineering, National University of Computer and Emerging Sciences (FAST-NUCES), Peshawar, Pakistan. He completed his Ph.D. in Electrical Engineering from FAST-NUCES in 2021. His research was a NATO-funded project in which he worked on information reconciliation techniques for quantum key distribution. His current research interests include information reconciliation using quantum and conventional channels, channel modeling & coding, optical communication, cyber security, quantum key distribution, and security aspects of advanced metering infrastructure.