Representation Learning based Time Synchronization Attack Detection for Synchrophasors

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.
†Corresponding authors
Email: {aiqbal, bsikdar}@nus.edu.sg, naveed.aman@unl.edu

***Abstract*—Phasor measurement units (PMUs) play a crucial role in ensuring the reliable operation of modern power grid monitoring systems, such as wide-area measurement systems (WAMS). These systems heavily rely on accurate time synchronization, typically achieved through the Global Positioning System (GPS). However, the open nature of civilian GPS signals exposes PMUs to potential time synchronization attacks (TSA), where malicious actors manipulate PMU time stamps by transmitting deceptive GPS signals in close proximity to the PMUs. In this paper, we propose a framework for TSA detection using machine learning (ML) models at the control center of a WAMS. The feature set used includes power and correlation distortion measurements, which can be extracted in real-time using any generic GPS receiver. We propose a spoof detector based on representation learning, which offers advantages over supervised ML methods by not requiring exhaustive coverage of all possible attack scenarios during training. Instead, it can be trained using only authentic GPS features. Experimental results highlight that our proposed method performs comparably or even surpasses the performance of the compared ML algorithms. This improvement is particularly evident when considering the TEXBAT subtle attack scenario DS-7, where the ML methods struggle to detect the presence of spoofers. In contrast, our proposed method achieves a detection probability of 98% at a false alarm probability of 2.5%.**

***Index Terms*—Detection technique, GPS spoofing attacks, PMU, representation learning, Smart Grid, TSA.**

I. Introduction

In the realm of modernized electrical distribution grids, such as the smart grid, continuous monitoring plays a vital role in ensuring the stability of the system. Wide-area measurement systems (WAMS) employ phasor measurement units (PMUs) to facilitate real-time monitoring of synchrophasors [1] that are a measurement techniques utilized in power systems to capture both the magnitude and phase angle of a sinusoidal voltage/current waveform at a specific moment, enabling real-time monitoring and control of the power system dynamics. The sampling frequency of these values typically ranges from 10 to 50 Hz, depending on the distribution system's requirements.

This work was supported in part by National Cybersecurity R&D Lab, Singapore under grant NCL-2022-01.

Given that PMUs are distributed across different geographical locations, it is crucial to synchronize their generated samples with Coordinated Universal Time (UTC) for temporal alignment. Although minor timing errors have negligible impacts on PMU measurements, significant timing discrepancies can have severe consequences as stated in the IEEE-C37.118 standard. For example, a time stamp error exceeding 26.5 μ-seconds could potentially lead to power grid blackouts [1]. To ensure precise synchronization, PMUs rely on the timing signal provided by the Global Positioning System (GPS), which offers reliable and highly accurate time references. The GPS timestamps, along with the synchrophasor measurements, are transmitted to the control center for system status analysis and to devise appropriate control measures.

PMUs obtain the GPS timing signal either through the public L1 channel or the new L1C channel [2]. Unlike the encrypted P(Y) military channel, these signals are unsecured and follow open standards. Additionally, the GPS ephemeris and satellite data are publicly accessible. As a result of being low power and open-standard, these signals are vulnerable to various sources of radio frequency (RF) interference, including intentional or unintentional interference. Intentional interference can be categorized as jamming or spoofing interference, where jamming interference involves high-power noisy transmissions in the GPS L1 band that completely overpower the authentic signal. Although this type of interference is relatively easier to detect, spoofing interference can be incredibly subtle and poses a significant threat to GPS-dependent applications.

The availability of programmable simulators and software-defined radios has significantly facilitated the act of spoofing GPS signals. Spoofing attacks can be classified into three categories based on their implementation complexity; simplistic, intermediate, and sophisticated [3]. Among the various intermediate spoofing techniques, induced spoofing (also known as carry-off spoofing) is the most prevalent and detrimental. It involves capturing the receiver's code tracking loop by transmitting a counterfeit signal precisely matched to the frequency and code phase of the genuine signal. Once the correlation peak at the targeted receiver aligns completely, the spoofer gradually gains control over the tracking loop

by manipulating the power and code rate of the counterfeit GPS signal [4]. In the absence of effective spoofing countermeasures at the targeted receiver, it remains unaware of the manipulation and continues to lock onto the spoofed signal. As a result, spoofer can manipulate the timing and positional information embedded in the GPS signals, leading to significant downstream consequences.

