A Time-Series Based Convolutional VAE for Spoof Detection in Commercial GPS Receivers

Asif Iqbal*, Muhammad Naveed Aman‡, and Biplab Sikdar*
*Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583.
‡School of Computing, University of Nebraska-Lincoln, Nebraska, USA.
Email: {aiqbal, bsikdar}@nus.edu.sg, naveed.aman@unl.edu

***Abstract*—Global Positioning System (GPS) technology is widely used in personal and industrial applications to acquire precise timing and positional information. However, its open-standard signals are vulnerable to spoofing attacks, which can cause serious damage if undetected. Employing detection methods is crucial in critical applications. Machine Learning (ML) methods have been successfully applied for spoofing detection, typically performing detection on individual samples. This work proposes a framework that takes a multivariate time-series window as input, enabling the neural network model to extract meaningful temporal information from the sample window for improved detection performance. We train a Convolutional Variational Autoencoder model using spoof-free samples under a representation learning framework. The detector's performance is evaluated using the publicly available TEXBAT dataset and simulated datasets. Our results show that the proposed detector achieves a True Positive Rate (TPR) above 99% for a low False Positive Rate (FPR) of 2% in both static and dynamic attack scenarios. Additionally, for the sophisticated attack scenario (DS-7) in the TEXBAT dataset, our detector achieved a TPR of 89% for an FPR of 3%, highlighting its robustness against different types of spoofing attacks.**

***Index Terms*—GPS spoofing attacks, machine learning, commercial gps receivers, spoofing detection technique.**

I. Introduction

The contemporary lifestyle heavily relies on an array of intelligent applications, streamlining various mundane tasks. These applications rely upon the acquisition of information from their environment to operate efficiently. Among the fundamental pieces of information is location, which serves as a cornerstone for a multitude of practical applications, including food delivery, navigation, ride-hailing services, fitness tracking, and emergency assistance, among others. The Global Navigation Satellite System (GNSS) like the US Global Positioning System (GPS) stands out as the primary source of reliable positional data that fuels these applications, making it the backbone of modern smart technology. Beyond civilian applications, GPS holds significant relevance and has been integrated into critical infrastructure and cybersecurity domains by governments worldwide. Sectors such as communications, dams, defense industrial base, emergency services, energy, and financial services have incorporated GPS technology [1].

GPS relies on signals received from a constellation of satellites orbiting the Earth. Typically, at any given time, 6-12 GPS satellites (a similar number for other systems like Galileo, GLONASS, etc.) may be visible from a specific location on the Earth's surface [2]. These satellites orbit at an altitude of over 19,300 kilometers from the Earth's surface, resulting in transmitted signals arriving with very low power, approximately around -158.5 dBW under nominal conditions [3]. Unlike the secured and encrypted P(Y) military channels, civilian channels are unsecured and follow open standards. This openness is a key factor in the widespread adoption of GPS systems across numerous everyday applications. Moreover, due to their open nature and low power, these signals are susceptible to various types of radio frequency (RF) interference, whether intentional (such as signal spoofing or jamming) or unintentional (such as side-channel interference or multipath fading) [4]. While unintentional interference is typically intermittent and can be mitigated, intentional interference like jamming—where a high-powered signal is transmitted in the same GPS transmission band—can completely disrupt GPS operations. However, jamming interference is easily detectable. In contrast, spoofing interference is more subtle and potentially more dangerous, as it can lead GPS receivers to output entirely incorrect positional, velocity, and timing (PVT) information, potentially causing significant harm to the applications that are downstream from the GPS module.

The accessibility of software-defined radios and programmable GPS simulators has improved the feasibility of carrying out GPS spoofing attacks. These attacks can be categorized into three levels of complexity: simplistic, intermediate, and sophisticated [5]. Simplistic spoofing attacks are the most straightforward to detect, as they involve transmitting the spoofed signal with a significantly higher power advantage compared to the genuine GPS signal. This results in a visible increase in the received in-band power and a rise in the Carrier-to-Noise Spectral Density Ratio (C/N_0), calculated at the RF and tracking module of a typical GPS receiver, respectively. Intermediate and sophisticated spoofing attacks utilize induced spoofing (or carry-off spoofing) techniques to seize control of the receiver's signal tracking loop [4]. Without countermeasures, the target remains unaware and remains locked onto the spoofed signal. The disparity between intermediate and sophisticated attacks lies in the adversary's capability to fully align the carrier phase of the spoofed signal with the genuine one. Such alignment necessitates either physical access to the target receiver or precise channel

Supported in part by Singapore Ministry of Education Academic Research Fund Tier 1 Grants A-0009040-00-00 and A- 0009040-01-00.

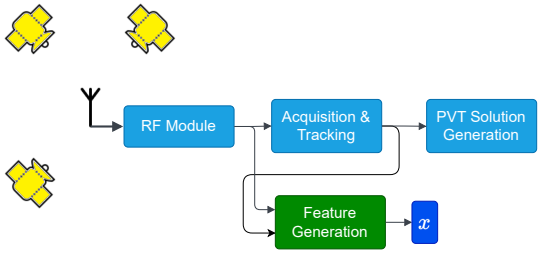


Fig. 1. A typical GPS receiver block diagram

information between the target and the adversaries' antenna [6], which is exceedingly difficult to achieve. Nonetheless, with sufficient financial motivation, attackers may invest in equipment capable of such alignment.

A. Related Works

Various techniques have emerged to detect spoofing by analyzing GNSS signals and receiver properties. These strategies include authentication, multiple antennas, inertial sensors, and single antenna based techniques [4]. Among these, single antenna-based methods stand out for their cost-effectiveness, ease of implementation, and minimal hardware requirements. Within this category, most approaches fall into two main groups: Bayesian-based and Machine Learning (ML)-based detection methods. Bayesian techniques include direct monitoring of in-band received power [7] or through an Automatic Gain Control (AGC) unit [8], Signal Quality Monitoring (SQM) techniques for measuring the correlation peak distortion during carry-off spoofing at the receiver's code tracking loop [9]–[12], or hybrid methods combining both [4], [13], [14]. All these methods analyze feature samples at each time point and conduct hypothesis testing based on carefully chosen signal models, priors, and likelihood functions to detect changes in feature distribution in the presence or absence of a spoofer. Through the selection of an appropriate detection threshold, a targeted detection probability can be attained. In contrast, ML-based methods do not necessitate the meticulous selection of signal models, priors, and noise models; instead, they learn intrinsic patterns found in signal features in a data-driven manner. Consequently, there has been a recent surge in methods utilizing ML models for spoofing detection.

