A Reinforcement Learning Approach for GNSS Spoofing Attack Detection of Autonomous Vehicles

Sagar Dasgupta*
Ph.D. Student
Department of Civil, Construction & Environmental Engineering
The University of Alabama
3014 Cyber Hall, Tuscaloosa, AL 35487
Tel : (864) 624-6210 ; Email: sdasgupta@crimson.ua.edu

Tonmoy Ghosh
Ph.D. Student
Department of Electrical and Computer Engineering
The University of Alabama
3067 South Engineering Research Center, Tuscaloosa, AL 35487
Tel : (205) 657-8375 ; Email: tghosh@crimson.ua.edu

Mizanur Rahman, Ph.D.
Assistant Professor
Department of Civil, Construction & Environmental Engineering
The University of Alabama
3015 Cyber Hall, Box 870205, Tuscaloosa, AL 35487
Tel : (205) 348-1717 ; Email: mizan.rahman@ua.edu

This material is based on a study partially supported by the National Science Foundation under Grant No. 2104999. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, and the U.S. Government assumes no liability for the contents or use thereof.

ABSTRACT

A resilient and robust positioning, navigation, and timing (PNT) system is a necessity for the navigation of autonomous vehicles (AVs). A Global Navigation Satellite System (GNSS) provides satellite-based PNT services. However, a spoofer can temper the authentic GNSS signal and could transmit wrong position information to an AV. Therefore, an AV must have the capability of real-time detection and feedback-correction of spoofing attacks related to PNT receivers, whereby it will help the end-user (the AV in this case) to navigate safely even if the GNSS is compromised. This paper aims to develop a deep reinforcement learning (RL)-based turn-by-turn spoofing attack detection method using low-cost in-vehicle sensor data. We have utilized Honda Driving Dataset to create attack and non-attack datasets, develop a deep RL model, and evaluate the performance of the RL-based attack detection model. We find that the accuracy of the RL model ranges from 99.99% to 100%, and the recall value is 100%. Furthermore, the precision ranges from 93.44% to 100%, and the f1 score ranges from 96.61% to 100%. Overall, the analyses reveal that the RL model is effective in turn-by-turn spoofing attack detection.

Keywords: Reinforcement Learning, Cybersecurity, GNSS, GPS, Autonomous vehicle, and Spoofing attack

INTRODUCTION

With the advancement of communication and automation technologies, the landscape of roadway mobility systems is changing radically [1]. Ground vehicles are becoming more automated as well as connected between themselves and with the transportation infrastructure. These advances provides a traveler an opportunity to efficiently move from one location to another location and use their time for personal use while traveling. Positioning, navigation, and timing (PNT) services are the key to the navigation of autonomous vehicles (AVs) [2]. Global Navigation Satellite System (GNSS) provides satellite-based PNT services. The US government-owned GNSS is known as the global positing system (GPS) [3]. GPS is a set of satellites that the US Department of Defense launched in 1970 for military use. However, GPS was known initially as Navigation Satellite Timing and Ranging (NAVSTAR). Such as GPS, several countries have their own satellites for providing PNT services. In the US, GPS provides two different services: Standard Positioning Service (SPS) and Precision Positioning Service (PPS) [3]. As PPS service is available for government and military use, it is expected that autonomous vehicles will use SPS, (also known as civilian GPS) for their PNT services.

AVs require reliable and real-time PNT services. However, the robustness and reliability of GNSS-based PNT services depend on secure and strong satellite signals and radio communications at the receiver end. The long-distance between satellites and GNSS receivers reduces the signal strength and decreases GNSS-based PNT services' reliability. The GNSS signal is also susceptible to natural vulnerabilities, which are known as unintentional vulnerabilities [4]. For example, a GNSS signal can be unavailable to an autonomous vehicle while passing through a tunnel. Even ceilings in garages and thick clouds in the sky could reduce the GNSS signal strength and interrupt the PNT services. In urban areas, tall buildings cause multipath propagation, which causes radio frequency interference and degrades GNSS signal strength [4]. Besides these vulnerabilities, jamming and spoofing are the common intentional threats to GNSS-based PNT services [2], [4]. A jamming attack makes the authentic GNSS signal unavailable to a receiver by flooding the receiver with a high-power signal. On the other hand, a spoofing attacker can temper an authentic GNSS signal and transmit wrong position information to an AV. Even an AV's destination and route choice can be corrupted, and the vehicle could be misguided turn-by-turn to an unwanted destination. This will compromise the safety of AV users. Currently, an attacker can use low-cost software-defined radios to conduct such a spoofing attack [5].

This paper develops a deep reinforcement learning (RL) approach using data from multiple low-cost in-vehicle sensors of an AV to detect a sophisticated turn-by-turn spoofing attack. The presented RL approach will be a new addition to the existing GNSS spoofing attack detection approaches without directly analyzing the GNSS signal characteristics and avoiding IMU/INS-based solutions because of their inability to provide accurate position and acceleration information. The rest of the paper is arranged as follows. In the Related Work section, different spoofing attack techniques, along with state-of-the-art detection methods are presented. Information regarding reinforcement learning is also included. The next section (Methodology) includes data description and data processing, attack model, and generation of turn-by-turn attack dataset, and also introduces the proposed reinforcement learning-based GNSS spoofing attack detection method. The results and conclusion section presents the proof of effectiveness of the developed GNSS spoofing attack detection method.

RELATED WORK

Spoofing is the most sophisticated type of intentional attack on GNSS wherein the AVs receive manipulated GNSS signal. Spoofing attacks can be classified into four types: replay spoofing attack (RSA), forgery spoofing attack (FSA), estimation spoofing attack (ESA), and advanced spoofing attacks (ASA) [6]. In an RSA, an attacker delays the GNSS signal transmission that introduces an error in the position estimation of the AVs. Alternatively, in an FSA, an attacker manipulates the estimated position of an AV by adjusting one or multiple signal parameters (e.g., phase difference) of the GNSS signal. In ESA, an attacker generates a fake GNSS signal matching the actual GNSS signal information. Finally, in an ASA, an attacker combines multiple spoofing attack techniques, which makes the attack very hard to detect. In this paper, a turn-by-turn scenario of a spoofing attack is studied where the attacker's fake signal matches all the turns of the real route. The turn-by-turn attack is a type of FSA in which an AV could be misguided turn-by-turn to an unwanted destination. As it is a sophisticated spoofing attack, an FSA is hard to detect. Although many studies exist related to spoofing attack detection of GNSS using encryption mechanisms, codeless-cross-correlation measures, signal statistics analyses, and antenna-based strategies [4], [7]–[11], existing literature is reviewed related to machine learning and in-vehicle sensors because of their relevance to this study's focus.

Use of In-vehicle Sensor Data for Spoofing Attack Detection

Researchers use in-vehicle inertial navigation system (INS) and inertial measurement units (IMU) sensors for spoofing attack detection strategies as these sensors are resilient against signal spoofing attacks and provide a low-cost solution. Gyroscope and accelerometer are two examples of IMU sensors. A spoofing attack can be detected by comparing IMU-based acceleration and GNSS-based acceleration [12]. However, this approach is not suitable for autonomous ground vehicles as they have a low vehicle dynamics signature compared to the aircraft. In [13], Manickam and O'Keefe have compared position information from the accelerometer and gyroscope with the position information from GNSS to flag a spoofing attack [13]. They also analyzed different types of IMU combined with different GNSS receiver grades to evaluate their performance for attack detection. Researchers also used INS to keep track of the position of a vehicle, and eventually, these methods help to detect GNSS spoofing attacks [14], [15]. Even only the dead reckoning approach has been used to determine the speed, orientation, and position of a vehicle using gyroscope and accelerometer data. However, these in-vehicle sensors provide less reliable position, speed, and orientation information because of scale factor and non-orthogonality errors [14], [16], [17]. In addition, these errors accumulate over time. Thus, there is no study that uses multiple low-cost in-vehicle sensors' data of an autonomous ground vehicle to detect a turn-by-turn spoofing attack.