In the context of smart grids, an adversary can launch a time synchronization attack (TSA) by manipulating the timestamps generated by the GPS module integrated into a PMU. This form of attack induces a phase angle shift in the PMU measurement, causing erroneous state estimations and generating incorrect control decisions at the control center, potentially resulting in large-scale power outages [1], [5].

Ensuring the reliable and secure operation of power grids necessitates TSA detection. Several approaches have been proposed to tackle this issue, including the utilization of multiple PMUs, generator data, GPS signal statistics, and machine learning (ML) algorithms. Sabouri et al. [6] employed the rotor angles of different generators to train a multi-layer perceptron (MLP)-based spoofing detector. Zhu et al. [7] developed a detection method that relied on the PMU locations and GPS signal statistics at the receiver. Fan et al. [8] proposed the use of multiple PMUs and the characteristics of various power grid sensors for TSA detection. Xie et al. [9] utilized the terminal voltage, rotor speed and angle of generators to propose a quasi-dynamic estimator for spoofing detection. Huang and Li [5] trained a vector neural network by employing the phase coding of PMU measurements to extract and encode the relationships between their phase and magnitude for TSA detection.

Various techniques that rely on GPS signal statistics and receiver properties have also been proposed, categorized as authentication-based, multiple-antenna-based, inertial- and sensor-based, and single-antenna-based methods [4]. The most practical and widely deployable approaches are those that require no additional hardware and can be implemented through a software or firmware update. These methods include monitoring the received power via automatic gain control (AGC) [10], monitoring the receivers' autocorrelation profile of the tracking loop [3], [11], and a combination of both [4], [12]. These techniques employ Bayesian detection framework.

In recent times, several supervised ML algorithms have also emerged as viable options for GPS spoofing detection. These methods leverage features obtained from various blocks of a typical GPS receiver, including RF, acquisition, tracking, and position-velocity-time (PVT) blocks. Studies highlighted in [13]–[16] have demonstrated the effectiveness of these algorithms, achieving rates as high as 95% in correctly identifying the counterfeit signals. Notably, these ML methods offer multiple advantages. They eliminate the need for meticulous selection of signal models and prior distributions for authentic and spoofed signals. Furthermore, they can be trained in a data-driven manner, providing flexibility and adaptability. Additionally, ML techniques based on artificial neural networks (ANN) have proven successful in tackling complex problems without relying on intricate signal modeling.

When dealing with time-series classification, however, supervised ML models face certain challenges. For instance, they have a tendency to over-fit when the dataset size is small or when testing samples closely resemble those in the training set. Moreover, supervised learning models often exhibit poor performance when tested on datasets that significantly differ from the training data. In this paper, we initially highlight that the exceptional performance observed in supervised ML methods [13]–[16] can be attributed to the close resemblance between the test set and the training set. Additionally, we demonstrate that these methods may struggle in detecting unseen spoofing attacks.

To address this issue, we propose a representation learning based GPS spoofing detector. This detector is trained on clean and authentic datasets, enabling it to capture the essential underlying structural information. Consequently, the trained detector can effectively detect spoofing even in datasets it has not encountered before. Notably, our proposed detector demonstrates robust performance against sophisticated spoofing attacks. Furthermore, the training feature set comprises of a combination of signal power measurements and signal quality monitoring (SQM) metrics acquired in real-time from the RF and tracking stages of a generic GPS receiver.

Our proposed detection framework involves the transmission of both normal measurement data and computed GPS receiver features from each PMU to the control center. The control center utilizes these features to train supervised ML or representation learning-based detectors for spoof detection. Once trained, the control center can assess the authenticity of the received GPS features from the PMUs. In the event of a positive spoof detection, the control center can employ computationally intensive TSA mitigation techniques, such as those outlined in [17], to safeguard against the use of counterfeit time stamps in power grid control.

The paper's structure is as follows. In Section II, we begin by introducing the TEXBAT dataset, then proceed to detail the approach for feature extraction, and outline the employed ML models. Section III discusses the proposed GPS spoof detection model, followed by the explanation of the detection framework. The performance evaluation of both the ML models and the proposed representation learning-based detector is presented in Section IV. Finally, the concluding remarks are provided in Section V.