The ML based methods require data features for training, which can be acquired from different modules of the GPS receivers. Majority of the proposed works differ on the training features and the models themselves used to perform the spoof detection. The features can be acquired from RF, acquisition, tracking, or PVT modules of a typical GPS receiver, see Fig. 1. Furthermore, researchers have used supervised as well as unsupervised learning frameworks to train their ML models. For instance, in [15], the authors used a Fully Connected Neural Network (FCNN), Naive Bayes, and K-Nearest Neighbors (K-NN) algorithms to perform spoof detection. These models were trained using the received power at the output of receiver's RF block and two SQM metrics. Authors in [16] trained an FCNN model under supervised learning using pseudorange, Doppler shift, and SNR as the training features. In [17], the authors used the Cross-Ambiguity Function (CAF) computed at the acquisition block of the GPS receiver to train a 2D Convolutional Neural Network (CNN) and an FCNN to detect the presence of multiple peaks in the CAF, which highlights the presence of another signal in addition to the genuine one. A similar approach was adopted in [18] using a Generative Adversarial Network (GAN) as the heart of their detection model. Similarly, [19] presents a spoofing detection model based on an FCNN which is trained on C/N_0, pseudorange, carrier phase, and Doppler shift. Whereas, [20] used a set of 13 features to train multiple ML models for classification. Authors in [21] used received power, C/N_0, and five SQM metrics to train an ensemble of ML classifiers for detecting time-push attacks in smart grid systems. In [22], the authors used the differential location data generated by the PVT module to flag irregularities in the mobility profile of the receiver. Similarly, in [5], authors used two power based, five SQM based metrics, and samples acquired from five additional tracking correlator pairs to train an FCNN and a Variational Autoencoder (VAE) based zero-day detector for spoof detection. Subsequently, the authors in [23] utilized the features proposed in [21] to train a hybrid model based on VAE and GAN for spoof detection. The methods presented in [5] and [23] demonstrated robust performance not only under intermediate-level attacks but also against the sophisticated attack scenario from [24], where a majority of other methods failed.

B. Motivation and Contributions

Most of the high performing methods discussed so far [5], [19], [20], [23] perform spoofing detection on each time point, consisting of multiple features, and do not incorporate the inert sample-sample correlations present in such time-series. In this work, instead of deriving a framework that works on individual data points in isolation, we propose a window based spoofing detection framework which can take advantage of additional information that is embedded in the time-series in the form of temporal correlations, hidden temporal dynamics, and feature interactions over time. In this framework, the spoofing detection is performed over multiple concurrent time samples, combined as a time window, where each time point consists of multiple features acquired from the RF and tracking modules of a typical GNSS receiver. Moreover, making spoofing detection over a time window also helps to reduce the effects of outlier samples often found within a sample window. The base learner used in our detection model is a neural network, consisting of 1D convolutional layers which are trained under the unsupervised representation learning based VAE framework, where the model is trained solely on the genuine spoof free data. Using 1D convolutional layers for learning from windowed multivariate time-series data offers a powerful framework for capturing local temporal patterns, modeling cross-variable dependencies, and extracting hierarchical representations of the data. Thereby, the reconstruction ability of the trained VAE model can then be used to

detect whether a given samples in the window are spoofed or not. The framework is analyzed under multiple attack scenarios, including both static and dynamic receivers. These scenarios contain dynamic simulated as well as intermediate to sophisticated attack scenarios from the public TEXBAT dataset. To summarize, our contributions are:

- A GPS spoofing detection framework which utilizes a computationally lightweight model that is trained under representation learning using only the genuine spoof free GPS features.
- Our proposed model uses a time-series feature window as input for training and detection purposes, using features that are generalized and easily extractable from any standard GPS receiver.
- An extensive evaluation of the proposed framework using the publicly available TEXBAT dataset and a simulated scenario to highlight the transferability of the detection model across geographical locations.

Rest of the paper is organized as follows, Section II presents the preliminary information on the GPS signal model and feature extraction procedure. Section III outlines the proposed framework including the model training and detector setup. The evaluation of the proposed framework is detailed in Section IV, followed by the conclusion in Section V.

II. Background and Preliminaries

In this section, we briefly discuss the relevant GPS receiver modules that are used during the feature extraction process and discuss why the selected features are useful.

A. Signal Model at the Tracking Stage

In this work, we focus our discussion on the most widely available GPS receivers, which are based on a single antenna. The operation of such a receiver can be divided into several distinct modules, as illustrated in Fig. 1. Among these, the most relevant for our purposes are the Radio Frequency (RF) module and the Acquisition & Tracking (A&T) modules. The RF module is responsible for down-converting the incoming GPS signal to an Intermediate Frequency (IF) or baseband and digitizing it for further processing in the downstream modules. In contrast, the A&T modules are responsible for extracting the digital information and pseudorange from the signal.

Under normal conditions, the genuine complex-valued digitized signal, denoted as $r_g[n]$, at the output of RF module can be written as:

$$r_g[n] = \sqrt{P_g}\; D[n-\tau_g]\; C[n-\tau_g]\; \exp\left(j\phi_g\right), \tag{1}$$

here, P_g is the received power of the genuine signal, n is the sample index, $D[\cdot]$ is the BPSK modulated navigation data, $C[\cdot]$ is the BPSK modulated Pseudorandom Noise (PRN) spreading code, τ_g and ϕ_g are the code phase and carrier phase (in radians), respectively. All of these variables are time-varying, however, for notational purposes, this effect has not been shown here. Moreover, as the navigation data is merely 50 bps, we can assume $D[n] = 1$ without loss of generality and ignore it [4].