Machine Learning Models for Spoofing Attack Detection

With the advancement of machine learning (ML) and DL models, several studies investigated the potential of sophisticated spoofing attack detection using these models [15], [18]. ML and DL concepts started from Artificial Intelligent (AI). An AI-enabled machine is a machine that has the capability to mimic the intelligence of a human brain. ML is nothing but a computer program that can learn from the relationship between input data and the feedback based on the error between the predicted data and real data. Neural Network (NN) is an example of machine learning, whereas DL is an advanced version of machine learning. In the DL model, additional layers can be added

to neural networks so that the model can learn more details while training. Each layer of a DL model consists of many neurons. On the other hand, a shallow ML model contains very few (two or three) layers. Logistic regression and support vector machines (SVM) are the two examples of shallow ML [4]. A shallow ML model is not able to learn complex real-world problems with non-linear relationships. Multi-layer perceptron (MLP), Restricted Boltzmann Machine (RBM), Deep Belief Network (DBN), Deep Boltzmann Machine (DBM), Autoencoder, Convolution Neural Network (CNN), and Recurrent Neural Network (RNN) are the popular DL models. MLP is the simplest deep neural network and contains an input layer, multiple hidden layers, and an output layer. An MLP can be fully connected if all neurons are connected with each other. In RBM, a Boltzmann distribution is used for each of the probability distributions, and all neurons are binary variables. DBN consists of multiple unsupervised networks, such as autoencoder or RBMs. DBM is similar to DBN as it consists of multiple RBMs. An encoder and a decoder network are the key components of an automatic encoder or autoencoder network. An encoder network uses training data to extract latent features, and a decoder network utilizes the extracted features to reconstruct the input. In a CNN, each neuron is connected with some of the next ones instead of all of them, allowing it to exploit the properties of two-dimensional data structures. Along with these models, one-class support vector machine (OCSVM) is used for anomaly detection. OCSVM can identify most of the nonlinear boundaries separating class of data due to the flexibility of its models [19]. RNN models work best for sequential data or time-series data due to the capability of using past input data for future prediction.

Borhani-Darian et al. used the Cross Ambiguity Function (CAF) feature to develop a deep learning approach for detecting spoofing attacks [20]. For the model development purpose, they have combined an MLP and two classes of CNNs, which include a complex CNN and a simple CNN, to provide a probability-based attack classification. The deep learning approach was trained and tested using simulated data and proved the potential for detecting spoofed GNSS signals. In another study, Sun et al. [15] used singular values of the wavelet transformation feature for the spoofed and actual GPS signal to train three different attack classifiers: (i) SVM; (ii) probabilistic neural networks (PNN); and (iii) decision tree (DT). Later, they fused the individual classification outcome of these three models with a K-out-of-N rule, which increases the detection accuracy on average by 3.75%, 5.06%, and 12.36% of the SVM, PNN, and DT, respectively. In addition, their K-out-of-N decision rule showed a fewer number of false-positives than each of those three classifiers. Panice et al. presented a GPS spoofing attack detection approach for an Unmanned Aerial Vehicle (UAV) using SVM [18]. They have used SVM to estimate the state of a UAV and identify anomalies of its current location. This approach constructs a decision boundary through training using the data from the actual state of the UAV. The uniqueness of this approach is that it provides a probability if it misses any detection, which is necessary for aviation applications. However, if a spoofer has complete knowledge of a UAV's trajectory data, their detection system could not detect the attack. Instead, it gives high position errors. On the other hand, Shafiee et al. presented a new MLP-NN approach for GPS spoofing attack detection [21]. They have selected three features from the GPS signal pattern—i.e., early-late phase criterion, delta criterion, and total levels of signal— to train and test the performance of their MLP-NN approach. They found 98.7% accuracy for detecting spoofed signals; moreover, the computation time was less than 0.5 seconds. They also compared the performance of K-Nearest neighbor (KNN) and naive Bayesian classifier with the MLP-NN to prove the efficacy of their method. They found that MLP-NN provides a high accuracy compared to the other three classifiers. All of these above-mentioned DL and ML-based

attack detection strategies detect the attacks in the GNSS signal level. In addition, vehicle position information has not been used to detect GNSS spoofing attacks.

In this study, an RL-based spoofing attack detection strategy using multiple low-cost in-vehicle sensor data of an autonomous ground vehicle is presented, an approach that has not been explored previously. Human bias could be incorporated into deep learning (DL) models as it follows a supervised learning approach. On the other hand, the primary benefit of deep reinforcement learning (RL) is that it teaches machines what to do by interacting with the environment. Thus, deep RL models allow a machine to be more efficient and robust compared to a machine that is trained solely through supervised training. Thus, our deep RL-based spoofing attack detection framework could be robust compared to DL models as it teaches machines how to detect an attack by interacting with AV sensors.

RL is also a type of ML that contains two primary components: an agent and an environment. An agent is the sole decision-maker and learner in an RL and interacts with an environment that is a physical world and tries to find the best possible action to increase its performance. To evaluate the performance, a reward function is used. The reward can be positive as well as negative. Hence the agent is self-trained by reward and punishment (negative reward). During each step, the agent receives the state of the environment and chooses an action; after performing the action on the environment, the agent receives a positive or negative (punishment) reward. The objective of the agent is to maximize the total reward value. In order to choose the action, the agent either uses exploration or exploitation. Exploration is the agent will choose a random action from the sample space and gather more information, whereas exploitation utilizes the current information and make the best decision based on that. The main benefit of RL is that it teaches machines what to do by interacting with the environment. Human bias is incorporated into the model in the case of deep learning or supervised learning. Deep reinforcement learning (DRL) is the combination of RL and DL. Deep RL allows the system to be more efficient and robust than a system trained solely through supervised training. The state value function and quality function(Q-value) are used by the RL algorithms. DL is used to estimate the best Q value. Deep Q Network (DQN) and double DQN models are the most used DL method for RL. Along with DQN and double DQN, attention models and generative adversarial network (GAN) DL methods are also used with RL. However, to the best of the authors' knowledge, there is no study that uses a deep RL approach to detect a sophisticated turn-by-turn spoofing attack. The contribution of this study involves the development of the deep RL strategy using a real-world Honda Driving Dataset.

METHOD

The developed GNSS spoofing attack detection framework utilizes a deep RL-based approach and uses data from multiple low-cost in-vehicle sensors of an AV to detect a sophisticated turn-by-turn spoofing attack. This section contains data description and data processing, attack model, generation of turn-by-turn attack dataset, and reinforcement learning-based detection model development subsections.

Dataset Description and Data Processing

This study uses a real-world driving dataset –i.e., the Honda Research Institute Driving Dataset (Honda dataset) [22], to develop an RL-based GNSS spoofing attack detection model. The Honda Dataset includes different vehicle sensor data for suburban and urban driving scenarios, which makes it more appropriate to be used for developing a spoofing attack detection model. Specifically, this dataset is suitable for training an AV GNSS spoofing detection model for the

urban environment in which the AV is more vulnerable. It is a challenge to detect the shift in location in the urban scenario because of compromised GNSS information. The dataset contains data of 104 hours of real human driving in the San Francisco Bay area with a vehicle equipped with AV sensors. Figure 1 presents a sample driving route of the vehicle**.**

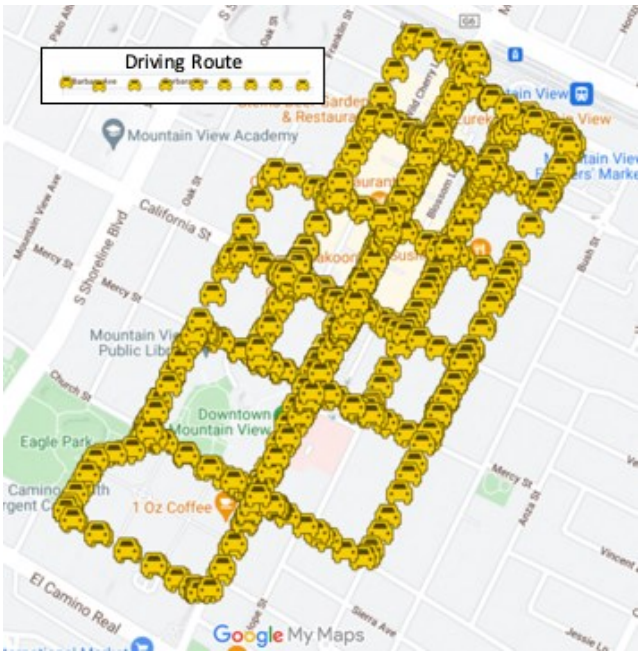


Figure 1. Sample driving routes from Honda Research Institute Driving Dataset

The vehicle was equipped with cameras, LiDAR, GPS, inertial measurement unit (IMU), and control area network (CAN). A GeneSys Eletrinik GmbH Automotive Dynamic Motion Analyzer with DGPS is used for sensing GPS, accelerometer, and gyroscope data at 120 Hz. However, the gyroscope and accelerometer data are not included in the Honda Dataset. The CAN output includes the throttle angle, brake pressure, steering angle, yaw rate, and speed sampled at 100 Hz. The 104 hours driving data includes multiple trips on different routes and on different days. A single-day's driving data is used in this study (route shown in Figure 1). It has been confirmed that the selected dataset successfully mimics an urban traffic scenario. The selected data are further processed. As different sensors have different data collection frequencies (e.g., 100Hz, 120Hz), all the sensors' data are synchronized with the GPS data, using GPS time as the reference time. To obtain steering angle, speed, and brake paddle data at the exact time as GPS reference time, interpolation is performed between the two closest observations in which the GPS observation exists. The synched data are then further resampled at 100 Hz.