II. Methodology

A. TEXBAT Dataset

The primary aim of this research is to detect instances of GPS spoofing in the GPS signals used by PMUs located at grid stations or generation points. To accomplish this, we employed the Texas Spoofing Test Battery (TEXBAT) dataset[1], which is publicly available and consists of binary recordings of various spoofing scenarios performed on civil GPS L1 C/A signals [18]. The recorded samples in the TEXBAT dataset have the following specifications: 16-bit resolution, centered

[1]https://radionavlab.ae.utexas.edu/texbat/

at a carrier frequency of 1575.42 MHz, a bandwidth of 20 MHz, and a complex sampling rate of 25 Msps. Given that PMUs are consistently situated at fixed locations and rely on timing data derived from GPS signals, and recognizing that the TSA seeks to manipulate these timestamps to desynchronize synchrophasor measurements, our study zeroes in on particular *static* TEXBAT datasets that simulate *time push attacks*. These include *DS-0*, containing authentic GPS signals; *DS-2*, featuring a spoof signal with a high power advantage of 10 dB; and *DS-3*, containing a spoof signal with a small power advantage of 1.3 dB. In both of these datasets, the objective of the spoofer is to manipulate the measured GPS time of the target receiver by gaining control of its tracking loop. Additionally, we included the *DS-7* dataset, which is similar to *DS-3*, but is the most challenging as it not only has relatively matched power, but it also performs carrier phase alignment.

For our analysis, we selected signals from 50 to 300 seconds from *DS-0, DS-2*, and *DS-3*, and 100 to 450 seconds from *DS-7*. This selection limits the dataset size while covering the timeframe during which the spoofed signal is introduced and takes control of the target receiver's tracking loop.

B. Feature Extraction

To analyze the GPS signals, we used a MATLAB based open-source, single antenna based GNSS receiver called FGI-GSRx [19]. Our feature set comprises of 6 run-time measurements which can be generated using the output of the RF and tracking blocks of any generic GPS receiver. To ensure stable measurements, we averaged each feature over a 20 ms time window, taking into account the high signal sampling rate of the receiver. This approach allowed us to achieve a sampling rate of 50 Hz while maintaining the measurement reliability.

- *Received Power:* As, for the case of GPS L1 band, most of the signal power is centered at the L1 carrier frequency in a small bandwidth of 2 MHz. Let $y_{RF}[n]$ be the complex-valued baseband samples at the output of the receivers' RF block. We pass it through a low pass filter with a bandwidth of 2 MHz to get a filtered version $\tilde{y}_{RF}[n]$, then for a given time interval, received power (in dBW) can be computed as
$$P[k] \triangleq 10\,log_{10}\left(\frac{1}{N}\sum_{n=(k-1)N+1}^{kN}|\tilde{y}_{RF}[n]|^2\right) \quad (1)$$
here N is the number of samples in a 20 ms time window.
- *Carrier to Noise Ratio (C/N_0):* Another power measurement feature comes from C/N_0, however, directly measuring C/N_0 is not feasible and requires estimation. In our study, we utilize the widely recognized Narrowband Wide-band Power Ratio (NWPR) method [19, 1.7.3] to estimate this metric. By considering both received power and C/N_0, we enable the trained classifier to effectively differentiate between genuine interference and cases of spoofing [12].
- Ratio Metric [20]
$$m_{ratio} \triangleq \frac{I_{-d}+I_{+d}}{I_0} \quad (2)$$
- Delta Metric [20]
$$m_{delta} \triangleq \frac{I_{-d}-I_{+d}}{I_0} \quad (3)$$
- Early Late Phase Metric [21]
$$m_{elp} \triangleq \tan^{-1}\left(\frac{Q_{-d}}{I_{-d}}\right)-\tan^{-1}\left(\frac{Q_{+d}}{I_{+d}}\right) \quad (4)$$
- Symmetric Differences [4]
$$m_{sd} = \frac{|\psi_{-d}-\psi_{+d}|}{\sigma_{N_0}} \quad (5)$$

here I_d and Q_d are the tracking correlators' In-phase and Quadrature components. We consider three correlators synced as prompt ($d=0$), early ($d<0$), and late ($d>0$) correlators. Our early and late correlators were kept at $d=0.5$ chips. ψ is the complex-valued correlator output and σ_{N_0} is the standard deviation of ψ_2 samples in the spoof free case.