The above signal is the input to the A&T module which uses the PRN codes of each GPS satellite vehicle to check which ones are currently present in the signal with high enough power by comparing the respective satellite signal power with a pre-defined threshold. For the ones that clear this threshold, their respective coarse code delays and Doppler shifts are estimated. Subsequently, the tracking module uses the coarse estimates to generate the properly aligned PRN code and correlates them with the input signal to track each satellite in parallel. In the presence of an interferer at the input of the GPS receiver, the complex-valued signal samples at output of the tracking correlators, denoted by Ψ_d, can be written as [4]:

$$\begin{aligned} I_d &= \sqrt{P_g}\; R[dT_c]\; \cos(\theta_g) + \sqrt{\gamma P_g}\; R[dT_c - \Delta\tau] \\ &\qquad \cos(\theta_g + \Delta\phi) + \xi_d^I, \\ Q_d &= \sqrt{P_g}\; R[dT_c]\; \sin(\theta_g) + \sqrt{\gamma P_g}\; R[dT_c - \Delta\tau] \\ &\qquad \sin(\theta_g + \Delta\phi) + \xi_d^Q, \\ \Psi_d &= I_d + jQ_d, \end{aligned} \tag{2}$$

here, I_d and Q_d are the in-phase and quadrature components of the correlator's output sample, d is a unit-less real number, T_c is the PRN code chip duration. The dT_c controls the spacing for the early $(d < 0)$, prompt $(d = 0)$, and late $(d > 0)$ correlators. The power advantage of the interferer signal w.r.t. the genuine one is denoted by $\gamma = P_i/P_g$, with P_i representing the interferer signal power over a single integration period. The code delay and carrier phase differences of the interferer relative to the genuine signal are denoted by $\Delta\tau$ and $\Delta\phi$, respectively. Similarly, ξ_d^I and ξ_d^Q are assumed to be zero-mean Gaussian thermal noise components with a constant noise spectral density N_0 present in the I and Q components, respectively. Finally, $R[\cdot]$ is the auto-correlation function of the BPSK signal, that under ideal conditions is written as:

$$R[dT_c] = \begin{cases} 1 - |dT_c|/T_c, & |dT_c| \le T_c, \\ 0, & |dT_c| > T_c. \end{cases} \tag{3}$$

In case of a regular interferer, the $R[\cdot]$ value is very low and does not distort the genuine signals' correlator peak, but only adds to the noise floor, leading to a lower C/N_0. However, in case of a spoofing attack, the attacker carefully aligns $\Delta\tau$ and $\Delta\phi$ parameters in order to overlap the genuine signals' correlator peak. With carefully changing γ and $\Delta\tau$ and $\Delta\phi$ values, the spoofer can then fool the tracking correlator into tracking its signal instead. Such an attack is called the carry-off spoofing attack.

B. Feature Extraction

In this work, we aim to utilize the feature set proposed in [5], which consists of seven features that can be computed in real-time from any commercial GPS receiver. Two features, the receiver in-band power (P_r) and C/N_0, are static in nature and sensitive to the presence of high-powered interferers or spoofed signals [13]. The remaining five features are sensitive to the subtle distortions observed in the tracking correlator's profile during carry-off spoofing attacks. By combining

both power and distortion monitoring features, the resulting trained model is effective against simplistic attacks, where the spoofer's power advantage is significant, as well as intermediate and sophisticated attacks, where the spoofer's power advantage is minimal. Below, we summarize the computations required to compute these features.

1) Received Power: For the civilian L1 GPS band, most of the signal power is concentrated within a 2 MHz band around the L1 carrier frequency of 1575.42 MHz. Therefore, we filter the RF output signal using a 2 MHz bandwidth filter. Let $y_{RF}[n]$ represent the complex-valued baseband samples at the output of the receiver's RF block. We pass these samples through a low-pass filter to obtain a filtered version, $\tilde{y}_{RF}[n]$. The received power (in dBW) within a given time interval can then be computed as follows:

$$P[m] = 10\,log_{10}\left(\frac{1}{N}\sum_{n=(m-1)N+1}^{mN}|\tilde{y}_{RF}[n]|^2\right), \quad (4)$$

here N refers to the total number of samples in the time window. The initial power value (in dB) is subtracted from the entire vector to create a relative received power feature.

2) Carrier to Noise Ratio: The C/N_0 is a key metric to measure the received GPS signals' strength. However, direct measurement of C/N_0 is not possible and requires estimation. To estimate this metric, we use the well-known Narrow-band Wide-band Power Ratio (NWPR) method [2].

3) Signal Quality Monitor: In order to capture the correlation peak distortion during the spoofed and authentic signal interaction, we use 5 different SQM metrics as input features. These metrics are given below:

Ratio Metric [25]: $$m_{ratio} = \frac{I_{-d} + I_{+d}}{I_P}, \quad (5)$$

Delta Metric [25]: $$m_{delta} = \frac{I_{-d} - I_{+d}}{I_P}, \quad (6)$$

Early Late Phase (ELP) Metric [26]:

$$m_{elp} = \tan^{-1}\left(\frac{Q_{-d}}{I_{-d}}\right) - \tan^{-1}\left(\frac{Q_{+d}}{I_{+d}}\right), \quad (7)$$

Symmetric Differences [4]: $$m_{sd} = \frac{|\psi_{-d} - \psi_{+d}|}{\sigma_{N_0}}, \quad (8)$$

Manfredini Metric [13]: $$m_{fred} = \frac{|E_x - L_x|}{|\psi_P|}. \quad (9)$$

Here $I_{\pm d}$, $Q_{\pm d}$, and $\psi_{\pm d}$ are taken from (2) by setting $d = 0.5$. I_P and ψ_P are the respective prompt correlator ($d = 0$) values. m_{fred} is computed using 9 equi-spaced correlators between $d = [-0.1016, 0.1016]$, with L_x and E_x being the linear combination of complex values from late and early correlator fingers, respectively. Finally σ_{N_0} is the noise power, which is the standard deviation of ψ_{-2} during the spoof free case.

Except for the m_{fred} feature, computing the rest does not necessitate any additional hardware. However, to compute m_{fred}, we must add 8 additional correlators positioned very close to the prompt correlator. As a result of this configuration, the computed feature can capture subtle correlation peak distortions when the spoofer's power advantage is minimal or when it has precisely matched the carrier phase, as is the case with sophisticated spoofing attacks. In the experimental section, we demonstrate that including the m_{fred} feature in the training set leads to approximately a 17-24% improvement in spoofing detection accuracy for the sophisticated attack scenario (DS-7) [24]. Apart from the sophisticated attack scenario, even when excluding m_{fred} from the training feature set, the proposed model achieved a detection accuracy of 99% for the remaining attack scenarios. Thus, while detecting regular spoofing attacks can be accomplished using generic GPS receivers, if the application in question is highly sensitive and there is a possibility of sophisticated attacks, having a custom GPS receiver capable of computing the m_{fred} feature can be advantageous in enabling the model to effectively detect such attacks with high precision.

III. Proposed Spoofing Detection Framework

In typical ML-based detectors, models are trained within a supervised learning framework, where the training set is expected to encompass data representing all possible classes that the model may encounter during deployment. Accumulating such a comprehensive training dataset is particularly challenging, especially in the context of GPS spoof detection. Researchers must strive to cover various environments and numerous attack scenarios to enable the trained model to perform effectively under diverse operational conditions. However, there is a risk that the model may struggle to reliably identify samples dissimilar to all the training class samples [27]. Our objective is to train a model that excels not only on data similar to the training samples but also on entirely novel samples. Unsupervised training frameworks such as representation learning can aid in training such models [28]. In this approach, models are trained exclusively on a single known class—in our case, genuine GPS data samples—and subsequently utilized to classify any test input based on its similarity to the distribution of genuine data.

