A Sensor Fusion-based GNSS Spoofing Attack Detection Framework for Autonomous Vehicles

Sagar Dasgupta, *Student Member, IEEE,* Mizanur Rahman, *Member, IEEE,* Mhafuzul Islam, and Mashrur Chowdhury, *Senior Member, IEEE*

***Abstract*— This paper presents a sensor fusion-based Global Navigation Satellite System (GNSS) spoofing attack detection framework for autonomous vehicles (AVs) that consists of two strategies: (i) comparison between predicted location shift—i.e., distance traveled between two consecutive timestamps—and inertial sensor based location shift in addition to monitoring of vehicle motion states—i.e., standstill/ in motion; and (ii) detection and classification of turns (left or right) along with detection of vehicle motion states. In the first strategy, data from low-cost in-vehicle inertial sensors—i.e., speedometer, accelerometer, and steering angle sensor—are fused and fed to a long short-term memory (LSTM) algorithm to predict the distance an AV will travel between two consecutive timestamps. The second strategy combines k-Nearest Neighbors (k-NN) and Dynamic Time Warping (DTW) algorithms to detect a turn and then classify left and right turns using steering angle sensor output. In both strategies, the GNSS-derived speed is compared with speedometer output to improve the effectiveness of the framework presented in this paper. To prove the efficacy of the sensor fusion-based attack detection framework, attack datasets are created for four unique spoofing attack scenarios—turn-by-turn, overshoot, wrong turn, and stop, using the publicly available real-world Honda Research Institute Driving Dataset (HDD). Analyses conducted in this study reveal that the sensor fusion-based detection framework successfully detects all four types of spoofing attacks within the required computational latency threshold.**

***Index Terms*— Global Navigation Satellite System (GNSS), Autonomous vehicle, Cybersecurity, Spoofing attack, LSTM.**

I. INTRODUCTION

AUTONOMOUS vehicles (AVs) require accurate, reliable, and continuous real-time localization information from a Global Navigation Satellite System (GNSS), which is known as Global Positioning System (GPS) in the United States, to perform their autonomous, navigational, security, and safety-critical applications. The term "GNSS" refers to a positioning, navigation, and timing (PNT) service based on satellites. In the United States, two levels of services are provided by GPS, i) Standard Positioning Service (SPS) and ii) Precision Positioning Service (PPS). While the SPS service is available for civil, commercial, and scientific use, a more secure PPS service is only used by the government and the military [1]. Current AVs use SPS signals. It is a reasonable assumption that commercial AVs will continue to use SPS signals in the future, as PPS is not accessible for commercial use. However, SPS has a positioning error at the meter level, and the lowest error for an SPS system is 1m. Thus, SPS alone is not suitable for an AV. For navigation purposes, an AV requires more precise location information— i.e., the positioning error should be much less than that of current SPS. Currently, an AV uses a differential global positioning system (DGPS) to correct SPS localization. In DGPS, the position error can be as low as 0.01m [2]. It corrects the position using a reference station on top of the regular GPS signal. In the rest of the paper, if we refer GNSS or GPS, we mean DGPS.

As the GNSS depends on satellites and radio communications, GNSS signal strength is subject to physical degradation due to natural or unintentional vulnerabilities, and intentional threats [3]. By the time GNSS signals reach AV's GNSS receiver end after passing through the Earth's atmosphere the signal strength deteriorates and the signal strength further degrades by reflected by the high-rise buildings that introduce position inaccuracy and interfering with continuous GNSS signal availability [4] [5]. The GNSS signal can also become disrupted due to natural or unintentional vulnerabilities, such as the absence of GNSS signal because of walls and ceilings in garages and tunnels, and signal degradation due to multipath and radio frequency interference [6], [7]. Jamming and spoofing are the two categories of intentional threats. In jamming, a high-power GNSS signal is transmitted to prevent an authentic signal to reach the target GNSS receiver. Among the foregoing, spoofing is the most sophisticated type of attack, as an attacker can tamper the authentic GNSS signal structure and transmit inaccurate location information to a target AV. A target AV trust the manipulated signal and update its navigation route based on spoofed signal [8]. To a sophisticated GNSS spoofing attack, a spoofer requires a target vehicle's destination, route, and sensor information. In this study, we only consider sophisticated spoofing attacks.

One of the primary purposes of manipulating a GNSS receiver during a spoofing attack is to tamper the GNSS signal so that a target AV is potentially misdirected to the wrong destination, compromising the safety and security of AV passengers as well as the transportation of goods. Generally, an expensive GNSS signal generator is required to perform a spoofing attack. However, the development of low-cost software-defined radios (SDR) has made GNSS spoofing

Manuscript received June 14, 2021.

S. Dasgupta is with the Department of Civil, Construction, and Environmental Engineering at the University of Alabama, Tuscaloosa, AL 35487, USA (e-mail: sdasgupta@crimson.ua.edu).

M. Rahman is with the Department of Civil, Construction, and Environmental Engineering at the University of Alabama, Tuscaloosa, AL 35487, USA (e-mail: mizan.rahman@ua.edu).

M. Islam is with the General Motors. Michigan, USA (e-mail: mdmhafi@g.clemson.edu).

And M. Chowdhury is with the Glenn Department of Civil Engineering, Clemson University, Clemson, SC 29634 USA (e-mail: mac@clemson.edu).

attacks easier to carry out [9]. Unfortunately, due to the dynamic nature of spoofing attacks, there is no single attack detection method that can detect all types of GNSS spoofing attacks [10]. Therefore, researchers are concentrating more on developing methods that make a spoofing attack more difficult to achieve.

In this study, a sensor fusion based GNSS spoofing attack detection framework, which consists of two strategies, is developed and evaluated to detect GNSS spoofing attacks. Motivated by the resilience of the inertial sensors against spoofing attacks and robustness of deep learning (DL) algorithms, in the first strategy, data from low-cost in-vehicle sensors—i.e., speedometer, accelerometer, and steering angle sensor—which are not only available in commercial AVs but also in the human-driven vehicles, are fused and fed to a long short-term memory (LSTM) algorithm to predict the distance an AV will travel between two consecutive timestamps. This predicted distance is then compared with the GNSS-derived distance to detect an attack. Along with predicting distance, an GNSS-based motion state is compared with AV's speedometer data. If they do not match with one another, an attack is detected. The second strategy includes detection and classification of left and right turns in addition to the AV's motion state. An AV's turning maneuvers and motion states using in-vehicle sensors are then compared with corresponding GNSS derived data to further detect an attack, which makes the framework more robust. Note that both strategies run simultaneously. The GNSS spoofing attack detection framework developed in this paper is targeted at AVs that navigate through a structured roadway (e.g., surface roadway network).

The rest of the paper is arranged as follows. Section II discusses the contribution of this paper. Section III reviews existing spoofing attack detection methods and identifies the research gap. Section IV introduces a real-world dataset, which is used to create the attack dataset, and presents the data preparation approach. GNSS spoofing attack models are presented in section V. Section VI presents our GNSS spoofing attack detection framework, the attack detection efficacy against different spoofing attack scenarios, and a comparison of attack detection performance between a baseline framework and our framework. After that, computational complexities of our framework are described in section VII. Finally, Section VIII presents the concluding remarks and future research direction.

II. CONTRIBUTIONS OF THIS STUDY

The presented framework in this paper is a new addition to the existing GNSS spoofing attack detection approaches because the detection framework can be deployed for different road navigation scenarios that use the location level information without directly analyzing the GNSS signal characteristics, whereas most of the existing approaches detect a spoofing attack by analyzing the GNSS signal itself. We have developed a single robust GNSS spoofing attack detection framework, which consists of two strategies: (i) comparison between predicted location shift—i.e., distance traveled between two consecutive timestamps—and inertial sensor based location shift in addition to monitoring of vehicle motion states—i.e., standstill/in-motion; and (ii) detection and classification of turns (left or right) along with detection of vehicle motion states. Our study differs from the existing studies as we have used the speedometer, steering wheel angle, and accelerometer data to predict the location shift using an artificial recurrent neural network (RNN) architecture, i.e., LSTM. The predicted location shift of an AV is then compared with the GNSS-based location shift to detect a spoofing attack. Moreover, a turn detection strategy is used to further detect more sophisticated attacks, such as a wrong turn attack. The primary contribution of this study is that none of the existing research uses predicted location shift and turn detection techniques for developing sophisticated spoofing attack detection framework utilizing GNSS information (latitude, longitude, and turn type) and in-vehicle sensor data without analyzing the GNSS physical signal characteristics. Although existing INS/IMU-based GNSS spoofing attack detection approaches derive vehicle position or compare a single in-vehicle sensor output with GNSS output, the framework presented in this study fuses multiple sensors and uses an LSTM network to predict a vehicle's location shift, which has not been explored by any researchers so far.