The developed deep RL model is trained using GPS and Inertial Measurement Unit (IMU) sensors data. The major advantage of using IMU data over LiDAR point cloud data or camera images/videos is that the size of combined data of GPS and IMU is insignificant (as all are numerical data) compared to camera and LiDAR. Therefore, even if the deep RL model is trained with driving data of multiple days, the storage requirement is significantly low. In addition, a pre-trained deep RL model is deployed inside the vehicle for attack detection. Furthermore, an autonomous vehicle will already have enough storage and computational capability to process all

types of in-vehicle sensor data, including a huge amount of LiDAR and camera data. Thus, the required processing power for deploying the developed method will add minimal cost as it only needs to run the pre-trained deep RL model for attack detection. Moreover, the model is taking input of the sensor's immediate previous timestamp. Thus, no memory is needed to store or buffer sensor data for the prediction model. Note that the presented computational time in the manuscript is calculated by running all the experiments in a workstation equipped with dual Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz processor with 64GB DIMM DDR4 Synchronous 2133 MHz (0.5 ns) RAM memories.

This study uses latitude and longitude information from GPS along with steering wheel angle (deg), speed (ft/s), and relative accelerator pedal position (%) from CAN to generate a dataset for training and testing a deep RL model for detecting GPS spoofing attacks. The GPS latitude and longitude information are used to calculate the distance traveled between two consecutive timestamps using the Haversine formula [23] discovered by James Andrew in 1805, which is universally used to calculate the great circle shortest distance between two pairs of coordinates on a sphere, as shown in Equation 1:

$$d = 2r\,sin^{-1}\left(\sqrt{sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + cos(\varphi_1)\,cos(\varphi_2)\,sin^2\left(\frac{\psi_2 - \psi_1}{2}\right)}\right) \qquad (1)$$

where d is the distance in meters between two points on the Earth's surface; r is the Earth's radius (6378 km); φ_1 and φ_2 are the latitudes in radians; ψ_1 and ψ_2 are the longitudes in radians of two consecutive time stamps. Though the Earth is slightly elliptical, among Haversine, Spherical law of cosines, and Equirectangular approximation methods, the Haversine formula is the most popular method to calculate the shortest distance between two locations on the Earth due to its better accuracy and less computational complexity.

Table 1 presents a sample raw sensor data from Honda Dataset. In addition, Figure 2 presents sensor data that we have used for creating the training and testing dataset to develop a deep RL model.

TABLE 1 Sample Data from Honda Dataset

Unix Timestamp	GPS		Speed (ft/s)	Steering Wheel Angle (deg)	Accelerator Pedal Position (%)
	Latitude	Longitude			
1488224209.42714	37.393	-122.077	0	-57.8	0
1488224209.43716	37.3939	-122.077	0	-57.8	0
1488224209.44696	37.3939	-122.077	0	-57.8	0
1488224209.45698	37.3939	-122.077	0	-57.8	0
1488224209.46698	37.3939	-122.077	0	-57.8	0

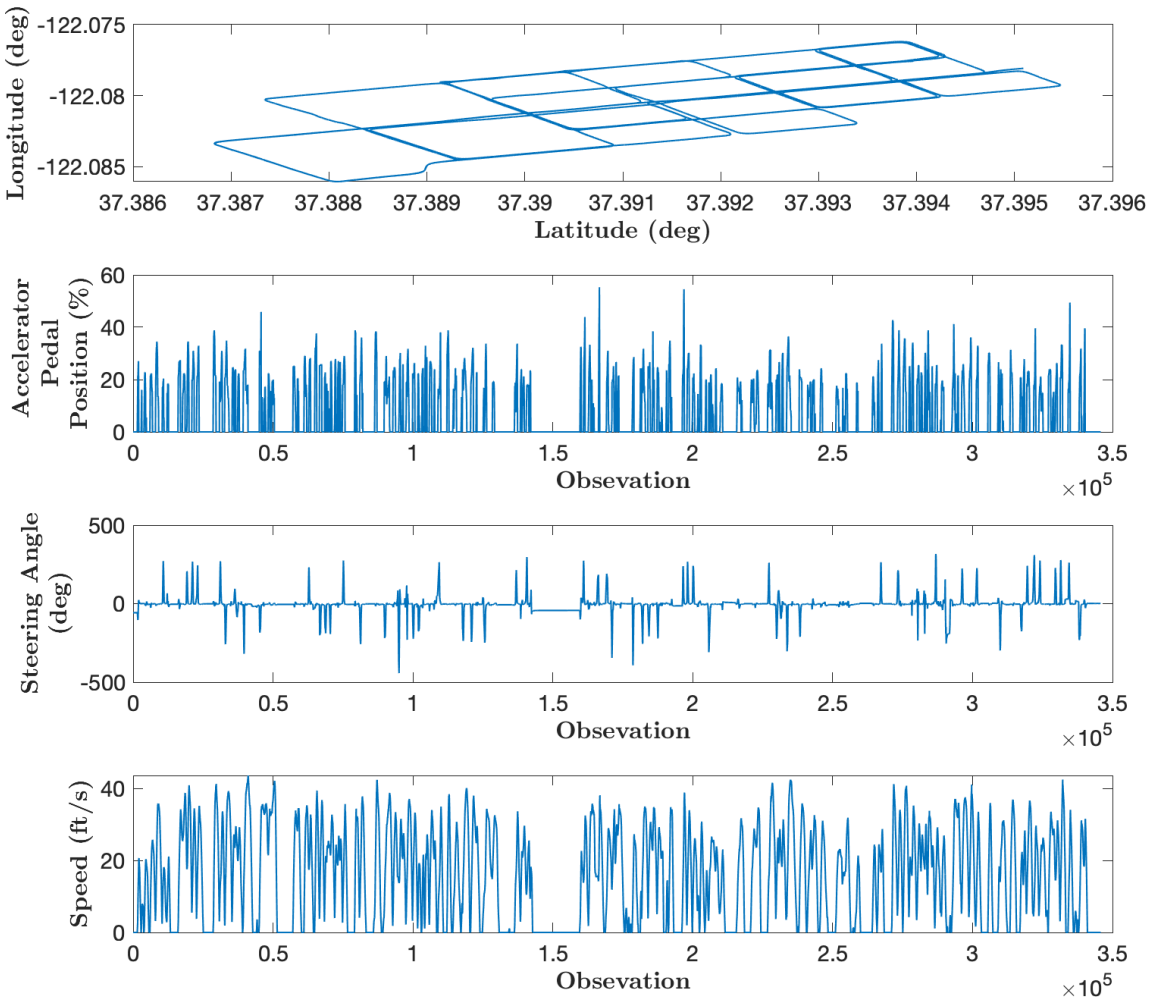


Figure 2 GPS and CAN data from Honda Dataset

Attack Model

In this study, we have created a turn-by-turn [5] spoofing attack, which is a practical and sophisticated GNSS spoofing attack on AV localization and navigation. In order to create such an attack, a spoofer requires the destination and route information of the target AV. During this attack, a spoofer generates wrong GNSS signal and makes the AV GNSS receiver lock onto the spoofed signal. An AV believes the spoofed signal to be an authenticated GNSS signal. After taking control over the GNSS signal, a spoofer creates a spoofed route matching all the turns of the actual route. Thus, it is a challenge to detect such anomalies. An example of such an attack is shown in Figure 3. Here, the blue-colored line is the suggested route created at the beginning of a trip based on the origin and destination information by an AV. When the AV reaches location A, the spoofer takes over the AV GNSS, and it shifts the location from A to B. Due to the change of current location, the navigation application creates another route from B to destination, whereas the AV's actual location is A. Now, the spoofer keeps updating the AV's location in such a way that vehicle will believe that it is moving along the black-dashed route; however, in reality the AV is moving along the orange route. Both the black and orange routes have the same number of turns and the same

type (right or left) of turns. As a result, the AV ends up at a spoofed destination without recognizing that it is under attack.