A. The Input Time-Window

As discussed earlier, in this work, instead of performing spoofing detection on individual data points in isolation, we propose a window based spoofing detection framework which can take advantage of additional information that is embedded in the time-series in the form of temporal correlations, hidden temporal dynamics, and feature interactions over time. In this framework, the spoofing detection is performed over multiple concurrent time samples, combined as a time window, where each time point consists of 7 features acquired from the RF and tracking modules of a typical GNSS receiver, as discussed in Section II-B.

Consider a time-series denoted by $\mathcal{X} \in \mathbb{R}^{n\times k}$, where n is the number of time points, and k is the number of features. The model input will then be a window of concurrent time points

TABLE I
THE VAE MODEL PARAMETERS.

	Layers	Input Size	Output Size	Input Ch	Output Ch	Kernel	Padding	Dilation	Stride	Activation
Encoder	Conv1D_1	100	100	7	32	5	2	1	1	PReLU
	Conv1D_2	100	98	32	64	3	1	2	1	PReLU
	MaxPool1D	98	32	-	-	3	0	1	3	-
	Linear_μ	2048	20	1	1	-	-	-	-	-
	Linear_Σ	2048	20	1	1	-	-	-	-	-
Decoder	ConvTrans1D_1	20	26	1	64	7	0	1	1	PReLU
	ConvTrans1D_2	26	53	64	32	5	1	1	2	PReLU
	ConvTrans1D_3	53	105	32	7	3	1	1	2	-

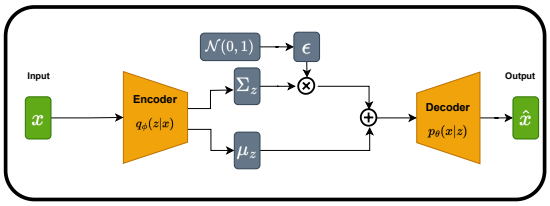


Fig. 2. The VAE model architecture.

taken from $\mathcal{X}$, denoted as $\mathbf{X}_j = [\mathbf{x}_j, \mathbf{x}_{j+1}, \ldots, \mathbf{x}_{j+N-1}]^\top \in \mathbb{R}^{N\times k}$, where each $\mathbf{x}_i \in \mathbb{R}^k$ is the feature vector at time point i and N is the window size. The time index $j \in [0, (1-\eta)N, (1-\eta)2N, \ldots]$ represents the specific starting sample index taken from $\mathcal{X}$, and $\eta \in [0, 1]$ is the window overlap parameter. For instance, $\eta = 0.75$ means that the concurrent sample windows will have 75% samples overlap.

B. The Base Learner

The base learner used in our detection model consists of the 1D convolutional and transposed convolutional layers (1D-ConvNets). By using the convolutional layers, instead of fully connected layers, we aim to fully capitalize on the information available in the multivariate sample time window. Instead of only looking at the samples in isolation, as done by the fully connected layers, the convolutional layers are effective at capturing the local temporal patterns and dependencies within in the sequential data [29]. These layers have translation-invariance property, where they can detect patterns regardless of their exact location within the windowed data, and they can capture interactions between different features. One major advantage of 1D-ConvNets is that they are much more computationally efficient as compared to the fully connected layers due to having fewer parameters, which indirectly, helps with reduction of data overfitting risk as well [29]. These qualities of 1D-ConvNets are particularly advantageous for our work, given that our input is multivariate time-series data. By leveraging 1D-ConvNets, we can effectively capture and learn patterns, trends, and relationships within this sequential data. Furthermore, their lower computational demands make 1D-ConvNets highly suitable for integration into resource-constrained GPS receivers.

C. The VAE Model

The proposed model is comprised of two separate network models, the Encoder ($\mathcal{E}_{\boldsymbol{\phi}}$) and the Decoder ($\mathcal{D}_{\boldsymbol{\theta}}$), which contain multiple 1D convolutional layers, which are connected and trained under the VAE framework [30]. The model architecture is shown in Fig. 2 and their exact layer and construction information is given in Table I. The model was created in Python using the PyTorch library.

The Encoder model takes $\mathbf{x} \in \mathbb{R}^{100\times 7}$ containing a multivariate time-series window as input and passes it through two `Conv1D` layers, followed by a `MaxPool1D` layer to reduce the size from 98 sample points to 32. The output of `MaxPool1D` layer is flattened into a 1D vector of size 2048, which is then passed through two `Linear` layers to generate the mean vector $\boldsymbol{\mu}_\mathbf{z}$ and a variance vector $\boldsymbol{\Sigma}_\mathbf{z}$, who together define a latent probability distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, parameterized by the encoder. During training, the objective is to enable the encoder to map its inputs into a continuous latent space, where each input corresponds to a region in the latent space rather than a single point [30]. This facilitates the subsequent generation of samples by the decoder (discussed next), which are not exact replicas of their input counterparts.

The decoder network takes the real valued latent variable $\mathbf{z}$ as input, which is sampled from $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ using the reparameterization trick [30] given as:

$$\mathbf{z} = \boldsymbol{\mu}_\mathbf{z} + \boldsymbol{\epsilon} \odot \boldsymbol{\Sigma}_\mathbf{z}, \tag{10}$$

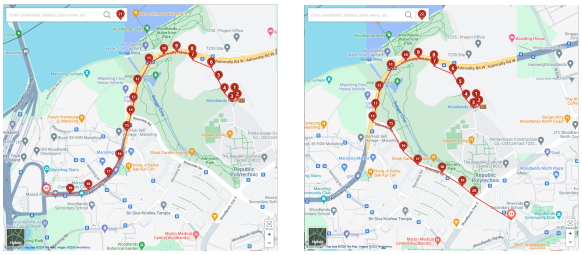
where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ represents element-wise multiplication. This enables gradient backpropagation through the sampling process into the encoder model while preserving the probabilistic nature of the sampling. Using $\mathbf{z}$, the decoder is trained to reconstruct the original input $\mathbf{x}$ by sampling from the distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})$ (parameterized by the decoder). Under the VAE framework, the encoder and decoder networks are trained by maximizing the following loss function

$$\begin{aligned}\mathcal{L}_{VAE}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}, \mathbf{z}) = \mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}(log\ p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})) \\ - D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})||p_z(\mathbf{z})).\end{aligned} \tag{11}$$

Equation (11) is also called the variational lower bound [30]. The first term in (11) is the log likelihood function and the second term represents the latent loss which uses Kullback-Leibler Divergence (KLD) between the learned distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ and a prior $p_z(\mathbf{z})$. Here, the KLD is used as a regularizer to induce some structure onto the latent distribution. In its current form, $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is intractable and the KLD between the latent and the selected prior needs to be estimated [30]. However, if we assume $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ to be Gaussian with an approximately diagonal covariance, and $p_z(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ to be a unit Gaussian, the KLD can be computed without estimation, with the solution given in [30].