III. RELATED WORK

The techniques of existing spoofing attack detection methods can be classified into four categories: (i) encryption mechanisms; (ii) codeless-cross-correlation measures; (iii) signal statistics analyses; and (iv) antenna-based methods [3]. The most common GNSS anti-spoofing methods use encryption algorithms to secure the GNSS signals. Although the military commonly uses the encryption mechanism approach to secure GNSS receivers, this is not a cost-effective solution due to its high infrastructural, computational, and management cost. The codeless-cross-correlation measures use the correlation among unknown encrypted GPS L1 P(Y) code signal from multiple receivers to detect a spoofing attack [19], [20]. Note that L1 is the primary GPS carrier signal, and P(Y) code is the precision or secure code. The effectiveness of such a method also depends on the cost of new instruments and associated signal processing complexity when the number of cross-checking GNSS receivers is increased. The GNSS signal statistics analysis-based approaches use different signal features, such as received signal strength (RSS) [21], spatial coherency [22], pseudo-range measurements, time of advent, and signal parameters estimation to detect GNSS spoofing attacks. Antenna-based methods include detecting a spoofing attack by using multi-antenna GNSS, reduced inertial sensor system (RISS), and inertial navigation system (INS) integration [23] to perform beat carrier-phase measurement processing using two antennas [24]. A single antenna combined with RISS can also be used to detect spoofing attacks [25]. These approaches require computationally expensive GNSS signal processing algorithms and sophisticated antenna arrays to ensure high spoofing attack detection accuracy [3], [26].

Besides these approaches, GNSS spoofing attacks can also be detected by comparing vehicle acceleration from IMU with the GNSS derived acceleration according to [10]. Although this approach performs well for an aircraft, it is not suitable for surface vehicles due to the low vehicle dynamics signature. In [12], the location information derived from IMU sensors (i.e., accelerometer and gyroscope) is compared with the GNSS-

derived location for spoofing attack detection. Furthermore, inertial navigation system (INS) has also been used to monitor the position of a vehicle for detecting GNSS spoofing attacks [11][12]. INS devices use gyroscope and accelerometer data and calculate the position, orientation and speed of a vehicle using dead reckoning without any input from GNSS. However, INS derived location is less accurate as the measurements from inertial sensors accumulate bias, scale factor, and non-orthogonality errors over time. In addition, multiple antennas are used to identify spoofing attacks through cross-checking GNSS signals [24].

In addition to the above-mentioned approaches, the development of machine learning (ML) and DL algorithms has recently increased for spoofing attack detection. In [14], a Multi-Layer Perceptron (MLP), a Complex Convolution Neural Networks (CNN), and a simple CNN are used to detect spoofed GNSS signals that demonstrate the potency of using deep neural networks for spoofed signal detection. In [16], the authors provide a decision fusion with the K-out-of-N decision rule-based method along with wavelet transformation coefficients. In [15], a Support Vector Machine (SVM) has been used for state estimation and detecting an attack on unmanned aerial vehicles based on it. The early-late phase, delta, and signal level are used as features together with the K-Nearest Neighbor (KNN) and naïve Bayesian classifier to detect spoofing attacks [17]. However, these ML and DL algorithms are used to detect an attack in the signal level, and no research has been conducted to predict the distance a vehicle can travel within a timeframe and detect an attack based on that. Only INS-based spoofing attack detection approaches, where an INS-based vehicle's position was compared with the GNSS-based position, are closely related to our study. However, position information from INS sensors is not reliable due to the error propagation of inertial sensors over time. Thus, none of the existing approaches used the location domain information to detect the spoofing attack.

Several existing companies [27]–[32] offer commercial GNSS jamming and spoofing detection services. These companies utilize proprietary algorithms to detect anomalies in the received GNSS signal. These attack detection technologies either use costly local precise atomic clock or the results of analysis of GNSS signal characteristics for spoofing attack detection. Thus, the attack detection framework, which is presented in this paper, is different from existing solutions as we use low-cost in-vehicle sensors for GNSS spoofing attack detection. In addition, detailed detection efficacy results of the commercial solutions are not available in the public domain. Therefore, it is unknown how these commercial technologies perform when used exclusively for an AV's navigation protection against GNSS spoofing and jamming attacks in a dynamic and mobile roadway traffic environment.

Our study differs from the previous studies as we have used the speedometer, steering wheel angle data, and accelerometer data to predict the location shift using an LSTM network. The predicted location shift by the A.V. is then compared with the GNSS-based location shift to detect any spoofing attack. Moreover, a turn detection strategy is used to further detect more sophisticated attacks such as wrong turn attacks. Our study is novel because none of the existing research uses predicted location shift and turn detection techniques for developing sophisticated spoofing attack detection framework utilizing GNSS information (latitude, longitude, and turn type) and in-vehicle sensor data without analyzing the GNSS physical signal characteristics.

IV. Data Preparation

The Honda Research Institute Driving Dataset (HDD) [33] is used in this study to develop and evaluate the GNSS attack detection framework. The HDD contains data from the camera, LiDAR, GNSS, inertial measurement unit (IMU), and controller area network (CAN) of a conventional vehicle, and it is collected from suburban and urban roadways as well as highways within the San Francisco Bay Area. As AVs are equipped with cameras, IMU, and GNSS, the HDD is suitable for generating attack datasets for an AV technologies for AVs. Figure 1 shows a sample route from the HDD.

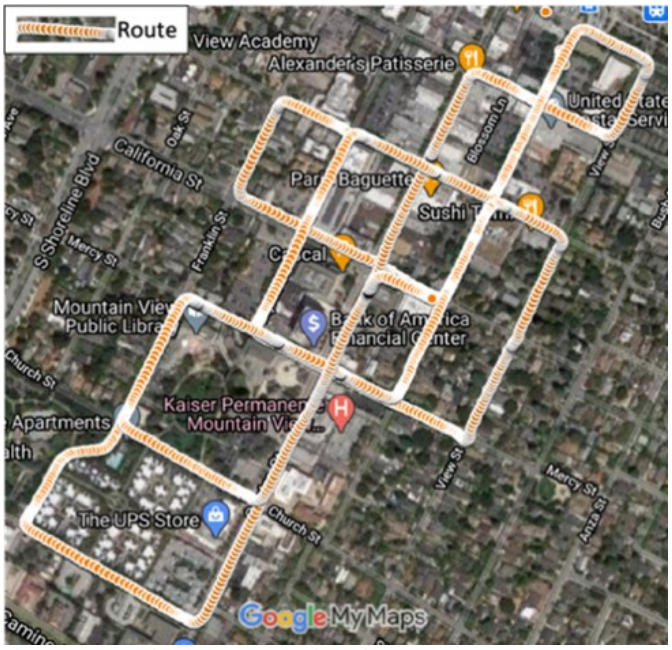


Fig. 1. An example of GNSS traces from the HDD.

Using an Automotive Dynamic Motion Analyzer (ADMA) from GeneSys Eletronik GmbH and a DGPS, the GNSS signals are recorded in HDD at a rate of 120Hz. The acceleration (m/s^2), steering wheel angle (deg), rotational speed of the steering wheel (deg/s), vehicle speed (ft/s), brake pressure (kPa), and yaw rate (deg/s) are collected from different sensors and recorded from vehicle's CAN bus at 100Hz. For our detection framework development and evaluation, the latitude and longitude, relative accelerator pedal position (%), steering wheel angle, and speed data are extracted from the HDD. Note that relative accelerator pedal position (%) represents the acceleration of an AV. We have then synchronized the extracted HDD by keeping GNSS UNIX timestamp as a reference by interpolating between the two closest observations. We have also calculated the perceived location shift, i.e., the distance traveled between two consecutive timestamps, with data from GNSS using the Haversine formula (see equation (1)) [34]:

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

where d is the distance in meter 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; and ψ_1 and ψ_2 are the longitudes in radians of two consecutive time stamps.

V. Attack Models

The developed GNSS spoofing attack detection framework is evaluated against four sophisticated spoofing attack scenarios, which are: (a) turn-by-turn attack; (b) overshoot attack; (c) wrong turn attack; and (d) stop attack. A turn-by-turn attack is a sophisticated type of spoofing attack because the spoofer has an AV's route and destination information which allows the spoofer to manipulate the GNSS signal in such a way that it is very hard to detect any change in the route. In this type of spoofing attack [35], a spoofer takes over a target AV's GNSS receiver and changes the AV's current location, resulting in a location shift between an AV's location before and after the attack. Due to a change in an AV's current location, the navigation application creates a new route to reach the destination, and an AV believes in the spoofed location, follows the newly created wrong route, and ends up in a wrong, possibly unsafe location instead of the desired destination. As a spoofer also try to make a realistic location shift, it is not possible for an AV to find any difference in speed and distance between actual and spoofed routes, which make this attack more believable.

Figure 2(a) illustrates a turn-by-turn attack in which an actual route after attack from an origin to a destination is shown in blue, an AV's ground truth route is shown in green, and the AV's perceived route, which matches the original route turn-by-turn, is shown in red. Thus, a spoofer creates a wrong route matching the new route's number of turns and guides the vehicle to a wrong destination by compromising the AV's GNSS receiver.

Figure 2(b) shows an overshoot attack [36] where after taking over an AV's GNSS receiver, a spoofer keeps sending the same location information. As a result, based on the GNSS output, an AV perceives that itself in a standstill state (shown as stopped at the red dot), although the AV is moving forward in reality. When a road split (at the green dot) or intersection occurs, the AV will be unable to identify the path to proceed.