We chose four different SQM metrics due to their complementary nature as each metric provides unique insights. For instance, when the value of m_{elp} is large, we tend to observe smaller values for m_{delta} and m_{ratio}, as noted in [11]. Additionally, the metric m_{sd} captures the absolute difference between early and late correlators, scaled by the standard deviation of the noise (in the absence of spoofing). Notably, this metric exhibits high values during and after the complete takeover of the receiver's correlator peak. The selection of these metrics allows us to gather comprehensive information about the spoofing scenario and its impact on the GPS receiver.

The feature extraction process resulted in a dataset denoted as $\mathbf{X} = \mathbb{R}^{55000\times 6}$, with 36% genuine and 64% spoofed samples. Each feature was scaled to the range of [0,1] to facilitate ML.

C. Machine Learning Models

In our paper, we incorporated three extensively utilized and effective ML models as base learners: the Random Forest classifier (RFC), Support Vector Machines (SVM), and Artificial Neural Network (ANN). These models have also demonstrated successful application in GPS spoof detection in prior works such as [13], [14], [16]. For their implementation detail, reader is referred to [22]. To ensure optimal performance, we fine-tuned the hyperparameters of each model using a grid search method. For the RFC model, we trained it with 11 estimators, while the SVM model utilized the radial basis function (*RBF*) kernel. The ANN model was constructed with 3 fully connected layers, consisting of $20-10-1$ nodes respectively. The first two layers employed the PReLU activation function, while a sigmoid function was applied at the final node.

III. Representation Learning Based Detector

In supervised learning, ML-based detection models are trained using data instances from both spoofed and authentic classes. On the other hand, representation or profiling-based detection models are trained solely on authentic data instances, treating it as a single class. These trained models can then be used to determine if a test sample belongs to the same class

it was trained on or not. The objective is to develop a robust model that can handle various attack instances without requiring retraining whenever a new attack variation is encountered. This type of detector is commonly referred to as a zero-day detector [23]. Latent variable models, such as autoencoders (AE), are particularly useful in this context. These models aim to learn the underlying explanatory factors (latent variables) from high-dimensional data samples by compressing them into low-dimensional representations. An AE consists of encoder and decoder networks: the encoder maps the input x to a compressed latent variable z, and the decoder attempts to reconstruct the input as $\tilde{x}$. The model is trained end-to-end by minimizing the reconstruction error $\epsilon = \|x - \tilde{x}\|_2$. After training, the model excels at reconstructing samples from the training class, but it struggles to accurately reconstruct samples from other classes.

A. The Variational AutoEncoder Model

There have been several updates to an AE model, out of which the Variational AE (VAE) is the most popular [24]. Instead of generating a latent vector z directly, in VAE, the encoder outputs mean (μ) and variance (σ) vectors constituting a latent probability distribution $q_\theta(z|x)$ from which z is sampled. As a result of this probabilistic setup, no two input samples have the same latent representation, which essentially forces the encoder to map similar input samples into a small region of the latent space. The latent variable z is input to the probabilistic decoder which reconstructs x using $p_\phi(x|z)$ distribution. The model is trained by maximizing the the variational lower bound [24] given by

$$\mathcal{L}(\theta, \phi; x) = -D_{KL}(q_\theta(z|x)||p(z)) + \mathbb{E}_{z\sim q_\theta(z|x)}(log\ p_\phi(x|z)). \tag{6}$$

Here the first terms is called the latent loss computed as the Kullback-Leibler divergence (KLD) between the learned distribution $q_\theta(z|x)$ and some prior distribution $p(z)$, which is typically set to be standard Normal $\mathcal{N}(\mathbf{0}, \mathbf{1})$. The second term is the reconstruction error term computed using the Binary cross entropy between reconstruction and input samples. Under this setting, the latent loss forces the latent distribution $q_\theta(z|x)$ to be symmetric around origin, ensuring a connected latent space. This is beneficial for our case as we want samples similar to the training class to lie as close to origin as possible, thereby, the distance from the origin can then be used as an indicator whether the testing sample is coming from the training set or not.