D. The Detector

After training, a well-trained VAE should proficiently reconstruct class samples it was trained on, while failing to accurately reconstruct out-of-distribution samples. This fundamental principle forms the basis of our detector setup, where the model is trained solely using genuine GPS features. Consequently, when a test sample window $\mathbf{x}$ is fed through the $\mathcal{E}_\phi$ model, it generates a latent variable $\mathbf{z}$ (computed using



(WL-1) (WL-2)

Fig. 3. The simulated route in Woodlands area of Singapore.

TABLE II
SPOOFING ATTACK SCENARIOS SUMMARY

	TEXBAT								Simulated		
	DS-GS	DS-GD	DS-2	DS-3	DS-4	DS-5	DS-6	DS-7	WL-1	WL-2	WL-C
Power Adv. γ [dB]	-	-	10	1.3	0.4	9.9	0.8	Matched	-	-	3
Difficulty Level	-	-	$\star$	$\star\star$	$\star\star$	$\star$	$\star\star$	$\star\star\star$	-	-	$\star\star$
RX Platform	St	Dy	St	St	St	Dy	Dy	St	Dy	Dy	Dy
Duration [sec]	300	300	300	300	350	300	350	451	120	120	120
Spoofer Onset [sec]	-	-	110	120	114	102	105	110	-	-	40

DS-GS: Genuine Static, DS-GD: Genuine Dynamic.
Difficulty Levels - Simplistic: $\star$, Intermediate: $\star\star$, Sophisticated: $\star\star\star$.
RX Platform - Static: St, Dynamic/Mobile: Dy.

(10)). Subsequently, $\mathbf{z}$ is passed through $\mathcal{D}_\theta$ to produce the reconstructed sample window $\hat{\mathbf{x}}$. For genuine samples, the model should aptly reconstruct them. Conversely, for spoofed samples, the model is expected to struggle in replicating them, as it was not exposed to such samples during training.

Under this premise, we calculate a test statistic for a given test sample window $\bar{\mathbf{x}}$ and its reconstruction $\hat{\bar{\mathbf{x}}}$ as $\zeta = \sqrt{\mathbb{E}\left[(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}})^2\right]}$, where the expectation is taken across feature dimension. Subsequently, employing a predefined threshold ρ, if $\zeta < \rho$, the test sample is classified as genuine; and spoofed otherwise.

Selecting an appropriate threshold is crucial for the detection performance of the model. To this end, the threshold ρ is determined based on the acceptable False Positive Rate (FPR) tolerance, typically provided in the detector's design specifications. After training the model, we calculate the test statistic ζ for the entire training dataset and determine their respective thresholds. These thresholds are set to correspond to the value that results in specified FPR over the training dataset. To summarize, when presented with a test sample $\bar{\mathbf{x}}$, the detection criteria will classify it as spoofed if

$$\zeta(\bar{\mathbf{x}}) \triangleq \sqrt{\mathbb{E}\left[(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}})^2\right]} \geq \rho. \tag{12}$$

In the upcoming experiments, we will analyze the detector's performance under multiple input FPRs.

IV. EXPERIMENTAL EVALUATION

A. Dataset Description

In this study, we use a simulated and a real-world dataset, called TEXBAT, for performance analysis of the proposed spoof detection framework. The TEXBAT dataset is the default testing dataset for GPS spoofing detection frameworks as it contains GPS spoof attacks of different difficulty levels, from simplistic to sophisticated attacks, covering static to dynamic scenarios [6]. The dataset is made publicly available by the Radio Navigation Laboratory (RNL) of the University of Texas in Austin. The digitized signals have a bandwidth of 20 MHz, with 16 bit quantization with a complex sampling rate of 25 Msps. We use the two clean and 6 attack datasets from TEXBAT in our analysis. For details on the threat model and simplistic to sophisticated attack types, please see [6], [24].

The simulated dataset consists of two 120 second spoof free GPS signal recordings, termed WL-1 and WL-2, whose mobility profile was generated using the SATGENv3[1], as shown in Fig. 3. These routes were converted into the simulated GPS signals using the gnss-sim-sdr[2] open source software. The simulated route is located in the Woodlands area of Singapore and the maximum speed allowed was set to 100 Km/h. To generate the spoofing attack, we considered WL-2 to be the spoofed trajectory and WL-1 as the genuine one. Consequently, we generated the spoofed dataset, called WL-C, by adding WL-2's recording to the WL-1 at approx. 40 seconds, with $\gamma = 3$ dB power advantage to simulate the spoofer's signal appearing at the target receiver. Subsequently, the spoofed signal was able to take over the target receiver's tracking loop. Both spoofed and genuine signals remain relatively close till 70 seconds, at which point, the WL-2's trajectory fully deviates from the WL-1 and goes in another direction. A short summary of the attack scenarios is outlined in Table. II.

B. Feature Preparation and Model Training

The datasets discussed in the previous sub-section are essentially samples acquired at the output of the RF block of the GPS receiver, see Fig. 1. In this work, we use the MATLAB based FGI-GSRx [2] open-source GNSS software receiver to decode the input datasets and perform feature extraction as outlined in Section II-B. We select the satellite with the highest received power for feature extraction at a sampling rate of 50 Hz (averaged over 20 ms time window). Rest of the model training, testing, and analysis were performed in Python.