During a wrong turn attack, a spoofer takes over a target AV's GNSS receiver just before a turn. While a target AV takes a right turn, as shown in Figure 2(c), a spoofer tampers the GNSS signal in a way so that the target AV perceives that it is taking a left turn. Similarly, if an AV takes a left turn, the GNSS will show a right turn. This will lead to the rerouting of the attacked AV, and the target AV will arrive at the wrong destination.

A stop attack [36] (see Figure 2(d)) is the opposite of an overshoot attack. A spoofer takes over the GNSS receiver when an AV is stopped at a stop sign (green dot) or in a queue due to traffic and then transmits a synthetic GNSS signal so that the corresponding AV perceives that it is moving along a road (shown as red route).

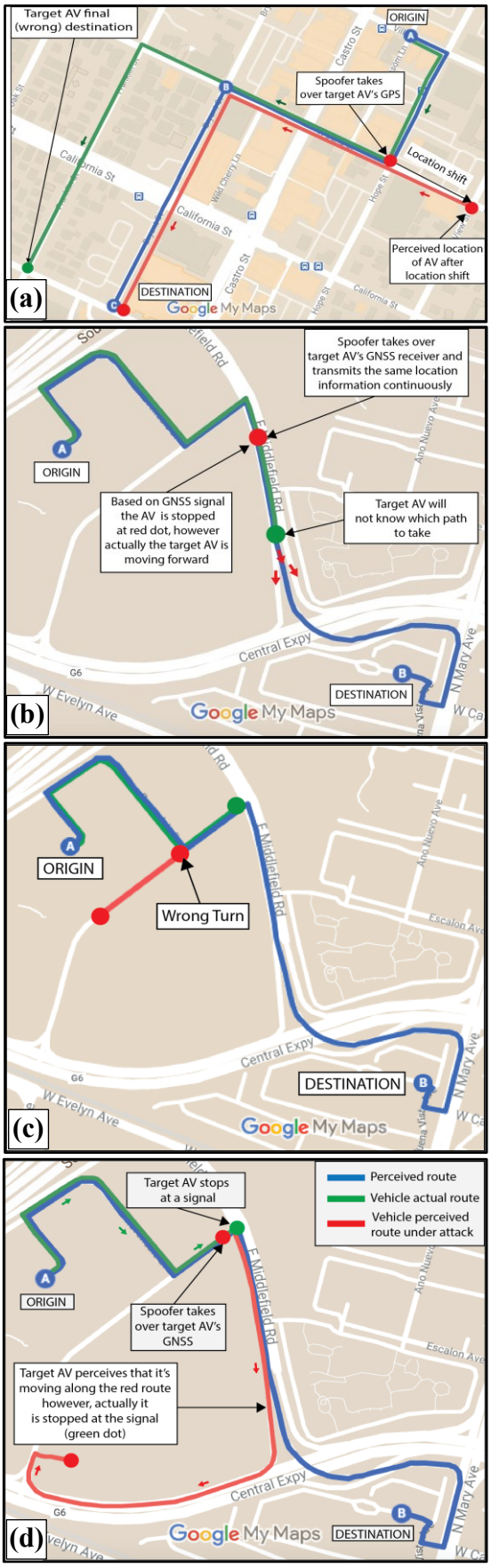


Fig. 2. GNSS spoofing attack models: (a) example of turn-by-turn; (b) example of overshoot; (c) example of a wrong turn, and (d) example of a stop.

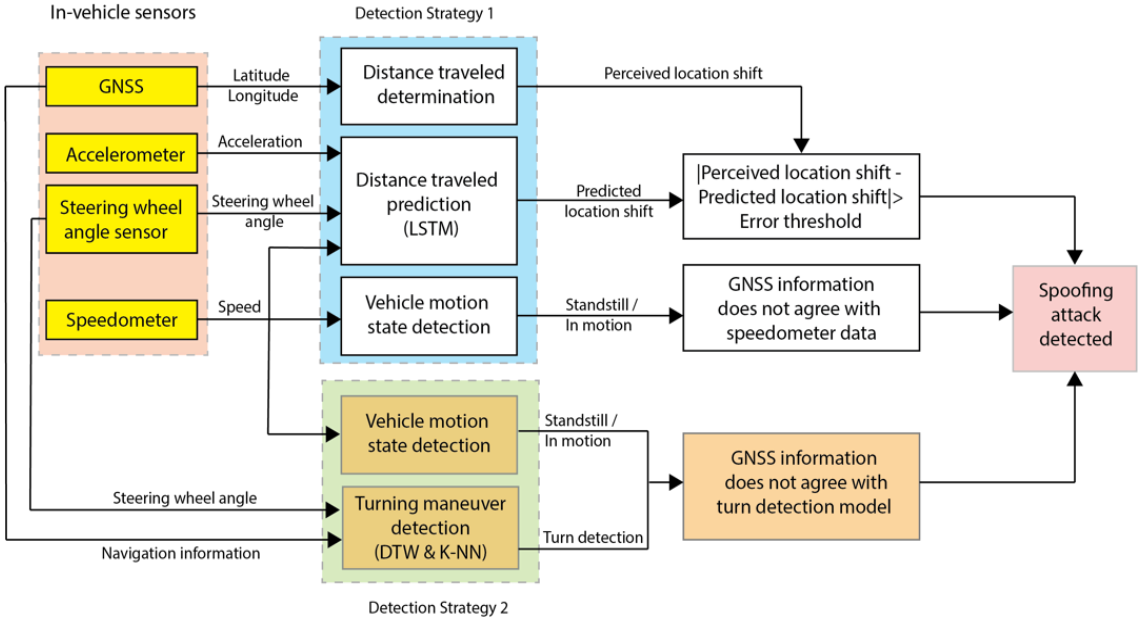


Fig. 3. Sensor fusion based GNSS spoofing attack detection framework.

We have created ten attack datasets sfor each of these four unique attack types using data from the HDD to mimic the actual spoofing scenarios. Note that it is not necessary to simulate spoofed GNSS signals to generate different attack datasets. Each attack dataset contains data of both non-spoofed and spoofed scenarios. The turn-by-turn attack scenario is modeled by selecting a route from the HDD; then at ten arbitrary locations (observations) of the continuous route, the distance travelled between two consecutive time stamps of an AV is manually changed to a value mimicking a realistic location shift. The location shift distance is selected using a random number generator that generates ten random values between 8 m and 200 m representing a block to a couple of blocks distance. The stop attack scenario is modeled by selecting routes from HDD and setting the speedometer data to zero. On the other hand, the GNSS-derived distances between two consecutive time stamps are set to zero for the overshoot attack type.

VI. GNSS Spoofing Attack Detection Framework

We have developed a robust GNSS spoofing attack detection framework (see Figure 3), and this framework involves two concurrent strategies in which data from in-vehicle low-cost sensors—i.e., GNSS, accelerometer, steering wheel angle, and speedometer—are fused to provide a unified and robust GNSS spoofing attack detection approach. These two strategies include: (i) comparison of predicted location shift with inertial sensor-based location shift—in addition to monitoring of vehicle motion states and (ii) detection of turning maneuvers (right and left turns).

The goal of the first strategy is to develop a vehicle state prediction model that can predict a subject vehicle's state information (such as distance traveled or location shift) by fusing data from multiple in-vehicle sensors (i.e., speedometer, accelerometer, and steering wheel angle sensors). For every timestamp, attack-free speed, acceleration, and steering angle data of an AV are fed to train a deep recurrent neural network model, which is LSTM. The LSTM model will predict the location shift between two consecutive timestamps. This model can predict the location shift considering the long-term dependencies by storing the temporal dependency of the time-series data in the recurrent hidden layer's memory blocks. Along with checking an AV's location shift, this strategy also continuously checks if the corresponding AV is in motion or in a standstill state using the speedometer output. Note that a predicted location shift of an AV alone cannot reliably measure its motion state. If the speed difference between the speedometer and the GNSS data is not within an error threshold, then an attack is detected. This comparison is useful to detect stop and overshoot attacks.