Our VAE encoder was constructed using a 3-layer neural network, with node configurations of $20 - 10 - 4$, and PReLU activation functions applied to all layers except the last one. The latent dimension was set to 2. The VAE decoder, on the other hand, utilized a 3-layer neural network with node configurations of $10 - 20 - n$, where n corresponds to the input dimensions. PReLU activation functions were employed for all inner nodes, while the final layer had no activation function. Additionally, we chose a latent dimension of 2 in order to facilitate easier visualization of the encoded representations.

B. Detection Framework

We will now present the proposed detection model, which utilizes the trained VAE encoder on authentic GPS signal samples. The underlying principle is that the VAE encoder maps authentic samples to the origin of the latent space, while spoofed samples are mapped away from the origin. Consequently, we can utilize the euclidean distance of the mean latent variable from the origin, represented as $\zeta(x) = \|\mu_z\|_2$, as an indicator function for classification into genuine or spoofed categories. In this context, $\|\cdot\|_2$ denotes the ℓ_2-vector norm. By defining a classification threshold τ, if $\zeta(x) \leq \tau$, the sample x is classified as genuine; otherwise, it is classified as spoofed.

The selection of an appropriate threshold, denoted as τ, plays a critical role in determining the overall detection performance. To ensure its relevance, we associate τ with the desired false positive rate (FPR), which is typically specified as part of a detector's technical requirements. Upon completion of model training, we apply the indicator function $\zeta(\cdot)$ to all the training samples and set τ to the threshold value that achieves the specified FPR for the training set. For instance, if the desired FPR is set at 1%, we consider samples as spoofed if their corresponding ζ values exceed the 99^{th} percentile for the entire training set. In the experimental section, we assessed the model's performance across different FPR settings, namely $[5\%, 1\%, 0.1\%]$, which represents a typical design choice for a detector.

By employing the representation learning-based detector, we aim to overcome the challenges associated with unseen attack patterns and improve the model's ability to differentiate between genuine and spoofed samples, as evidenced by the distinct and discriminative latent representations achieved through the VAE encoder, see Fig. 2.

IV. Experimental Evaluation

The implementations of all models discussed in this section were carried out in Python 3.9. The RFC and SVM models were implemented using the Scikit-Learn library [25], while the ANN and VAE models were trained using the PyTorch library in Python [26]. To optimize the loss functions of the ANN and VAE models, we utilized the Adam optimizer with learning rates of $1e^{-3}$ and $5e^{-4}$, respectively. The training process involved 50 epochs, with a batch size of 256. Evaluation of all methods was conducted based on overall accuracy as well as probability-based metrics, such as *probability of detection* (PD), *false alarm* (PFA), and *miss-detection* (PM).

A. Supervised Learning

In our initial test, we utilized the TEXBAT datasets and performed a 50-50 split to create separate train and test sets. The RFC, SVM, and ANN models were trained on the training set and subsequently evaluated on the testing set. Remarkably, all three models achieved accuracy scores exceeding 99.5%. To delve into the reasons behind such exceptional performance, we conducted a comparison between each individual time sample from the test set and all samples in the training set.

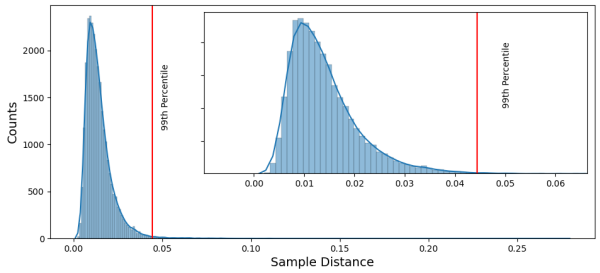

Fig. 1. Training and testing set sample difference distribution.

This allowed us to identify the sample from the training set that was closest to each test sample, and we recorded the ℓ_2 distance between them. The resulting histogram of these distances is depicted in Fig. 1. Notably, the histogram clearly illustrates that the samples in the test and training sets are extremely similar, indicating their equivalence. This finding provides an explanation for the high accuracy obtained by the RFC, SVM, and ANN models. Consequently, we assert that a simple train-test split should be approached with caution when working with high-frequency time series data, such as that of GPS spoofing datasets.