For efficient learning, training dataset features were scaled to $[0, 1]$ range before training and the same scaling parameters were used to scale the test dataset features as well, ensuring no information leakage between training and testing sets. The proposed Convolutional VAE (C-VAE) model was trained in Pytorch using the model architecture outlined in Table I. The model parameters were trained by minimizing the loss given in (11) for 200 epochs using Adam optimizer with adaptive learning rates of $[10^{-4}, 5e^{-5}, 10^{-5}]$, changing at 70 and 150 epochs. The batch size was fixed at 128 and latent space size was fixed at 20. The input to the model is a time-series window consisting of $N = 100$ time points (2 seconds worth of data), with successive window overlap of 75% ($\eta = 0.75$). To assess model's performance, we used three metrics, i.e., the overall Accuracy (ACC), the True Positive Rate (TPR) or detection

[1]https://www.labsat.co.uk/index.php/en/products/satgen-simulator-software
[2]https://github.com/osqzss/gps-sdr-sim

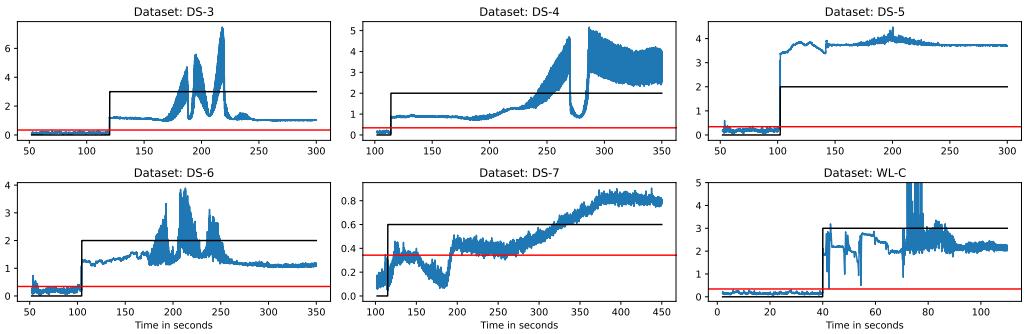


Fig. 4. The model test statistics (in blue) for an input FPR of 10^{-5}, respective threshold ρ is in red and spoofer onset is shown using black lines.

TABLE III
DETECTION PERFORMANCE OF THE PROPOSED DETECTOR.

Input FPR	10^{-2}			10^{-3}			10^{-5}		
Test DS	ACC	TPR	FPR	ACC	TPR	FPR	ACC	TPR	FPR
DS-GS	99.81	-	0.19	100.00	-	0.00	100.00	-	0.00
DS-GD	98.23	-	1.77	99.84	-	0.16	100.00	-	0.00
DS-2	99.76	99.96	0.90	99.95	99.96	0.07	99.97	99.96	0.00
DS-3	99.72	100.00	1.03	99.88	100.00	0.44	99.89	100.00	0.41
DS-4	100.00	100.00	0.00	100.00	100.00	0.00	100.00	100.00	0.00
DS-5	97.94	100.00	10.20	99.50	100.00	2.47	99.78	100.00	1.08
DS-6	97.74	100.00	12.82	99.09	100.00	5.15	99.61	100.00	2.21
DS-7	89.49	89.20	3.08	86.41	85.92	0.77	78.92	78.10	0.00
WL-1	98.15	-	1.85	99.85	-	0.15	100.00	-	0.00
WL-2	98.52	-	1.48	99.96	-	0.04	100.00	-	0.00
WL-C	99.04	99.97	2.69	99.98	99.97	0.00	99.98	99.97	0.00

rate, and False Positive Rate (FPR). The Miss Rate (MR) can be computed directly from TPR.

C. Spoofing Detection

To perform spoof detection, we trained the proposed C-VAE model using the features extracted from spoof free genuine data recordings, i.e., the genuine static (DS-GS) and genuine dynamic (DS-GD) datasets from TEXBAT, see Table II. With a trained model at hand, we can then test the incoming sample windows to deduce whether they are similar to the training class (spoof-free) or are significantly different (spoofed) from it. As discussed in Section III-D, reconstruction error is used as the test statistic ζ to classify each sample of the test window as spoofed or not. The required threshold ρ is computed using the training data under multiple input FPRs of 10^{-2}, 10^{-3}, and 10^{-5}. The resulting detection scores for both TEXBAT and simulated scenarios are presented in Table III.

It is evident from Table III that the detector, trained solely on genuine GPS features, not only excelled on the dataset it was trained on but also successfully detected out-of-distribution spoofed samples across all spoofing attack scenarios. The detection performance for DS-2 to DS-6 and WL-C attack datasets is remarkable. For instance, at an input FPR of 10^{-3}, the detector achieved a TPR as high as 100%, with the lowest being 99.96%. Additionally, it maintained the FPR below 5%, as observed for DS-6. Moreover, as the input FPR decreases, the output FPR for all datasets decreases while maintaining the same TPR. However, the TPR reported for the sophisticated attack dataset DS-7 is relatively modest, reaching 85.92% for an input FPR of 10^{-3}. This is attributed to the sophisticated attacker achieving full alignment with the genuine signal's carrier phase and matching power with respect to the genuine signal. For visualization, we depict the computed test statistics for multiple attack datasets (in blue), along with the detection thresholds (in red), and spoofer onset (in black) in Fig. 4.

1) The Sophisticated DS-7 Attack: From the test statistics depicted in Fig. 4, it's apparent that, except for DS-7, the remaining datasets exhibit well-defined class boundaries. Therefore, reducing the input FPR does not diminish their TPR, as the threshold adjusts accordingly. However, these scenarios differ from the sophisticated attack of DS-7, particularly during the correlator peak pull-off stage, initiated around 150 seconds. During the 150 to 200-second window, severe destructive interference occurs between genuine and spoofed signals, resulting in a drop in feature values to a range similar to that observed in the absence of a spoofer. Nevertheless, once the spoofer gains control of the correlator's peak, the test statistic returns to its higher range, enabling detection of the spoofer's presence with nearly 100% TPR from that point onward.

2) Window-wise Detection: The detection scores presented in Table III and illustrated in Fig. 4 are computed for each individual time sample within the input time-series window. To provide additional insights, we also calculated the detection accuracy across each time-series window of size $N = 100$ and displayed these results in Fig. 5. This analysis reveals a notable drop in detection accuracy for DS-7, particularly between the 26th and 43rd windows, corresponding to the time frame from 152 to 186 seconds (as observed in Fig. 4). However, from the 44th window onwards, the accuracy returns to 100%, with only a minor dip around the 78th window (at approximately 256 seconds). These findings align with the per-sample detection performance discussed earlier, further highlighting the model's robustness and its transient challenges during the sophisticated spoofing attack scenario.