According to the second strategy, steering angle data is used to recognize different types of turns (left or right). For example, steering angle sensor or gyroscope output can provide turn maneuvering data. In this study, we use steering angle sensor output instead of gyroscope as the steering angle data provide better vehicle maneuvering information and are available in the HDD. A vehicle's turn can be divided into three categories: left-turns, right-turns, and U-turns. In this paper, we have only concentrated on detecting left and right turns, which are the most common types of turns. Because of variability in driving maneuvers with the different roadway turning curvatures, the length of the duration of the steering angle sensor data may differ between different left (or right) turns. In order to recognize turning maneuvers, we train a dynamic time warping (DTW) algorithm using data from different turning maneuvers in the HDD dataset to learn the pattern of left and right turns. The DTW algorithm determine the degree of similarity between two time series data. A k-NN classifier algorithm is then used to categorize various turns. This detection system continuously compares an output from the inertial sensors with turning information from GNSS to detect and classify a turning maneuver. There is a possibility that an AV is at a standstill state, but the steering wheel is rotating for adjustment; in such

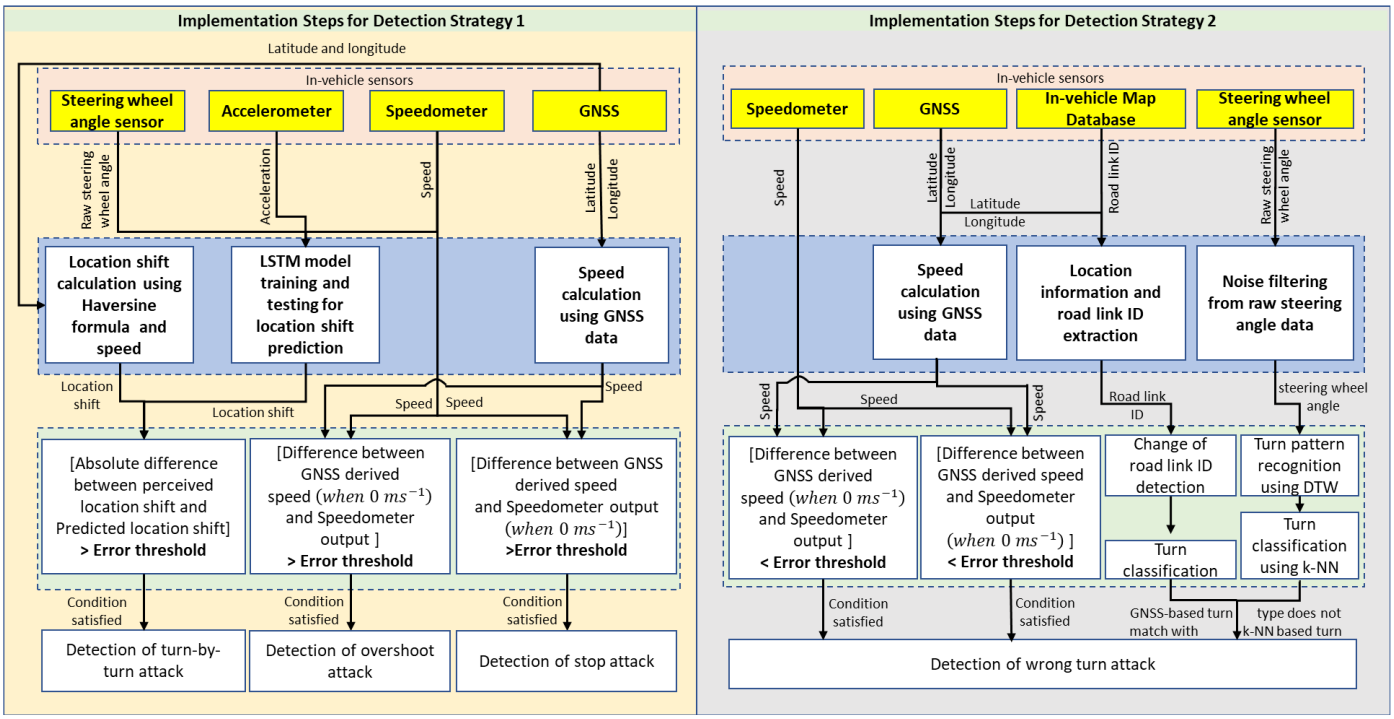


Fig. 4. Strategy 1 & 2 implementation steps.

a case, the steering wheel data can resemble a turn pattern; however, no turning maneuver occurs in reality. The turn detection strategy also takes input from the speedometer sensor to address such a scenario. If a turn signal is detected and the corresponding vehicle's speed is higher than the maximum possible error, our strategy will detect a turn; otherwise, no turn will be detected. It is a common scenario when an AV will be at a parking lot and try to park itself. Overall, any GNSS spoofing attack can be detected based on these two strategies—i.e., (i) comparison of predicted location shift with inertial sensor-based location shift—in addition to monitoring of vehicle motion states and (ii) detection of turning maneuvers (right and left turns). Figure 4 presents detailed implementation steps for both strategies, and the following subsections describe these implementation steps in detail.

A. Development of Detection Strategy 1

The first strategy incorporates two different types of vehicle state information: (a) predicted and perceived location shift between two consecutive timestamps; and (b) vehicle motion state detection (standstill/in motion). With 128 and 64 neurons in the first and second hidden layers, respectively, a 2-stacked LSTM [37] architecture is used to predict the location shift made by an AV between two consecutive timestamps. The training and validation data includes acceleration, steering wheel angle, and speed. The output is the location shift between the current timestamp and the immediate future timestamp. In this study, the data generation frequency is 120 Hz; hence 0.00833s is the time difference between two consecutive timestamps [33]. The HDD dataset contains driving data of multiple days and various routes. One of the routes is selected to train and validate the LSTM model. The route is chosen in such a way so that it represents an urban road network and includes frequent stops and turning maneuvers. The continuous driving data of the route is split into training with 241,990 observations and validation with 103,709 observations. Before feeding the sensor output to the LSTM training, the input features are normalized between 0 and 1. The LSTM hyperparameters, i.e., number of neurons, number of epochs, batch size, and learning rate, are selected by a trial-and-error approach [37] as it is a time series-based prediction model. The hyperparameters' values and the optimizer's name are listed in Table I. After testing the LSTM-based prediction model, we found that the Root Mean Square Error (RMSE) of the predicted location shift is 0.02 m, and the maximum absolute error is 0.5 m. We also found that the distribution of the error represents a normal distribution. Choosing a very low false-positive probability of 10^{-9} (as described in [10] and see Table III), the prediction model error threshold is 0.056 m.

A detection threshold is established by adding the prediction model error threshold and GNSS positioning error, as presented in (2). In HDD, a GeneSys Eletronik GmbH ADMA with DGPS is used for collecting the position data. This DGPS model has a relative position error of 0.01m and an expected position error (standard deviation) of 0.001m. Although the positioning error of GPS used for the HDD dataset has a relative position error of 0.01 m, a typical DGPS system has a position error of 0.1 m after accounting for the biases. To make the threshold value more generalized, it is assumed that the GPS position error is 0.1 m. Therefore, the error threshold value is 0.156 m. An attack will be detected if the difference between the perceived location shift using GNSS and the predicted location shift is greater than the error threshold.

In this study, the developed LSTM-based vehicle location shift prediction model is trained using HDD that represents different real-world urban driving scenarios in the San Francisco Bay Area. It includes different types of left turns, right turns, and lane-change maneuvers, as well as deceleration and acceleration behaviors, which make the dataset suitable for

developing, evaluating, and validating the LSTM model for urban road network. As the training dataset features cover different types of driving maneuvers, the trained model can handle any real-world driving pattern for any large urban area. Optimum hyperparameters have been used to train the model. In addition, we have proved the generalizability of our LSTM architecture through validation. Figure 5 presents the Mean Absolute Error (MAE) loss profile (learning curve). Here, the y-axis represents the mean absolute error loss for both the training and the validation datasets, and the x-axis presents the number of epochs. The learning curve reveals that the training loss first decreases and then stabilizes, i.e., there is no significant change in training loss, proving that the LSTM model is not under-fitted. Moreover, as both training and validation losses stabilize with experience (with increasing epochs), the LSTM model is not overfitted. Furthermore, the training and validation losses are low. The initial peak in the training and validation indicates that the model was not generalized at that point, but with increasing epochs, the model became stable and generalized. The training and testing data represent real-world driving data for different times of the day on urban routes, which represents the generalized behavior of the neural network model in an urban network. Above all, the training and testing datasets are exclusive, i.e., the training and testing datasets represent not only different days but also multiple routes and diverse driving behaviors. Hence, it can be concluded that the adopted neural network is transferable to a similar urban network and has a strong generalization ability and wider applicability to the prediction error threshold for predicting location shift.

$$Error\ Threshold = Prediction\ Model\ Error\ Threshold \text{ (using False Positive Probability of } 10^{-9}\text{)} + Positioning\ Error\ of\ the\ GNSS \quad (2)$$

TABLE I
LSTM MODEL HYPERPARAMETERS

Hyperparameters and Optimizer	Value
Number of neurons (1st layer)	128
Number of neurons (2nd layer)	64
Number of epochs	90
Batch size	50
Learning rate	0.01
Optimizer	Adam

The vehicle motion state detection mechanism is used to detect stop and overshoot attacks. Here, the GNSS-derived speed is compared with the speedometer output. Latitude and longitude data from a GNSS, along with the time difference between two measurements, are used to calculate the GNSS-derived speed of an AV. We assume that an AV is in a standstill state if the speedometer output is zero. An error threshold is established based on the deviation between the GNSS-derived speed and the speedometer speed (zero speed or standstill state of a vehicle) to avoid false attack detection. The tradeoff between the error threshold and attack detection performance is considered using a trial-and-error method. The error threshold for the stop attack detection is set to the 90th percentile of the error because setting a threshold value with a higher percentile significantly increases false-negative instances.