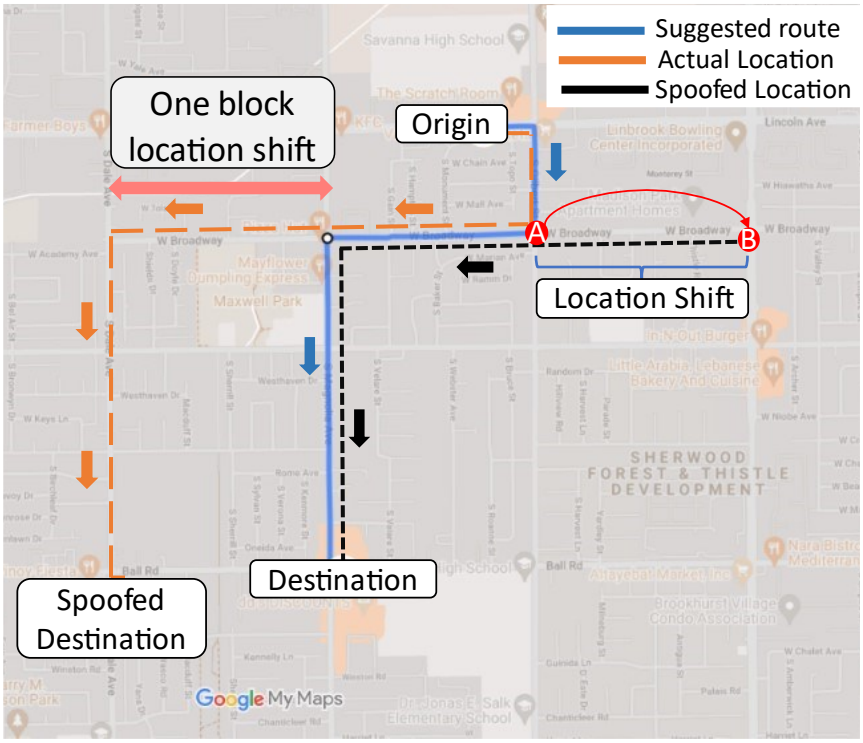


Figure 3 Turn-by-turn GNSS spoofing attack

Generation of Turn-by-Turn Attack Dataset

An attack dataset is created for developing the GNSS spoofing attack detection model. A total of ten different attack datasets are created to train and test data in the RL model. Note that all attack datasets shown in Figure 4 include multiple turn-by-turn attacks. One of the basic features of the turn-by-turn attack is the location shift. In these datasets, the location shifts range from one block to a couple of blocks (50m to 180m) shift of location (See Figure 3). To generate the location shift, a random number generator is used to generate a location shift value between 50m and 180m. Figure 4 presents location shift values for all ten datasets. The X-axis represents in which observation the location shift (or attack) is happening, and the Y-axis represents the value of location shift. Scenario 1 contains the highest number of attacks, and scenario 10 contains the least number of attacks in a single scenario.