3) The m_{fred} Feature Importance: Out of the 7 selected features, the most computationally expensive feature is the m_{fred} [13] which requires an extra 8 correlators in the tracking module for computation. The rest of the features do not require any additional computations or hardware and are readily

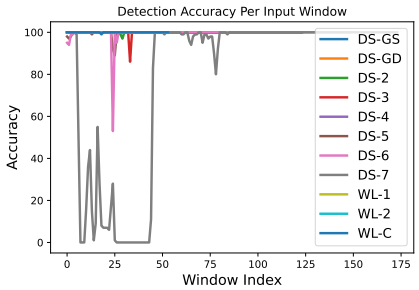


Fig. 5. Overall accuracy over each time window for input FPR of 10^{-5}.

computed by the GPS receivers for their routine operations. To estimate the effectiveness of m_{fred} feature, we retrained our model after removing m_{fred} from the training featureset. The detection performance of this model was very similar to the ones reported in Table III for every attack dataset, except for the sophisticated DS-7 attack. For comparison, we report the results of models trained with and without m_{fred} feature for DS-7 in Table IV, where we can see that the m_{fred} feature alone amounts to approx. 24% improvement in the TPR for the multiple input FPRs. This is expected as the m_{fred} feature is designed to be sensitive to very subtle correlator peak distortions, as it uses correlators placed very close to the center (prompt) correlator at both sides, and thus was able to capture the subtle distortions created in the DS-7 attack.

From the above discussion, it can be inferred that if ensuring the correct operation of downstream applications reliant on the GPS receiver is crucial, augmenting the receiver with additional correlators could be warranted to ensure detection of sophisticated attacks. However, for everyday use cases where the incentive to carry out a sophisticated attack is minimal, a model trained on the remaining six features would still be capable of detecting other types of attacks. Moreover, when running on a desktop PC with an Intel Core i7-13700K and an Nvidia RTX 3080, our framework performs detection on a single time window within 3-5 ms (200-333 Hz).

4) Training Data Diversity: A fundamental prerequisite for our framework is the availability of high-quality, spoof-free GPS data for training the model. In our analysis, we utilized the static and dynamic genuine datasets from the TEXBAT dataset, comprising real-world signals. The static dataset was recorded under clear skies, while the dynamic recordings were conducted in urban settings [6], where satellite visibility was occasionally obstructed, leading to prominent multipath effects in these signals. By training the models using these diverse and genuine signals, our model became aware of the presence of various interfering components alongside the GPS signal. Furthermore, the features employed were generic and not specific to any particular geographical area. Consequently, when evaluated on simulated datasets generated in entirely separate geographical areas, the model demonstrated nearly 100% classification accuracy (see Table III).

However, if the training data lacks wireless interference and is excessively clean, e.g., simulated, the resulting model may underperform. To assess this possibility, we trained the model using features extracted from the WL-1 and WL-2 simulated dynamic datasets and tested it on all dynamic datasets from TEXBAT. The results are summarized in Table V. While the model exhibited excellent detection performance on simulated data, the FPRs for the real-world datasets from TEXBAT were notably high, indicating that the model was overly conservative in classifying samples as genuine for these datasets due to the presence of other contaminants. This underscores the importance of having a meticulously curated real-world training dataset to ensure robust spoof detection.

5) Comparative Study: In this section, we provide a concise comparison of the proposed method with other similar ML and statistical hypothesis-based detection methods, including those described in [4], [9], [10], [13], [23]. The reason for selecting these methods for comparison is that they also use features accessible at the tracking stage of a standard GPS receiver, such as received power, C/N_0, and multiple correlation finger outputs. For this evaluation, we concentrate on the DS-7 scenario dataset, representing the most challenging scenario among all the TEXBAT datasets. All models were trained using the default configuration provided in their respective papers. The results are summarized in Table VI.

Upon analyzing the TPR across all input FPRs, [9] demonstrates the lowest performance. This outcome is expected given its transient nature, relying on conventional SQM metrics that are sensitive solely during the correlator peak pull-off phase. Once the pull-off exceeds 1 chip length, the TPR of [9] rapidly declines to zero. In contrast, [10] shows high sensitivity to correlation peak distortion, swiftly detecting the peak pull-off owing to the utilization of multiple equidistant correlators. However, the TPR of [10] is approximately 10 points lower than that of the proposed method across all input FPRs.

Although both [4] and [13] employ SQM metrics, they exhibit lackluster performance in the sophisticated attack scenario of DS-7. This can be attributed to the spoofer's minimal power advantage and their ability to achieve frequency matching with the genuine GPS received signal. Although, the

TABLE IV
PERFORMANCE GAIN OFFERED BY THE m_{fred} FEATURE FOR DS-7.

Input FPR	10^{-2}			10^{-3}			10^{-5}		
m_{fred} ?	ACC	TPR	FPR	ACC	TPR	FPR	ACC	TPR	FPR
No	68.76	67.66	2.62	63.21	61.82	0.77	55.78	54.08	0.31
Yes	89.49	89.20	3.08	86.41	85.92	0.77	78.92	78.10	0.00

TABLE V
PERFORMANCE OF THE MODEL TRAINED ON SIMULATED SCENARIOS.

Input FPR	10^{-2}			10^{-3}			10^{-5}		
Test DS	ACC	TPR	FPR	ACC	TPR	FPR	ACC	TPR	FPR
DS-1	5.68	0.00	94.32	41.60	0.00	58.40	61.27	0.00	38.73
DS-5	79.75	100.00	100.00	79.90	100.00	99.28	80.56	100.00	96.02
DS-6	82.50	100.00	99.50	83.21	100.00	95.46	84.50	100.00	88.10
WL-1	98.74	0.00	1.26	99.91	0.00	0.09	100.00	0.00	0.00
WL-2	98.72	0.00	1.28	99.96	0.00	0.04	99.98	0.00	0.02
WL-C	99.80	99.97	0.53	99.98	99.97	0.00	99.98	99.97	0.00

TABLE VI
COMPARATIVE RESULTS ON TEXBAT SCENARIO DS-7.

Input FPR →	5×10^{-2}			10^{-2}			10^{-3}		
Methods ↓	ACC	TPR	FPR	ACC	TPR	FPR	ACC	TPR	FPR
Wesson *et al.* [4]	71.25	73.96	8.56	70.76	69.4	5.08	70.18	67.73	1.98
Manfredini *et al.* [13]	69.35	68.34	11.68	63.22	65.87	7.67	59.68	62.18	3.58
Sun *et al.* [9]	56.34	57.68	8.64	53.84	54.08	5.68	51.35	52.54	1.28
Zhou *et al.* [10]	78.55	81.65	9.32	79.89	79.35	6.58	77.76	75.97	2.85
Iqbal *et al.* [23]	93.64	93.36	0.00	91.04	90.65	0.00	78.13	76.74	0.00
Proposed	93.13	93.14	7.32	90.49	90.20	3.08	86.41	85.92	0.77

results of the proposed method closely resemble those reported for [23], however, as the input FPR decreases to 10^{-3}, the proposed method's ACC and TPR scores outperform [23]. Moreover, the 1D-ConvNets used here are computationally efficient than the FCNNs used there.