Uncompromised GNSS data from HDD are used to calculate the error threshold. After examining speedometer and GNSS data from the HDD dataset, we found that the difference (i.e., error) between GNSS-derived and speedometer speed ranges from 0 m/s to 1.26 m/s in a standstill state of a vehicle. It indicates that the GNSS-derived speed is not zero for all observations when the speedometer shows a zero speed. We found that the error between the speedometer and the GNSS-derived speed is equal to or less than 0.6 m/s for 90% of the observations, i.e., the 90th percentile of the speed error is 0.6 m/s. Analysis of HDD dataset also reveals that their vehicle takes equal to or less than 0.7 s to reach the speed of 0.6 m/s from a standstill state. Hence, even if a vehicle starts moving from a stopped condition and tries to attain 0.6 m/s, our framework can detect a stop attack (if a GNSS receiver is compromised) within 0.7 s because the data frequency is 100 Hz (0.01 s) and the computation time for each observation in our framework is 0.691 µs second. A stop attack is flagged when the speedometer shows zero speed, and the GNSS-derived speed is more than the 90th percentile value. On the other hand, if GNSS-derived speed is zero and the speedometer output is more than the 90th percentile value, an overshoot attack is flagged.

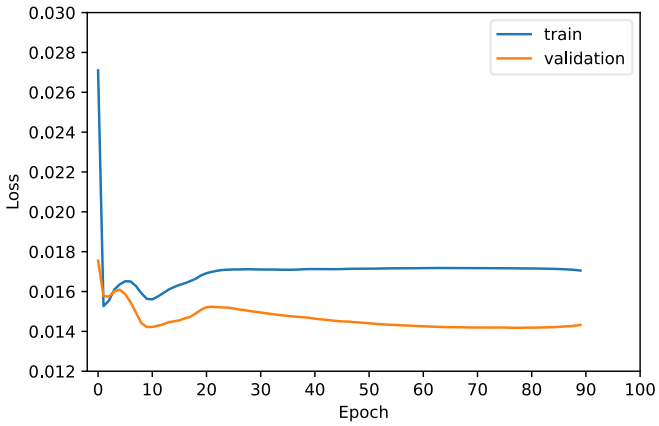


Fig. 5. Comparison of Mean Absolute Error (loss) profiles with the optimal parameter set.

B. Development of Detection Strategy 2

The second strategy consists of two detection mechanisms: turning maneuver detection and vehicle motion state detection. A vehicle turning maneuver can be detected and classified by comparing the steering angle sensor output and the GNSS output. If a turn is detected using steering angle data, but GNSS shows no turn, then an attack will be detected. Moreover, if the steering angle change represents a right turn and the corresponding GNSS detects a left turn, an attack will be detected. Using a standard GNSS navigation system, a turn can be detected using the GNSS location output and road link ID. from a built-in map [38]. When an AV takes a turn, the link ID will change. The current and previous links are determined using the GNSS-derived location. The angle between the GNSS location before and after the road link change is used to detect the turn type. The clockwise angle for a right turn is between 0^o and 180^o. In the case of a left turn, the clockwise angle

should be between 180^o and 360^o.

The AV steering angle readings create unique shapes for left and right turns (as shown in the top left corner of Figure 6). For instance, if an AV makes a right turn, the AV first turns the steering to the right to enter the road and then turns the steering to the left to align itself along the lane marking, which forms a distinct vehicle trajectory path as shown in Figure 6. However, due to varying steering behavior and road geometry, the length of the duration of the maneuvering data for different turns is not uniform.

A k-Nearest Neighbors (k-NN) clustering algorithm is combined with a dynamic time warping (DTW) algorithm for developing a left and right turn detection strategy. The steering wheel angle data are used as input to the turn classification model. A DTW algorithm compares the patterns and measures the similarity between two different time-series data of the different number of observations. The DTW iteratively warps the time axis to align two input time series and searches for an optimal match; it then calculates the warp path distance, which is the cumulative distance between each pair of observations. The path with minimum total cost represents the DTW distance between two different time series data as shown in (3):

$$DTW(T,S) = \underset{w = w_1, w_2, w_3, \dots, w_k, \dots, w_K}{argmin} \sqrt{\sum_{k=1, w_k=(i,j)}^{K} (t_i - s_j)^2} \quad (3)$$

where T and S are ground truth and training steering angle data, respectively; w represents a warping path; t_i is the ith observation of time series T; and s_j is the jth observation of the time series S. We have used the FastDTW [39] algorithm to reduce the computational time and satisfy the real-time detection requirement.

Fig. 6. Patterns of right and left turning maneuvers.

k-NN is a widely used classifier. The k-NN algorithm assigns a common class among its k nearest neighbors based on a distance metric. In this study, DTW is used as the distance metric for k-NN. The k-NN model is trained and tested using the steering angle readings from the HDD, which are related to right and left turns. In the training dataset, there are 19 right turns and 13 left turns, while there are 7 right turns and 6 left turns in the testing dataset (see Table II). The sample left and right turn locations using black and purple lines are shown in Figure 6. The combined k-NN—DTW turn detection model can classify left, and right turns with an accuracy of 100% along with precision value, recall value, and F1-score of 1. Note that our training dataset contains different patterns of left and right turns. The concept, method, and basis for selection of the error threshold for the stop, overshoot, and turn-by-turn attack are presented in Table III.

TABLE II
DATA USED FOR K-NN—DTW MODEL TRAINING

Turn type	Dataset	Number of turns	Number of observations
Right	training	19	15969
	testing	7	8209
Left	training	13	12974
	testing	6	6706

TABLE III
ERROR THRESHOLD SELECTION CRITERIA

Attack type	Concept	Method	Basis
Stop attack	Tradeoff between attack detection performance and percentile value of difference between GNSS derived speed ($when\ 0\ ms^{-1}$) and Speedometer output	Trial and error method	90th percentile of error— i.e., difference between GNSS derived speed ($when\ 0\ ms^{-1}$) and Speedometer output
Overshoot attack	Tradeoff between attack detection performance and percentile value of difference between GNSS derived speed ($when\ 0\ ms^{-1}$) and Speedometer output	Trial and error method	90th percentile of error— i.e., difference between GNSS derived speed ($when\ 0\ ms^{-1}$) and Speedometer output
Turn-by-turn attack	Analyzing distribution of absolute difference between perceived location shift and Predicted location shift	Probability of false	False positive probability of 10^{-9}

VII. EVALUATION RESULTS

To prove the efficacy of our detection framework, we have evaluated the detection framework against four attack scenarios: turn-by-turn, overshoot, wrong turn, and stop. Ten datasets are created for each scenario, containing both compromised and uncompromised GNSS data. These datasets mimic actual GNSS spoofing attacks as they are created based on real-world urban driving data. Thus, the evaluation results represent our framework's performance for diverse urban driving and road conditions. In addition, our framework's performance is also compared with a baseline spoofing attack detection framework, which further proves the efficacy of our approach.

A. Baseline Framework

Figure 7 presents a baseline framework for GNSS spoofing attack detection as presented in [10]. This framework consists of three detection monitors. In these monitors, low-cost in-

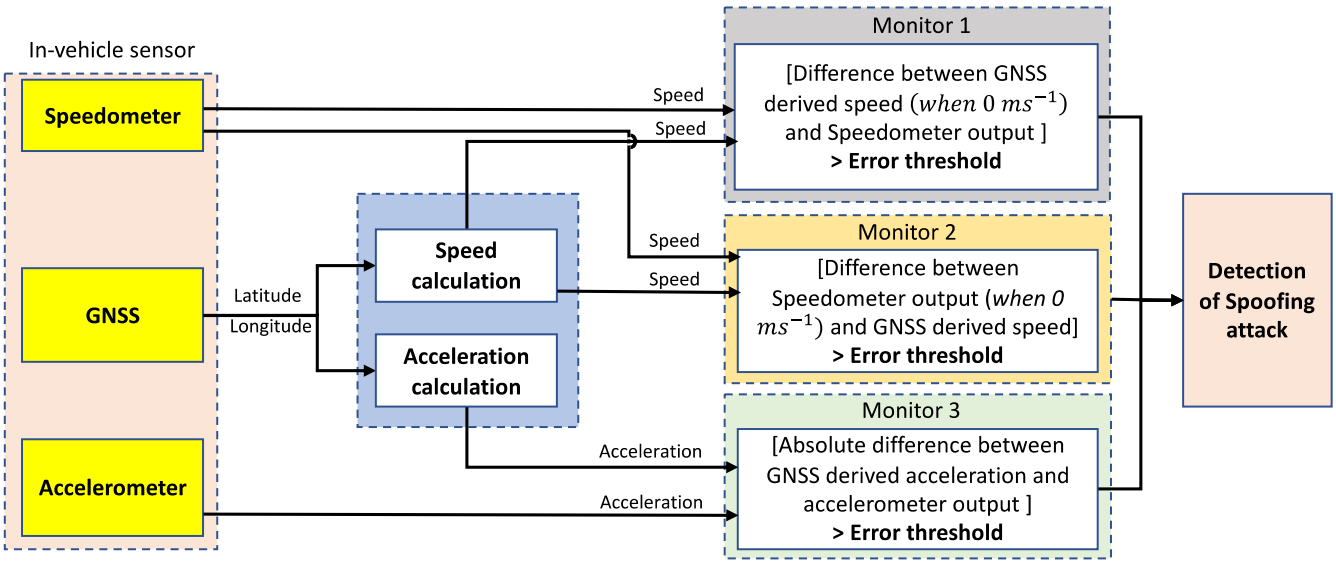


Fig. 7. Baseline-framework for GNSS spoofing attack detection.

vehicle sensors'—i.e., speedometer and accelerometer— data along with GNSS-derived data are compared for attack detection. The first two monitors check the speedometer and GNSS derived speed. Then, an error threshold is established to address the sensor alignment and bias. To obtain the error threshold, an uncompromised GNSS dataset is used. We have selected the absolute maximum difference between the GNSS-derived speed and the speedometer's speed as the error threshold, and the error threshold is 3.44 m/s. The basis of selecting maximum speed error as an error threshold is that an attack (stop or overshoot) can be detected even if there are any speedometer sensor measurement issues exist, such as bias and alignment.