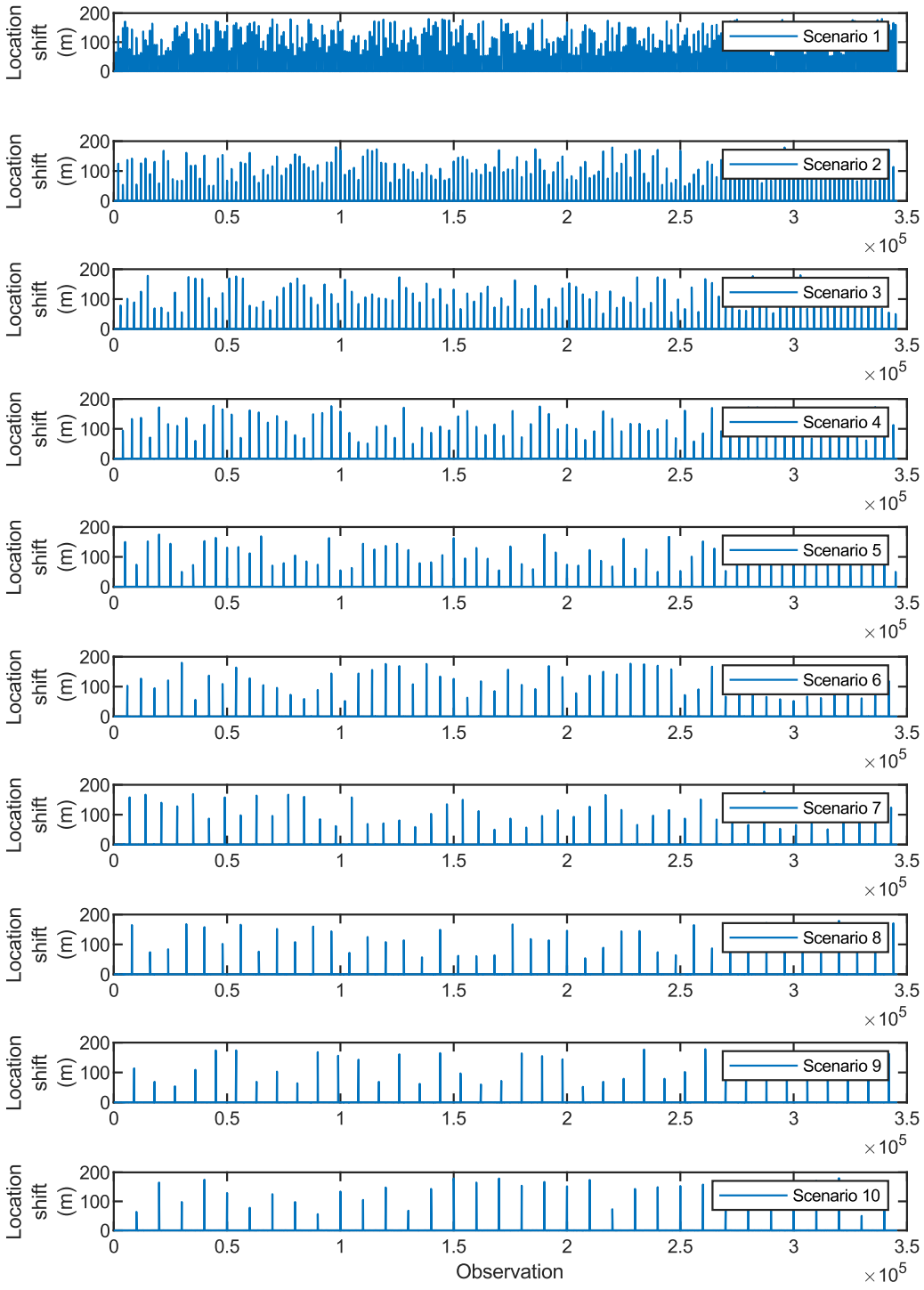


Figure 4 Attack datasets

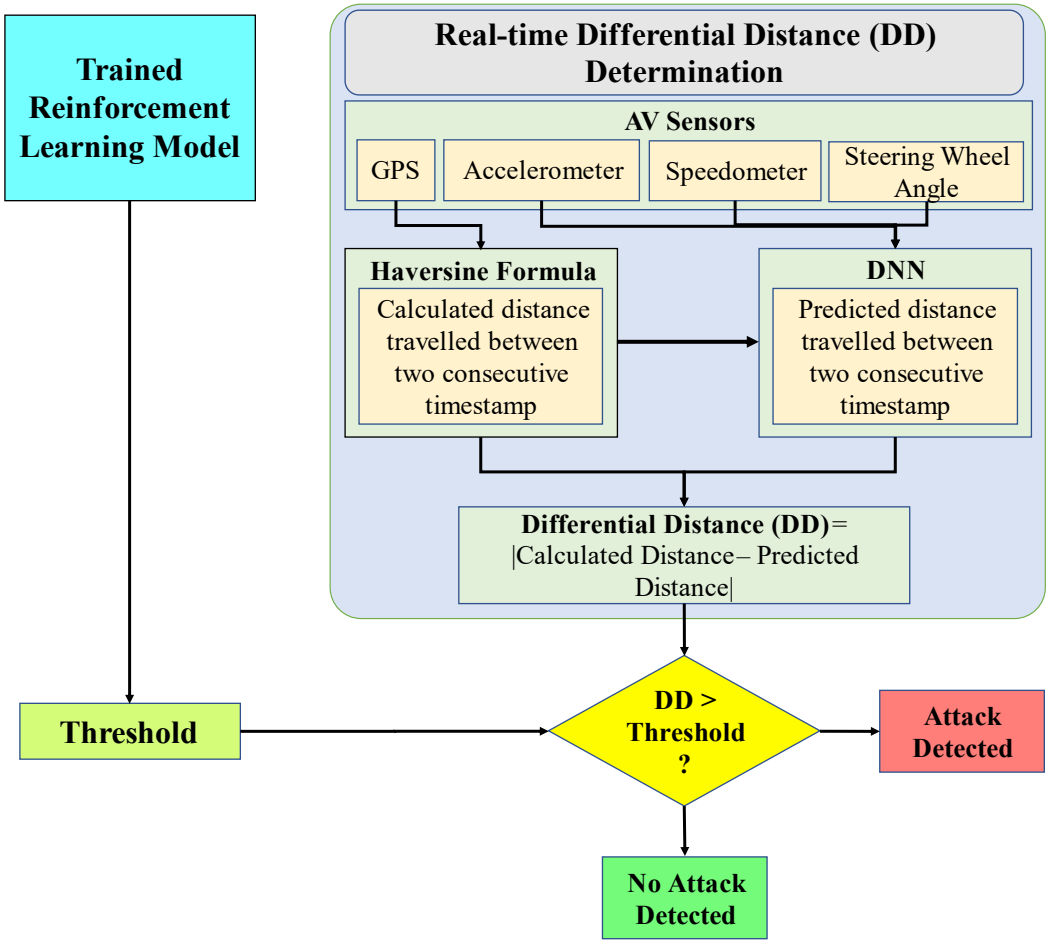
Reinforcement Learning Based Detection Model Development

As shown in Figure 5(a). The threshold value obtained from the trained RL model is compared with the differential distance (DD), which is calculated using real-time GNSS data from an AV. The absolute difference between the predicted and calculated distance is defined as DD. If real-time DD is greater than the threshold, then an attack is detected; otherwise, no attack is detected. Figure 5(b) presents an RL model, which consists of two components: (i) agent and (ii) environment. We have defined the RL problem inside the environment. Here, AV sensor data (GPS and CAN) are used to calculate and predict the distance traveled by an AV between two consecutive timestamps. The agent will adjust the threshold value and compare it so that the reward is maximized.

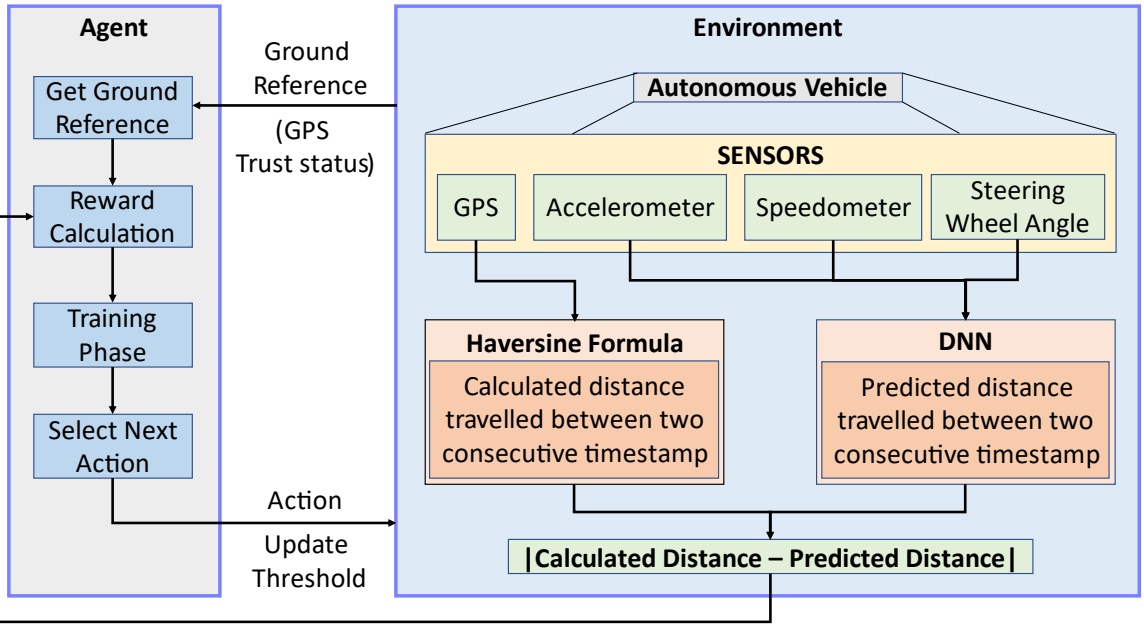
An agent is designed to observe the environment and trained so that it behaves optimally in a given environment state, which results in a partial or complete solution. To achieve the optimal solution, an agent interacts with the environment in discrete time steps. An agent's primary purpose is to choose actions that will maximize the overall future reward. The agent is always modifying its policy in order to discover the optimal one. The proposed RL method is represented in Figure 5 as a flow diagram. In this study, we use the "keras-rl" reinforcement learning framework, which is built on keras, as a base. Note that, TensorFlow backend is used in keras.

In our deep RL framework, the environment consists of an AV equipped with various sensors. The input sensor data consists of latitude, longitude, speed, steering wheel angle, and relative acceleration pedal position (%). We consider that the AVs GNSS receiver is compromised during the spoofing attack; however, CAN output—i.e., speed, steering wheel angle, and relative acceleration pedal position— are not affected by the attacker. Distance traveled by an AV between two consecutive timestamps is calculated using the GNSS coordinates as formulated in Equation 1. We have also predicted the distance traveled by the AV using a deep neural network (DNN) using the CAN sensor data. The training data also include the current GNSS trust status—i.e., whether the GNSS is compromised or not. The DNN predicts the distance traveled by an AV between two consecutive timestamps using CAN and GNSS data.

DNN is an artificial neural network (ANN). It consists of multiple layers of interconnected single or multiple neurons in an input layer and an output layer. Figure 6 depicts the DNN architecture, which is used in this study. The input layer has 4 neurons corresponding to three input data from CAN and one from GNSS, which is the distance traveled between two consecutive timestamps. There are three fully interconnected hidden layers with the number of neurons of 16, 8, and 4. The output layer has one neuron, and the output data is the predicted distance traveled between two consecutive timestamps by an AV. A rectified linear unit (ReLU) is used as the activation function for hidden layers.



(a) Detection framework



(b) Reinforcement learning mode

Figure 5 Deep RL-based turn-by-turn spoofing attack detection framework

The DNN model is trained and validated using uncompromised (Honda Dataset) acceleration, steering wheel angle, speed, and GPS coordinate-based distance traveled data. These raw sensor data are normalized between 0 and 1 prior to training. We have used 241,989 (70%) observations for training and 103,709 (30%) observations for validation. The DNN model hyperparameters—i.e., number of layers, inner layers width, inner layers neurons, epochs, optimizer, loss function, and activation function—for inner layers are selected by a trial-and-error approach. Table 2 provides optimal hyperparameters values of the DNN model. To find a set of optimal hyperparameters, we have used Mean Absolute Error (MAE) metric for the loss function, which helps to identify if there are any model underfitting and overfitting issues. After validating the DNN model, we find that Root Mean Square Error (RMSE) and maximum absolute prediction error for the DNN are 1.18 x 10^{-5} m and 0.07m, respectively. Figure 7(a) presents a comparison between the ground reference and predicted distance traveled (i.e., location shift) for the validation result that shows that the model can predict the distance traveled with a low prediction error, and Figure 7(b) shows the absolute error profile between the ground reference and the predicted data. The profile shows that except for one single observation of an absolute error of 0.07m for all the observations, the absolute error is less than 0.035m. Hence the error in predicting the distance traveled by the DNN is model is low.