V. CONCLUSION

In conclusion, this paper addresses the critical challenge posed by spoofing attacks on GPS receivers, which are increasingly integrated into various aspects of our daily lives. The proposed solution leverages features that can be extracted in real-time from a standard GPS receiver. Our detection framework is built upon a Convolutional-VAE model, trained on spoof-free multivariate time-series data, capable of detecting spoofing at each sample point or over a time window. The proposed detector underwent rigorous testing against a spectrum of simulated and real-world spoofing attacks, spanning from static to dynamic receivers and from simplistic to sophisticated spoofing techniques. Across multiple input FPRs, our detector achieved up to 100% detection accuracy for simplistic and intermediate spoofing attacks. Even for the sophisticated attack (DS-7), it achieved a respectable TPR of 85% for an FPR of 0.77%. Furthermore, our investigation reveals that for regular spoofing attacks, we can improve computational efficiency by excluding the computationally expensive m_{fred} feature without sacrificing performance.

REFERENCES

[1] "Critical infrastructure sectors," tech. rep., US Cybersecurity & Infrastructure Security Agency, 2021. Accessed: 2024-02-12.

[2] K. Borre, I. Fernández-Hernández, J. A. López-Salcedo, and M. Z. H. Bhuiyan, *GNSS Software Receivers*. Cambridge University Press, 2022.

[3] F. S. T. Van Diggelen, *A-GPS: Assisted GPS, gnss, and SBAS*. Artech house, 2009.

[4] K. D. Wesson, J. N. Gross, T. E. Humphreys, and B. L. Evans, "GNSS signal authentication via power and distortion monitoring," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 2, pp. 739–754, 2017.

[5] A. Iqbal, M. N. Aman, and B. Sikdar, "Machine and representation learning based GNSS spoofing detectors utilizing feature set from generic GNSS receivers," *IEEE Transactions on Consumer Electronics*, pp. 1–1, 2023.

[6] T. E. Humphreys, J. A. Bhatti, D. Shepard, and K. Wesson, "The texas spoofing test battery: Toward a standard for evaluating GPS signal authentication techniques," in *Radionavigation Laboratory Conference Proceedings*, 2012.

[7] X. Wei, M. Aman, and B. Sikdar, "Exploiting correlation among GPS signals to detect GPS spoofing in power grids," *IEEE Transactions on Industry Applications*, vol. PP, 2021.

[8] D. M. Akos, "Who's afraid of the spoofer? GPS/GNSS spoofing detection via automatic gain control (agc)," *NAVIGATION: Journal of the Institute of Navigation*, vol. 59, no. 4, pp. 281–290, 2012.

[9] C. Sun, J. W. Cheong, A. G. Dempster, H. Zhao, and W. Feng, "GNSS spoofing detection by means of signal quality monitoring (sqm) metric combinations," *IEEE Access*, vol. 6, pp. 66428–66441, 2018.

[10] W. Zhou, Z. Lv, X. Deng, and Y. Ke, "A new induced GNSS spoofing detection method based on weighted second-order central moment," *IEEE Sensors Journal*, vol. 22, no. 12, pp. 12064–12078, 2022.

[11] A. M. Khan, N. Iqbal, A. A. Khan, M. F. Khan, and A. Ahmad, "Detection of intermediate spoofing attack on global navigation satellite system receiver through slope based metrics," *The Journal of Navigation*, vol. 73, no. 5, pp. 1052–1068, 2020.

[12] E. Schmidt, N. Gatsis, and D. Akopian, "A GPS spoofing detection and classification correlator-based technique using the lasso," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4224–4237, 2020.

[13] E. G. Manfredini, D. M. Akos, Y.-H. Chen, S. Lo, T. Walter, and P. Enge, "Effective GPS spoofing detection utilizing metrics from commercial receivers," in *Proceedings of the 2018 International Technical Meeting of The Institute of Navigation*, pp. 672–689, 2018.

[14] D. Miralles, A. Bornot, P. Rouquette, N. Levigne, D. M. Akos, Y.-H. Chen, S. Lo, and T. Walter, "An assessment of GPS spoofing detection via radio power and signal quality monitoring for aviation safety operations," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 136–146, 2020.

[15] 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," *The Journal of Navigation*, vol. 71, no. 1, pp. 169–188, 2018.

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

[17] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Deep neural network approach to detect GNSS spoofing attacks," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 3241–3252, 2020.

[18] J. Li, X. Zhu, M. Ouyang, W. Li, Z. Chen, and Q. Fu, "GNSS spoofing jamming detection based on generative adversarial network," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22823–22832, 2021.

[19] S. C. Bose, "GPS spoofing detection by neural network machine learning," *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 6, pp. 18–31, 2022.

[20] G. Aissou, S. Benouadah, H. E. Alami, and N. Kaabouch, "Instance-based supervised machine learning models for detecting GPS spoofing attacks on uas," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 0208–0214, 2022.

[21] A. Iqbal, M. Aman, and B. Sikdar, "Machine learning based time synchronization attack detection for synchrophasors," in *2023 IEEE Global Communications Conference (GlobeCom)*, pp. 1–6, 2023.

[22] C. Kim, S.-Y. Chang, D. Lee, J. Kim, K. Park, and J. Kim, "Reliable detection of location spoofing and variation attacks," *IEEE Access*, vol. 11, pp. 10813–10825, 2023.

[23] A. Iqbal, M. N. Aman, and B. Sikdar, "A deep learning based induced GNSS spoof detection framework," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 457–478, 2024.

[24] T. Humphreys, "Texbat data sets 7 and 8," *The University of Texas*, 2016.

[25] R. E. Phelts, *Multicorrelator techniques for robust mitigation of threats to GPS signal quality*. Stanford University, 2001.

[26] O. M. Mubarak and A. G. Dempster, "Performance comparison of elp and delp for multipath detection," in *Proceedings of the 22nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2009)*, pp. 2276–2283, 2009.

[27] A. Iqbal, M. N. Aman, and B. Sikdar, "Representation learning based time synchronization attack detection for synchrophasors," in *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–6, 2023.

[28] Y. Wang, J. Zhang, S. Guo, H. Yin, C. Li, and H. Chen, "Decoupling representation learning and classification for gnn-based anomaly detection," in *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pp. 1239–1248, 2021.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, vol. 1. MIT press Cambridge, MA, USA, 2017.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2013.