As shown in Figure 7, the first monitor is used to detect overshoot attacks, and it becomes active when the GNSS-derived speed is zero. After activation, it checks whether the speedometer output is higher or lower than the error threshold. If the speedometer output exceeds the error threshold, an attack is flagged. The second monitor is used to detect stop attacks, and it detects an attack when the speedometer output is zero. This monitor checks whether the GNSS-derived speed is higher or lower than the error threshold. If the GNSS-derived speed exceeds the error threshold, an attack is flagged. In the third monitor, GNSS-derived acceleration is compared with the accelerometer output. The GNSS-derived acceleration is calculated using the GNSS-derived speed and the time difference between two consecutive measurements. Note that all three monitors run simultaneously to detect any GNSS spoofing attacks in real-time.

As the HDD dataset does not contain acceleration data, we have calculated acceleration from the speedometer data and the time difference between the two consecutive measurements. Generally, the acceleration measurements are noisy and consequently not suitable for observation-by-observation comparison. Thus, 5-second moving averaged accelerations are used for both GNSS and speedometer-derived acceleration [10]. The acceleration difference between two consecutive timestamps is used to detect turn-by-turn spoofing attacks. An error threshold is estimated based on accelerometer sensor error to increase the effectiveness of the detection framework. The error threshold is established first by comparing the GPS and speedometer derived acceleration for a setup where GPS is not compromised. The distribution of the difference between the GPS and speedometer derived acceleration represents a normal distribution. We choose a very low false-positive probability of 10^{-9} (same as [10]) to determine the detection threshold, which is 0.77 m/s^2. To prove the effectiveness of our framework, we compared the performance of this baseline framework with ours in four unique spoofing scenarios as presented in section V.

B. Turn-by-turn Spoofing Attack

Figure 8 illustrates how to detect turn-by-turn GNSS spoofing attacks. The number of observations is displayed on the x-axis, and the y-axis presents the difference between perceived and predicted location shifts. In this scenario, the location shift for the uncompromised GNSS case is shown in Figure 8(a), where the location shifts never cross the error threshold. When an attack is generated, the difference between the perceived and predicted location shift (Figure 8(b)) crosses the threshold; thus, the attack is detected. Figure 9 presents profiles for the absolute difference between the perceived location shift and predicted location shift for ten different turn-by-turn type spoofing attack scenarios (AS). Note that the absolute difference between the perceived and predicted location shifts is plotted on a log scale in Figure 9 to show the negligible difference. At the point of beginning of a spoofing attack (as shown using different markers), the difference between perceived and predicted location shift is higher than the error threshold value; thus, it detects the attack. It is worth mentioning that no false attack is detected using our detection framework (see Figure 9).

The baseline framework is evaluated against the same attack datasets as mentioned above, and the results are presented in Figure 10. Although the baseline framework's monitor 3 can detect location shifts, as can be seen in Figure 10, a false location shift is detected after 500 observations for each attack dataset. Moreover, a comparison of GPS and accelerometer data is impossible if the GPS and accelerometer measurements are not aligned. Thus, it will exponentially increase the

probability of false positives, making the baseline method a failure. The baseline framework requires axis alignment between GNSS and accelerometer reference frames, and the GPS measurements also need to be first translated to speed and then to acceleration, which may further add error in the acceleration calculation. Furthermore, sensor accuracy issues and measurement noise can also jeopardize detection performance.

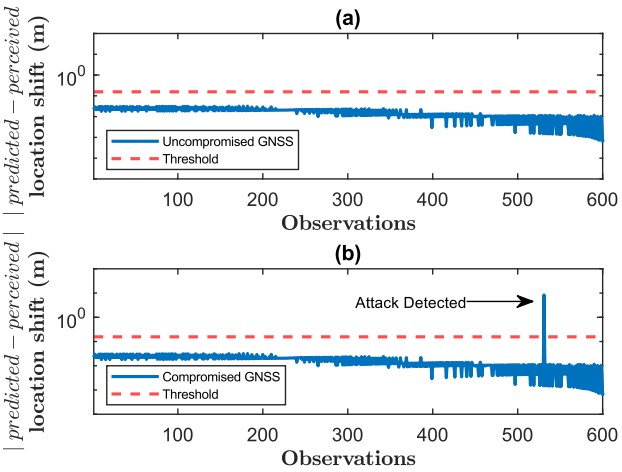


Fig. 8. An example GNSS spoofing scenario: (a) location shift for an uncompromised GNSS observation; and (b) location shift for a compromised GNSS observation.

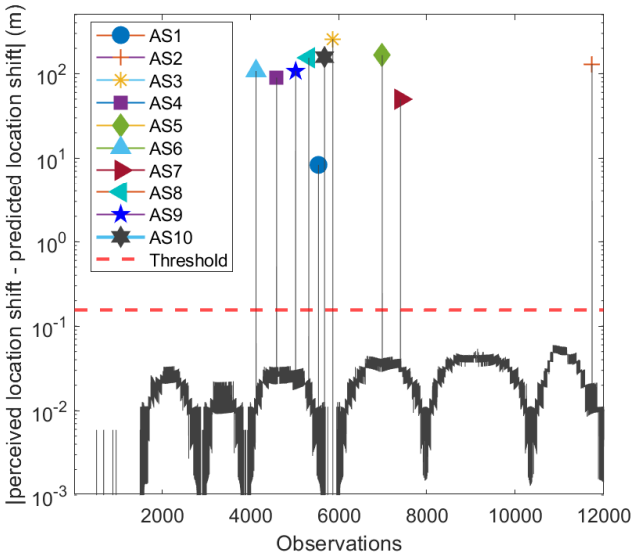


Fig. 9. Attack detection for ten turn-by-turn attack scenarios (AS) (our framework).

C. Stop Spoofing Attack

Based on our strategy developed in this study, a stop attack can be detected by vehicle motion state detection—i.e., detection strategy 1: comparing between perceived and predicted location shift. In this paper, we have presented results of stop attack detection by vehicle motion state detection. Figure 11 presents the evaluation outcome of both the baseline strategy and our developed stop attack detection strategy. We have plotted the attack dataset number on the x-axis, and the y-axis presents the percentage of true negative (cornflower blue), false negative (punch), true positive (goldenrod), and false-positive (grape) detection (see the "Notes" at the bottom of Figure 11).

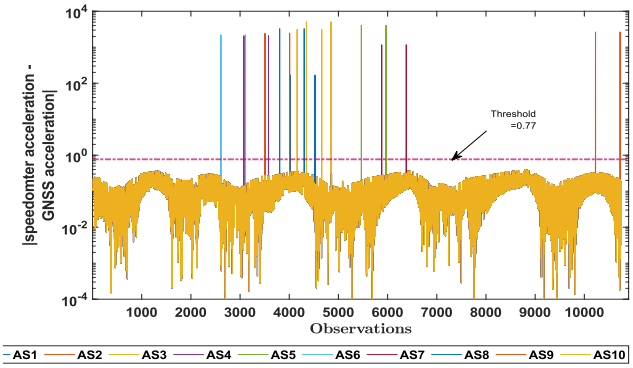
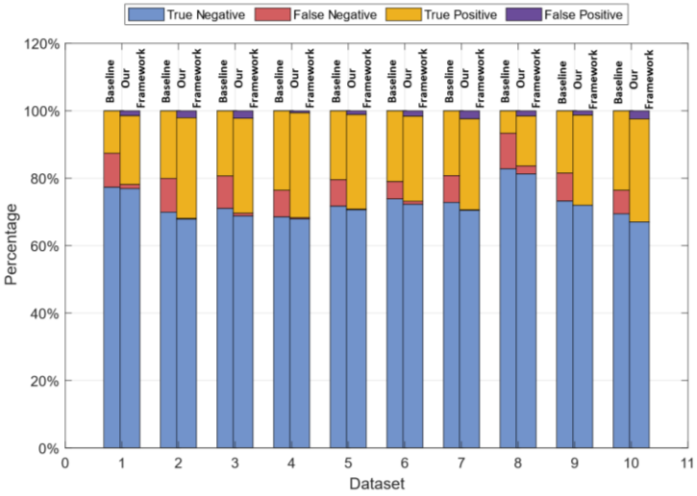


Fig. 10. Attack detection for ten turn-by-turn attack scenarios (AS) (baseline framework).