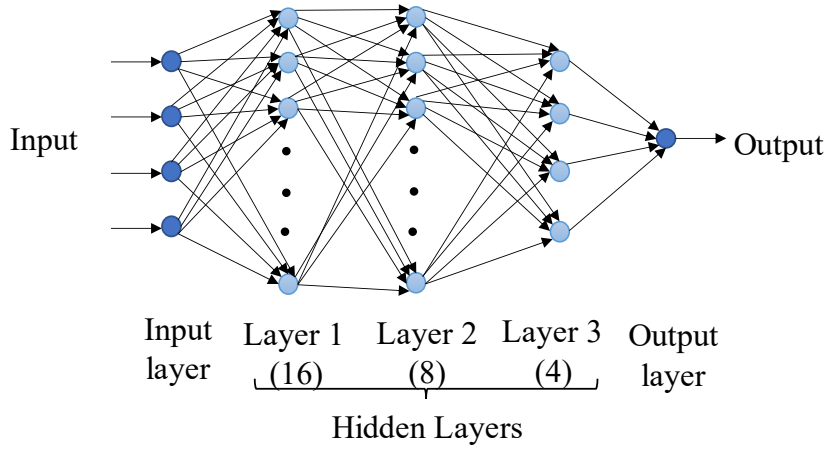
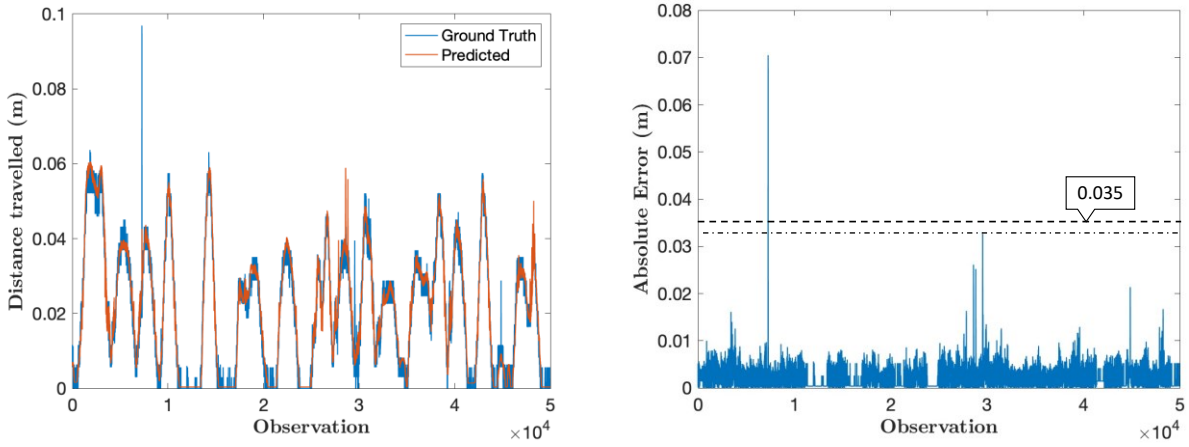


Figure 6. DNN-based distance traveled prediction model architecture

TABLE 2 DNN Model Hyperparameters

Hyperparameters	Value
Number of layers	5
Inner layers width	3
Number of neurons each layer	4, 16, 8, 4, 1
Number of epochs	1000
Optimizer	ADAM

Loss function	Mean Average Error
Activation function	ReLU



(a) (b)

Figure 7 (a) Comparison of ground reference (Ground Truth) data and predicted data; and (b) Absolute error profile

Agent

We have presented an RL architecture comprised of a single-agent system. Deep Q-Networks (DQN) agent in "keras-rl" is used as an agent. There are four steps inside the agent: (i) get the ground reference; (ii) calculate reward; (iii) training phase; and (iv) select next action. After obtaining the ground reference GPS trust status and the differential distance from the environment, an agent calculates the reward function. However, an agent acts based on an optimal policy. The state of this agent is a threshold value, which is defined based on the maximum prediction error for distance traveled between two consecutive timestamps. If the assigned threshold value is lower than the differential distance, it detects that the GPS is compromised. Later, the agent will also check with the ground reference data; if the ground reference matches with the agent detection, a positive reward (+1) is given. On the other hand, if an agent's detection outcome does not match with the ground reference, then a higher negative reward (-100) is given. We assign a much higher negative reward to prioritize detecting an attack.

Deep Q-learning Algorithm

We have used a Deep Q-learning algorithm for the deep RL framework. Q in Q-learning represents quality, and learning represents an objective to choose a policy that will maximize the total reward. The Q-learning function takes random actions outside the current policy and learns the detection policy. For this reason, Q-learning is called an off-policy RL algorithm. It also creates a Q-learning table Q[s,a] (it represents the current state (s) of the environment) and corresponding rewards for each possible action (a), which is Q-value. The algorithm chooses an action (a) with the highest reward, which is called Q-score. The Q-value is defined as formulated in Equation 2.

$$Q(s_t, a_t) = Q(s_t, a_t) + \alpha(r_{t+1} + \gamma\, maxQ(s_{t+1}, a) - Q(s_t, a_t)) \quad (2)$$

where, s_t is the time state, a_t is the action state, α is the learning rate, r is the reward, and γ is the discount factor. Therefore, we have calculated the current Q-values based on the current and next states and actions, learning rate, and discount factor. The objective of the algorithm is to optimize the total reward intake and Q-learning by using the experience from current and next states and actions. The discount factor can take a value between 0 and 1, which regulates the importance of the immediate and future reward.

Figure 8 presents the details of our deep RL framework. Here, a DNN model is also used to approximate the Q-learning function. The input for training the DNN is the predicted DD obtained from the environment. This DNN architecture consists of four layers. The first layer is the input layer, and then there are two fully interconnected hidden layers with 24 neurons in each layer. The last layer is the output layer, which has three neurons. The input neuron accepts the differential distance, and the output layer can give one of the three following outputs: (i) increase threshold; (ii) decrease threshold; and (iii) keep the threshold the same. As described before, we have also used the same "keras-rl" reinforcement learning framework. TensorFlow backend is used in keras, and DQN agent is used in keras-rl. Only data for scenario 2 from the attack dataset is used for the model training. The DNN model parameters are listed in Table 3.

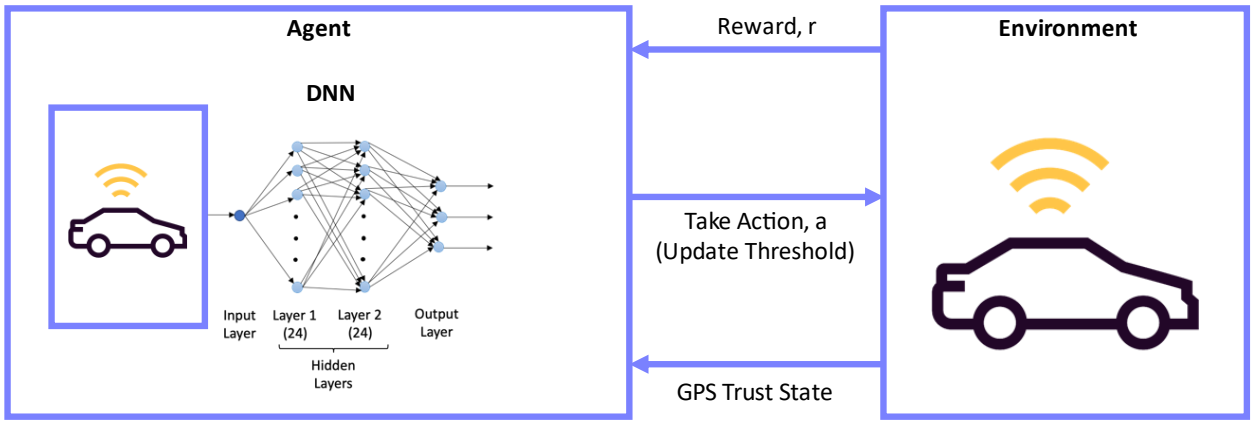


Figure 8. Deep Q-learning Algorithm

Table 3. RL Model Hyperparameters

Hyperparameters	Value
Number of layers	4
Inner layers width	2
Number of epochs	10000
Optimizer	ADAM
Loss function	Mean Average Error
Activation function	ReLU

RESULTS AND DISCUSSION

The performance of the deep RL framework is evaluated based on accuracy, precision, recall, and f1 score metrics using nine test datasets (scenarios 1, 3, 4, 5, 6, 7, 8, 9, 10; see Figure 4). The accuracy, precision, and recall are calculated using the confusion matrix generated from each model considering equal class weights. The precision-recall curve for the developed RL model is presented in Figure 9, which presents the model performance. A high precision value indicates fewer false positives, i.e., fewer cases of detecting a non-attack observation as an attack, and a high recall value indicates fewer false negatives, i.e., fewer cases of failing to detect an attack correctly. So, it can be concluded that the developed deep RL-based spoofing attack detection model performs well as it shows high precision and recall value. Figure 10 presents a grouped stacked bar chart showing the testing results for all nine scenarios. Here, X-axis represents attack scenarios, and Y-axis represents the number of observations. For each scenario, the left stacked bar shows the number of observations for both location shift (Attack) and for those where the GNSS data is not compromised (Attack free). It is evident from the plot that the total number of observations is the same for all the scenarios. The right stacked bar represents the true positive, i.e., successfully detecting an attack, false positive (red), and true negative i.e., correctly detecting a non-attack observation. For all the scenarios, our proposed method successfully detected the attacks. For scenarios 4 and 7, no false positive is detected. It shows that the RL model can successfully detect an attack as soon as the attack is created—i.e., location shift occurred.