Notes: True Negative— GNSS not compromised, and no attack flagged; False positive— GNSS not compromised, and an attack flagged; True Positive— GNSS compromised, and an attack flagged; False-negative— GNSS compromised, and no attack flagged.

Fig. 11. Stop attack detection results.

It is evident from Figure 11 (see "true negative (cornflower blue)" and "true positive (goldenrod)") that the detection accuracy of our framework is high. Note that we have used ten different datasets to evaluate our strategy. We found that our strategy provides a high percentage of true positives and true negatives with a low percentage of false positive and false negative. The false-negative instances occur for two reasons. The first reason is that our framework does not flag an attack for the observations where both the compromised GNSS-derived speed and the speedometer speed are zero. Such a condition arises whenever an AV is stopped during an ongoing stop attack. The second reason is that our framework cannot detect an attack when an AV starts moving from the stopped condition to moving condition, and the speedometer speed is lower than the error threshold. As we mentioned before, a vehicle takes less than or equal to 0.7s to attain the error threshold limit—i.e., 0.6 m/s. We found that the former reason mostly causes false-negative cases.

Table IV presents a summary of stop attack evaluation results in terms of precision, recall, accuracy, and F1 score. Precision

TABLE IV
EVALUATION SUMMARY OF STOP ATTACK SCENARIO

	Our Framework				*Baseline Approach*			
Dataset	*Precision*	*Recall*	*F1 score*	*Accuracy*	*Precision*	*Recall*	*F1 score*	*Accuracy*
1	*0.98*	*0.94*	*0.98*	*0.97*	*0.88*	*0.56*	*0.94*	*0.90*
2	*1.00*	*0.99*	*0.98*	*0.98*	*0.87*	*0.67*	*0.90*	*0.90*
3	*0.99*	*0.97*	*0.98*	*0.97*	*0.88*	*0.67*	*0.90*	*0.90*
4	*0.99*	*0.99*	*0.99*	*0.99*	*0.90*	*0.75*	*0.92*	*0.92*
5	*1.00*	*0.99*	*0.99*	*0.99*	*0.90*	*0.72*	*0.92*	*0.92*
6	*0.99*	*0.97*	*0.98*	*0.97*	*0.93*	*0.80*	*0.95*	*0.95*
7	*1.00*	*0.99*	*0.98*	*0.97*	*0.90*	*0.71*	*0.92*	*0.92*
8	*0.97*	*0.86*	*0.98*	*0.96*	*0.89*	*0.39*	*0.89*	*0.92*
9	*1.00*	*1.00*	*0.99*	*0.99*	*0.90*	*0.69*	*0.92*	*0.93*
10	*1.00*	*1.00*	*0.98*	*0.98*	*0.91*	*0.77*	*0.93*	*0.90*

is the measure of how accurately an attack is detected out of all the attack detection instances considered in this study. The precision of our detection model varies from 97% to 100%, whereas for the baseline case, precision varies from 87% to 93%. Recall refers to the percentage of the observations where attacks are detected out of all the compromised observations. As provided in Table IV, the recall of our framework varies from 97% to 99%, and that of the baseline varies from 39% to 80%. The accuracy of our framework ranges from 97% to 99%. The F1 score reflects the balance between precision and recall. For the high frequency of true negative, the F1 score is a better measure. The F1 score of our framework ranges from 0.98 to 0.99, which proves that the precision and recall are well balanced. The F1 score for the baseline approach ranges between 89% and 95%. Thus, our framework performs better in all four parameters than the baseline approach.

D. Overshoot Spoofing Attack

An overshoot attack can also be detected by comparing a GNSS-derived speed with speed from the speedometer. The evaluation results of baseline, as well as our overshoot attack detection strategy, are presented in Figure 12. The attack dataset number is presented on the x-axis, and the y-axis presents the frequency of true negative (green), false negative (purple), true positive (red), and false-positive (orange). Like the stop attack detection strategy, our overshoot attack detection strategy effectively detect attack and non-attack cases. The reasons for false negatives are the same as we described in the stop attack result subsection.

Table V presents a summary of overshoot attack evaluation results in terms of accuracy, precision, recall, and F1 score. The accuracy of our framework ranges from 83% to 100%. The precision of our detection model varies from 86% to 100%. As provided in Table V, the recall of our framework varies from 80% to 100%. The precision and accuracy for overshoot attack detection are lower than those of stop attack due to more instances where both the speedometer and the GNSS-derived speed is zero. As the vehicle stops during an ongoing overshoot attack, both speeds become zero, and no attack is flagged. The F1 score of our framework ranges from 90% to 98%, which shows that the precision and recall are well balanced. The baseline framework is tested against the same attack datasets as our framework. Accuracy for baseline case ranges from 81% to 100%; F1 score ranges from 81% to 100%, and precision ranges from 74% to 100%. The recall for the baseline approach varies from 57% to 100%.

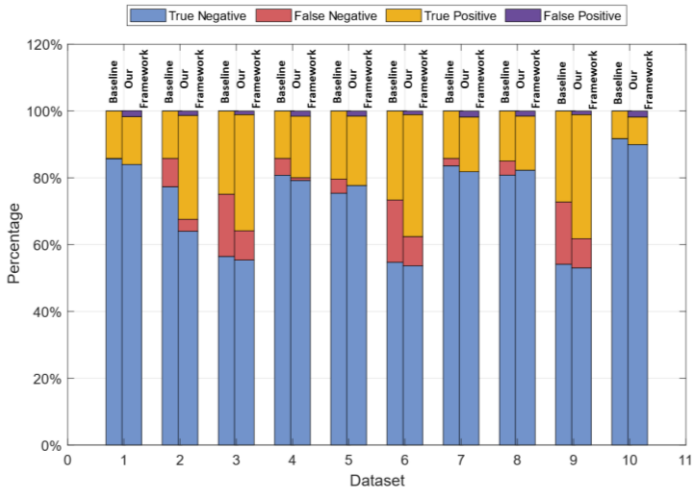


Fig. 12. Overshoot attack detection results.

E. Wrong Turn Spoofing Attack

The efficacy of the turn detection and classification model using the K-NN—DTW combined method is shown in Table VI. As the calculated precision and recall are 1, all turns are correctly detected and classified, and there is no false detection and classification occurred. The F1 score is also 1 (100% effective in turn detection and classification), which proves the efficacy of the turn detection strategy.

Overall, the results reveal that our detection framework can successfully detect all four attack types in each of the ten scenarios. As per our knowledge, we do not find any existing in-vehicle sensor fusion approach that can be used for comparison with our framework. Thus, we presented our framework's performance results, which show the highest accuracy.

F. Computation Time Requirements

The average computational latency for our first strategy, i.e., the location shift prediction strategy is 0.691µs for each observation, which is less than the GNSS data generation frequency (i.e., 120Hz or 0.0083s or 8300µs). In our second strategy, we have resampled the steering angle sensor data to 5Hz from 120Hz because our experiments showed that the 5Hz

TABLE V
EVALUATION SUMMARY OF OVERSHOOT ATTACK SCENARIO

	Our Framework				*Baseline Approach*			
Dataset	*Precision*	*Recall*	*F1 score*	*Accuracy*	*Precision*	*Recall*	*F1 score*	*Accuracy*
1	*1.00*	*1.00*	*0.98*	*1.00*	*1.00*	*0.99*	*1.00*	*1.00*
2	*0.95*	*0.90*	*0.95*	*0.92*	*0.90*	*0.63*	*0.92*	*0.74*
3	*0.86*	*0.80*	*0.90*	*0.83*	*0.75*	*0.57*	*0.81*	*0.65*
4	*0.99*	*0.96*	*0.98*	*0.97*	*0.94*	*0.74*	*0.95*	*0.83*
5	*1.00*	*1.00*	*0.98*	*1.00*	*0.95*	*0.83*	*0.96*	*0.88*
6	*0.86*	*0.81*	*0.90*	*0.83*	*0.75*	*0.59*	*0.81*	*0.66*
7	*1.00*	*1.00*	*0.98*	*1.00*	*0.97*	*0.87*	*0.98*	*0.92*
8	*1.00*	*1.00*	*0.98*	*1.00*	*0.95*	*0.78*	*0.96*	*0.86*
9	*0.86*	*0.81*	*0.90*	*0.83*	*0.74*	*0.59*	*0.81*	*0.66*
10	*1.00*	*1.00*	*0.98*	*1.00*	*1.00*	*1.00*	*1.00*	*1.00*

sampling rate reduces the computational time while preserving the observational (or sensing) integrity of the right and left turns. The k-NN-DTW model takes 0.08s on average to detect a turn, which is less than the data sampling frequency (i.e., 5Hz or 0.2s). Note that the computational time presented in this study applies to a workstation equipped with a dual Intel Xeon Gold 5215 2.5GHz processor with 128GB DDR4 2666MHz RDIMM ECC RAM memories used to run our experiments.