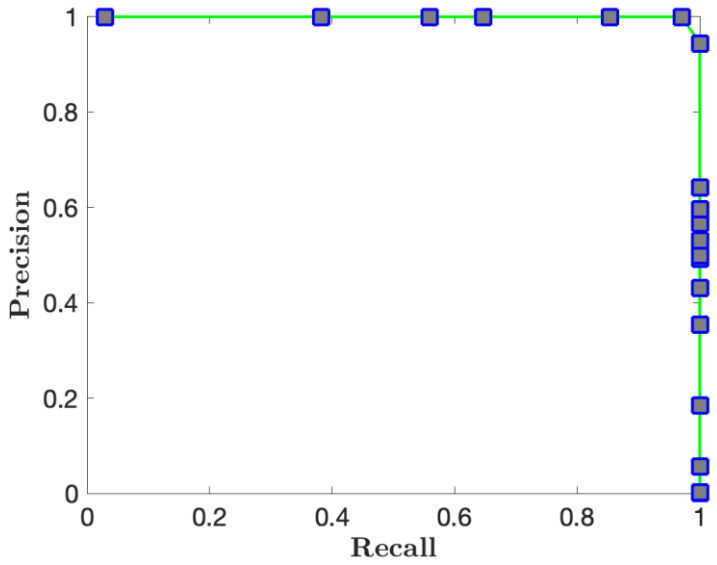


Figure 9 Precision-recall curve for the testing datasets

Figure 10. Detection results

Table 4 provides the recall, precision, accuracy, and f1-score for nine test datasets, which represent nine scenarios. The recall value for all the test cases is 100%, which indicates that the developed model can detect all the attacks. The precision values range from 93.44% (scenario 6) to 100%. 100% precision value means that no false attack was detected. Precision less than 100% means there are instances where the model detected an attack incorrectly—i.e., there is no attack, and an attack is detected. The model achieved the lowest accuracy of 99.99% for seven attack scenarios, and it achieved the highest accuracy of 100% for 2 scenarios. The f1-score ranges from 96.61% to 100%, which means the false positives and false negatives are very low. Based on the performance for all the attack scenarios, we conclude that the deep RL model is effective in detecting GPS turn-by-turn spoofing attacks.

TABLE 4 Model Evaluation Results

Attack Scenario	Recall	Precision	Accuracy	f1-score
1	100%	98.57%	99.99%	99.28%
3	100%	98.29%	99.99%	99.14%
4	100%	100%	100%	100%
5	100%	98.57%	99.99%	99.28%
6	100%	93.44%	99.99%	96.61%
7	100%	100%	100%	100%
8	100%	97.72%	99.99%	98.85%
9	100%	97.43%	99.99%	98.70%
10	100%	94.44%	99.99%	97.14%

Note that the threshold obtained in this study is based on the dataset used to train the model. The training dataset includes different types of left turns, right turns, and lane-change maneuvers, as well as deceleration and acceleration behavior. As the dataset features cover different types of driving maneuvers, it can be concluded that the trained model can handle any real-world driving pattern for any large urban area. Here, the objective of this study is to show the effectiveness of the deep RL method to detect GNSS spoofing attacks. The proposed model can be further trained using data from different types of vehicles (passenger car, bus, heavy trucks) and from various driving scenarios (rural, urban) to increase its robustness.

In this study, we assumed that any in-vehicle sensors (sensor data) other than the GNSS (or GPS) are not compromised. However, in-vehicle sensors' data, which are used in this study, can be altered by an attacker if an AV's in-vehicle network between electronic control units (ECUs) is compromised. Controller area network (CAN), CAN with flexible data-rate (CAN-FD), and Flexray are examples of the existing in-vehicle network[24]. Each of these in-vehicle networks provides an interface for ECUs, which is connected with different sensors, such as GPS, IMU (accelerometer for our study), steering wheel angle sensor, and speedometer. However, CAN, CAN-FD and Felxray are prone to different types of intentional attacks,[25] such as denial of service attack (DoS), impersonation attack, replay attack, amplitude-shift attack [26], [27]. In this way, the security of in-vehicle sensors and data privacy can be compromised. There are three popular strategies to create protection against such security and privacy issues: (i) cryptography, (ii) firewall, and (iii) intrusion detection system (IDS) [28]. Cryptography technique is used for protecting data privacy and security through encryption and decryption of sensitive data, such as vehicle ID and location information of a vehicle. For example, in symmetric key cryptography [29] [30], the same key is used for both encryption and decryption of data, and in asymmetric cryptography,[31] two different keys are used for encryption and decryption of data, which provides additional security. On the other hand, a firewall [32] technique uses multiple layers of security to control the incoming and outgoing data through the in-vehicle network. The intrusion detection system strategies include CAN bus attack detection using signal voltages and inter-signal arrival times at ECUs [33][34][35]. In addition, ML methods are also used utilizing CAN data-frame data to detect anomalies within in-vehicle network [36][37][38][39][40]. Although the security of in-vehicle sensors and data privacy as well as protection strategies against such issues are briefly discussed here, a complete survey of this topic is beyond the scope of this paper.

CONCLUSIONS

An AV must have the capability of real-time detection and feedback-correction of GNSS spoofing attacks related to PNT services, whereby it will help the AV to navigate safely during an attack. This paper developed an RL-based turn-by-turn spoofing attack detection model using low-cost in-vehicle sensor data. In this study, Honda Driving Dataset is used to create ten attack and non-attack datasets, an RL model is developed, and the performance of the RL model is evaluated for GNSS spoofing attack detection. The accuracy of the RL model ranges from 99.99% to 100%, and the recall value is 100%. However, the precision ranges from 93.44% to 100%, and the f1 score ranges from 96.61% to 100%. Thus, the developed RL model detects all the attacks, and there are some instances where attacks are detected incorrectly. Overall, the analyses show that the RL model is effective in turn-by-turn attack detection. In our follow-up study, we will explore the effectiveness of the RL-based approach for other types of sophisticated spoofing attack detection.

ACKNOWLEDGMENTS

This material is based on a study partially supported by the National Science Foundation under Grant No. 2104999. Any opinions, findings, and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, and the U.S. Government assumes no liability for the contents or use thereof. Matthew Shrode Hargis of the University of Alabama provided valuable assistance in editing the paper.

AUTHOR CONTRIBUTIONS

The authors confirm contribution to the paper as follows: study conception and design: S. Dasgupta, T. Ghosh, and M. Rahman; data collection: S. Dasgupta and M. Rahman; interpretation of results: S. Dasgupta, T. Ghosh, and M. Rahman; draft manuscript preparation: S. Dasgupta, T. Ghosh, and M. Rahman. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES

[1] L. Deka, S. M. Khan, M. Chowdhury, and N. Ayres, "Transportation Cyber-Physical System and its importance for future mobility," in *Transportation Cyber-Physical Systems*, Elsevier, 2018, pp. 1–20. doi: 10.1016/B978-0-12-814295-0.00001-0.

[2] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "Prediction-Based GNSS Spoofing Attack Detection for Autonomous Vehicles," *Transportation Research Board*, Oct. 2020, Accessed: Feb. 25, 2021. [Online]. Available: http://arxiv.org/abs/2010.11722

[3] Y. Lu, "Brief Introduction to the GPS and BeiDou Satellite Navigation Systems," Springer, Singapore, 2021, pp. 37–72. doi: 10.1007/978-981-16-1075-2_2.

[4] J. Zidan, E. I. Adegoke, E. Kampert, S. A. Birrell, C. R. Ford, and M. D. Higgins, "GNSS Vulnerabilities and Existing Solutions: A Review of the Literature," *IEEE Access*, pp. 1–1, Feb. 2020, doi: 10.1109/access.2020.2973759.