TABLE VI
K-NN & DTW TESTING OUTPUT

Perceived turn type via GNSS	Number of attack scenarios	Detected turn types using k-NN and DTW	Baseline /actual turn type (ground truth data)	Accuracy & Precision & Recall & F1 Score
Left Turn	5	Right turns	Right turns	1.00
Right Turn	5	Left turns	Left turns	
Left or Right turns	20	No turns	No turns	

VIII. Conclusion

A robust GNSS spoofing attack detection framework is presented in this paper. Data from low-cost in-vehicle sensors are used for detecting sophisticated GNSS spoofing attacks. The framework developed in this study is unique compared to existing approaches in two ways. First, our approach uses deep learning to predict the location shift to detect an attack, while existing GNSS attack detection approaches use deep learning to analyze GNSS signal data to detect an attack. Second, sensors are used neither to determine the vehicle position nor to compare data from a single sensor output with GNSS-derived information. Instead, in our first strategy, we have used data from multiple sensors, i.e., speedometer, steering angle, and accelerometer, to predict the location shift by the next timestamp using LSTM, which is based on an artificial recurrent neural network (RNN) architecture. The predicted location shift for an AV is then compared with the location shift estimated based on the GNSS data to detect a spoofing attack. The vehicle motion state from GNSS and speedometer data are also compared to detect spoofing attacks. Moreover, a turn detection strategy, our second strategy, is used for classifying turns to further detect more sophisticated attacks. The combination of two strategies allows the framework to detect the most sophisticated spoofing attacks where a spoofer has the capability of tampering with a target vehicle's destination, route, and sensor information. The framework presented in this paper is also validated against the four unique attack types. A comparison between a baseline framework and our framework has also been presented. Analyses revealed that our attack detection framework is able to detect different types of attacks with a high degree of success. Further research can be performed to evaluate and validate the framework through real-world experiments. Due to the diverse nature of spoofing attacks, a single strategy cannot detect and mitigate different types of attacks. A GNSS interference can be intentional by an attacker or unintentional because of natural vulnerabilities. It is a challenge to distinguish between intentional and unintentional interference. Separating an authentic GNSS signal from a spoofed signal introduces further complexity. Thus, future research focuses on developing robust anti-spoofing technologies to mitigate spoofing attacks on GNSS receivers.

Acknowledgment

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.

References

[1] 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.

[2] "SAPOS ® Precise Positioning in Location and Height Satellite Positioning Service of the Offical German Surveying and Mapping Agency for Geoinformation and State Survey of Lower Saxony (LGLN)".

[3] 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.

[4] M. Adjrad and P. Groves, "3D-Mapping-aided GNSS exploiting Galileo for better accuracy in dense urban environments," in *2017 European Navigation Conference, ENC 2017*, Jun. 2017, pp. 108–118. doi: 10.1109/EURONAV.2017.7954199.

[5] M. Adjrad and P. D. Groves, "Intelligent urban positioning using shadow matching and GNSS ranging aided by 3D mapping," in *29th International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2016*, Sep. 2016, vol. 1, pp. 534–553. doi: 10.33012/2016.14845.

[6] "Interference Effects and Mitigation Techniques," in *Global Positioning System: Theory and Applications, Volume I*, American Institute of Aeronautics and Astronautics, 1996, pp. 717–771. doi: 10.2514/5.9781600866388.0717.0771.

[7] "Multipath Effects," in *Global Positioning System: Theory and Applications, Volume I*, American Institute of Aeronautics and Astronautics, 1996, pp. 547–568. doi: 10.2514/5.9781600866388.0547.0568.

[8] 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

[9] 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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] C. Tanil, S. Khanafseh, M. Joerger, and B. Pervan, "Kalman filter-based INS monitor to detect GNSS spoofers capable of tracking aircraft position," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016*, May 2016, pp. 1027–1034. doi: 10.1109/PLANS.2016.7479805.

[15] 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

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

[17] 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.

[18] 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.

[19] 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

[20] 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

[21] 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.

[22] 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

[23] N. Vagle, A. Broumandan, and G. Lachapelle, "Multi-antenna GNSS and INS/Odometer Coupling for Robust Vehicular Navigation," *International Technical Symposium on Navigation and Timing*, p. 2017.

[24] 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

[25] Y. Hu, S. Bian, B. Li, and L. Zhou, "A Novel Array-Based Spoofing and Jamming Suppression Method for GNSS Receiver," *IEEE Sensors Journal*, vol. 18, no. 7, pp. 2952–2958, Apr. 2018, doi: 10.1109/JSEN.2018.2797309.

[26] S. Liu *et al.*, *Stars Can Tell: A Robust Method to Defend against {GPS} Spoofing Attacks using Off-the-shelf Chipset*. {USENIX} Association, 2021. Accessed: Jan. 27, 2022. [Online]. Available: https://www.usenix.org/conference/usenixsecurity21/presentation/ma

[27] "Protection Against GNSS Spoofing - GNSS Solutions for OEMs." https://oemgnss.trimble.com/technologies/protection-against-gnss-spoofing/ (accessed Apr. 25, 2022).

[28] "Spoofing: is your GPS attack proof? | Septentrio." https://www.septentrio.com/en/learn-more/insights/spoofing-your-gps-attack-proof (accessed Apr. 25, 2022).

[29] "GNSS Anti Spoofing, GPS Performance Testing - Spirent." https://www.spirent.com/blogs/with-gnss-spoofing-attacks-on-the-rise-resilience-and-robustness-go-hand-in (accessed Apr. 25, 2022).

[30] "WO2021067790A1 - Methods for detecting replay attacks in gnss systems and devices thereof - Google Patents." (accessed Apr. 10, 2022).

[31] "Nobody's Fool: Spoofing Detection in a High-Precision Receiver - Inside GNSS - Global Navigation Satellite Systems Engineering, Policy, and Design." https://insidegnss.com/nobodys-fool-spoofing-detection-in-a-high-precision-receiver/ (accessed Apr. 10, 2022).

[32] "US10338229B2 - Method and apparatus for providing secure timing and position synchronization from GNSS - Google Patents." (accessed Apr. 10, 2022).

[33] 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

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

[35] K. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang, "A practical GPS location spoofing attack in road navigation scenario," in *HotMobile 2017 - Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*, Feb. 2017, pp. 85–90. doi: 10.1145/3032970.3032983.

[36] J. R. Van Der Merwe, X. Zubizarreta, I. Lukčin, A. Rügamer, and W. Felber, "Classification of Spoofing Attack Types," in *2018 European Navigation Conference, ENC 2018*, Aug. 2018, pp. 91–99. doi: 10.1109/EURONAV.2018.8433227.

[37] 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.

[38] Y. Zhao, T. Yamamoto, and T. Morikawa, "An analysis on older driver's driving behavior by GPS tracking data: Road selection, left/right turn, and driving speed," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 5, no. 1, pp. 56–65, Feb. 2018, doi: 10.1016/J.JTTE.2017.05.013.

[39] S. Salvador and P. Chan, "FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space."

Sagar Dasgupta (Student Member, IEEE) is a Ph.D. student at the University of Alabama, Tuscaloosa, Alabama. He received his B.Tech. in Mechanical Engineering from Motilal Nehru National Institute of Technology Allahabad, Prayagraj, UP, India. He received his M.S. in Mechanical Engineering form Clemson University, South Carolina. His research interest includes data analytics, machine learning, connected and automated vehicles, cybersecurity, and sensor fusion.

Mizanur Rahman (Member, IEEE) is an assistant professor in the Department of Civil, Construction, and Environmental Engineering at the University of Alabama, Tuscaloosa, Alabama. After his graduation in August 2018, he joined as a postdoctoral research fellow for the Center for Connected Multimodal Mobility (C^2M^2), a U.S. Department of Transportation Tier 1 University Transportation Center (cecas.clenson.edu/c2m2). After that, he also served as an Assistant Director of C2M2. His research focuses on traffic flow theory and transportation cyber-physical systems for connected and automated vehicles and smart cities.

Mhafuzul Islam is currently working as a Senior Researcher at General Motors. He received his Ph.D. in Civil Engineering in 2021 from Clemson University. He also received a B.S. degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2013 and the M.S. degree in Civil Engineering from Clemson University in 2018. His research interest includes transportation cyber-physical systems with an emphasis on data-driven connected autonomous vehicles utilizing machine learning.

Mashrur Chowdhury (Senior Member, IEEE) received his Ph.D. degree in civil engineering from the University of Virginia in 1995. He is the Eugene Douglas Mays Chair of Transportation with the Glenn Department of Civil Engineering, Clemson University, SC, USA. He is the Director of the USDOT Center for Connected Multimodal Mobility (a TIER 1 USDOT University Transportation Center). He is Co-Director of the Complex Systems, Data Analytics and Visualization Institute at Clemson University. Dr. Chowdhury is a Registered Professional Engineer in Ohio, USA. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is a Fellow of the American Society of Civil Engineers and a Senior Member of IEEE.