[5] K. Zeng *et al.*, *All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems*. 2018. Accessed: Jun. 19, 2021. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/zeng

[6] Z. Wu, Y. Zhang, Y. Yang, C. Liang, and R. Liu, "Spoofing and Anti-Spoofing Technologies of Global Navigation Satellite System: A Survey," *IEEE Access*, vol. 8, pp. 165444–165496, Sep. 2020, doi: 10.1109/access.2020.3022294.

[7] B. W. O'Hanlon, M. L. Psiaki, J. A. Bhatti, D. P. Shepard, and T. E. Humphreys, "Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals," *NAVIGATION, Journal of the Institute of Navigation*, vol. 60, no. 4, pp. 267–278, Dec. 2013, Accessed: Jun. 15, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=j&articleID=102607

[8] B. W. O'Hanlon, M. L. Psiaki, T. E. Humphreys, and J. A. Bhatti, "Real-Time Spoofing Detection Using Correlation Between two Civil GPS Receiver." pp. 3584–3590, Sep. 21, 2012. Accessed: Jun. 15, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=10533

[9] B. W. O'Hanlon, M. L. Psiaki, T. E. Humphreys, and J. A. Bhatti, "Real-Time Spoofing Detection in a Narrow-Band Civil GPS Receiver." pp. 2211–2220, Sep. 24, 2010. Accessed: Jun. 15, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=9335

[10] J. Yang, Y. J. Chen, W. Trappe, and J. Cheng, "Detection and localization of multiple spoofing attackers in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 44–58, 2013, doi: 10.1109/TPDS.2012.104.

[11] S. Daneshmand, A. Jafarnia-Jahromi, A. Broumandon, and G. Lachapelle, "A Low-Complexity GPS Anti-Spoofing Method Using a Multi-Antenna Array." pp. 1233–1243, Sep. 21, 2012. Accessed: Jun. 14, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=10336

[12] A. Neish, S. Lo, Y. H. Chen, and P. Enge, "Uncoupled accelerometer based GNSS spoof detection for automobiles using statistic and wavelet based tests," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*, Sep. 2018, pp. 2938–2962. doi: 10.33012/2018.15903.

[13] S. Manickam and K. O'Keefe, "Using Tactical and MEMS Grade INS to Protect Against GNSS Spoofing in Automotive Applications," *Proceedings of ION GNSS+2016*, 2016.

[14] C. Tanil, S. Khanafseh, and B. Pervan, "An INS monitor against GNSS spoofing attacks during GBAS and SBAS-assisted aircraft landing approaches," in *29th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2016*, Sep. 2016, vol. 4, pp. 2981–2990. doi: 10.33012/2016.14779.

[15] M. Sun, Y. Qin, J. Bao, and X. Yu, "GPS Spoofing Detection Based on Decision Fusion with a K-out-of-N Rule," *International Journal of Network Security*, vol. 19, no. 5, pp. 670–674, 2017, doi: 10.6633/IJNS.201709.19(5).03.

[16] Ç. Tanıl, P. M. Jimenez, M. Raveloharison, B. Kujur, S. Khanafseh, and B. Pervan, "Experimental validation of INS monitor against GNSS spoofing," in *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*, Sep. 2018, pp. 2923–2937. doi: 10.33012/2018.15902.

[17] M. L. Psiaki, B. W. OHanlon, S. P. Powell, J. A. Bhatti, K. D. Wesson, and T. E. H. A. Schofield, "GNSS Spoofing Detection Using Two-Antenna Differential Carrier Phase." pp. 2776–2800, Sep. 12, 2014. Accessed: Jun. 15, 2021. [Online]. Available: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=12530

[18] G. Panice *et al.*, "A SVM-based detection approach for GPS spoofing attacks to UAV," Oct. 2017. doi: 10.23919/IConAC.2017.8081999.

[19] K. Yang, S. Kpotufe, and N. Feamster, "AN EFFICIENT ONE-CLASS SVM FOR ANOMALY DETECTION IN THE INTERNET OF THINGS".

[20] P. Borhani-Darian, H. Li, P. Wu, … P. C.-M. of the S. D. of, and undefined 2020, "Deep Neural Network Approach to Detect GNSS Spoofing Attacks," *ion.org*, Accessed: Jun. 14, 2021. [Online]. Available: https://www.ion.org/publications/abstract.cfm?articleID=17537

[21] E. Shafiee, M. R. Mosavi, and M. Moazedi, "Detection of Spoofing Attack using Machine Learning based on Multi-Layer Neural Network in Single-Frequency GPS Receivers," *Journal of Navigation*, vol. 71, no. 1, pp. 169–188, Jan. 2018, doi: 10.1017/S0373463317000558.

[22] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, "Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7699–7707, Nov. 2018, Accessed: Jan. 31, 2021. [Online]. Available: http://arxiv.org/abs/1811.02307

[23] C. C. Robusto, "The Cosine-Haversine Formula," *The American Mathematical Monthly*, vol. 64, no. 1, p. 38, Jan. 1957, doi: 10.2307/2309088.

[24] R. N. Charette, "This car runs on code," *EEE Spectrum: Technology, Engineering, and Science News*, vol. 46, no. 3, p. 3, 2009.

[25] S. Woo, H. J. Jo, and D. H. Lee, "A Practical Wireless Attack on the Connected Car and Security Protocol for In-Vehicle CAN," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, Apr. 2015, doi: 10.1109/TITS.2014.2351612.

[26] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A novel intrusion detection system for in-vehicle network by using remote frame," *Proceedings - 2017 15th Annual Conference on Privacy, Security and Trust, PST 2017*, pp. 57–66, Sep. 2018, doi: 10.1109/PST.2017.00017.

[27] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 130–139, Dec. 2016, doi: 10.1109/DSAA.2016.20.

[28] F. Fakhfakh, M. Tounsi, and M. Mosbah, "Cybersecurity attacks on CAN bus based vehicles: a review and open challenges," *Library Hi Tech*, 2021, doi: 10.1108/LHT-01-2021-0013/FULL/PDF.

[29] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. Liu, "LEAP: A Lightweight Encryption and Authentication Protocol for In-Vehicle Communications," *2019 IEEE Intelligent Transportation Systems Conference, ITSC 2019*, pp. 1158–1164, Oct. 2019, doi: 10.1109/ITSC.2019.8917500.

[30] M. Lakshmanan and S. Kumar NATARAJAN, "Security enhancement in In-vehicle Controller Area Networks by Electronic Control Unit authentication," *ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY*, vol. 22, no. 4, pp. 228–243, 2019.

[31] P. Mundhenk *et al.*, "Security in Automotive Networks," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 2, Mar. 2017, doi: 10.1145/2960407.

[32] S. Rizvi, J. Willett, D. Perino, T. Vasbinder, and S. Marasco, "Protecting an Automobile Network Using Distributed Firewall System," *Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing*, 2017, doi: 10.1145/3018896.

[33] J. Ning, J. Wang, J. Liu, and N. Kato, "Attacker Identification and Intrusion Detection for In-Vehicle Networks," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1927–1930, Nov. 2019, doi: 10.1109/LCOMM.2019.2937097.

[34] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of CAN bus signal injection attacks * A data-driven approach to

in-vehicle intrusion detection," *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, 2017, doi: 10.1145/3064814.

[35] A. Gazdag, D. Neubrandt, L. Buttyán, and Z. Szalay, "Detection of Injection Attacks in Compressed CAN Traffic Logs," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11552 LNCS, pp. 111–124, Sep. 2018, doi: 10.1007/978-3-030-16874-2_8.

[36] M. J. Kang and J. W. Kang, "Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security," *PLOS ONE*, vol. 11, no. 6, p. e0155781, Jun. 2016, doi: 10.1371/JOURNAL.PONE.0155781.

[37] X. Mo, P. Chen, J. Wang, and C. Wang, "Anomaly Detection of Vehicle CAN Network Based on Message Content," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 284, pp. 96–104, Apr. 2019, doi: 10.1007/978-3-030-21373-2_9.

[38] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," *Vehicular Communications*, vol. 21, p. 100198, Jan. 2020, doi: 10.1016/J.VEHCOM.2019.100198.

[39] Z. Khan, M. Chowdhury, M. Islam, C. Y. Huang, and M. Rahman, "Long Short-Term Memory Neural Network-Based Attack Detection Model for In-Vehicle Network Security," *IEEE Sensors Letters*, vol. 4, no. 6, Jun. 2020, doi: 10.1109/LSENS.2020.2993522.

[40] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185489–185502, 2020, doi: 10.1109/ACCESS.2020.3029307.