In our second test, we aimed to assess the generalization capability of the ML models when confronted with entirely unseen data. To achieve this, we employed a leave-one-out train-test strategy, where the models were trained on all but one dataset and evaluated on the remaining dataset. This approach ensured that the models were exposed to multiple GPS attack and clean scenarios during training, while being tested on an unseen attack dataset, thereby discouraging the mere memorization of the training datasets. The results of this test are presented in Table I. From the table, it can be observed that the authentic dataset proved to be the easiest to classify, as all three models achieved accuracy rates above 99%. Similarly, for the DS-2 dataset, which contained a spoofed signal with an approx. 10 dB power advantage, all methods exhibited a PD exceeding 99.7%. However, their PFA also exceeded 10%. Regarding the DS-3 dataset, featuring a 3 dB power advantage, both SVM and ANN achieved 100% PD and less than 1% PFA, while RFC only managed a PD of 79%.

Moving on to the DS-7 dataset, which involved power-matched and highly subtle GPS spoofing attacks, all three ML models completely failed in their detection performance. The SVM achieved the highest PD of 36%, underscoring the limitations of supervised ML approaches when confronted with unseen attack patterns at the receiver. This highlights a significant drawback of supervised ML models, namely their inherent inability to handle previously unseen attack patterns.

B. Representation Learning

In our final test, we employed our proposed representation learning-based detector, described in Section III, to address the limitations highlighted in the previous section and enhance its resilience to attack variations. We trained our VAE model following the methodology outlined in Section III-A,

TABLE I
PERFORMANCE UNDER LEAVE-ONE-OUT TRAINING STRATEGY.

Model	Test DS	ACC	PD	PFA	PM
RFC	Authentic	99.70	-	0.30	-
	DS-2	96.58	99.72	13.43	0.28
	DS-3	84.60	79.15	1.29	20.85
	DS-7	4.28	0.11	0.00	99.89
SVM	Authentic	99.12	-	0.88	-
	DS-2	97.39	99.95	10.78	0.05
	DS-3	99.88	100.00	0.43	0.00
	DS-7	38.86	36.19	0.00	63.81
ANN	Authentic	99.21	-	0.79	-
	DS-2	96.22	99.96	15.71	0.04
	DS-3	99.82	100.00	0.63	0.00
	DS-7	31.70	28.73	0.00	71.27

utilizing the authentic dataset for training. Subsequently, the trained encoder and the authentic dataset samples were utilized to compute the classification threshold τ for different false positive rates (FPRs) of $[5\%, 1\%, 0.1\%]$. Using the trained encoder, we mapped data samples from every dataset into their respective 2D latent representations. The visual representations of these mappings are presented in Fig. 2. Notably, the samples from the authentic dataset were mapped close to the origin of the latent space, signifying VAE's effective training. Conversely, the spoofed samples from the DS-2 dataset were mapped considerably farther away from the origin. Similarly, a significant portion of the DS-3 samples were also mapped away from the origin, with only a few samples remaining close to it. Intriguingly, several samples from the sophisticated attack dataset DS-7 were mapped very close to the origin, indicating their resemblance to the authentic samples. This phenomenon explains why the supervised learning-based methods misclassified them, as demonstrated in Table I.

The computed threshold τ, which can be interpreted as the radius of a circle centered at the origin, is utilized to classify whether a data point is spoofed or genuine. The classification results for different input FPRs are presented in Table II. Notably, even with a lenient 5% FPR, the detection rates for all datasets surpass 99.63%, albeit with a higher PFA exceeding 10%. However, when employing a stricter FPR of 0.1%, the detection rates for DS-2 and DS-3 datasets remain close to 100%, while the detection rate for DS-7 drops to 91.78%. Additionally, the PFA decreases significantly for all datasets.

As illustrated in Fig. 2, a few data points from the DS-7 dataset are located in close proximity to the origin of the latent space. Consequently, for a sufficiently low FPR, these points fall below the classification threshold, resulting in a reduction in both the detection rate (PD) and the PFA. Thus, the selection of the threshold can be determined by striking a balance between the PD and PFA for the subtle attacking scenario presented in DS-7.

V. CONCLUSION

In this article, we introduced a novel framework for detecting time synchronization attacks on synchrophasors using six

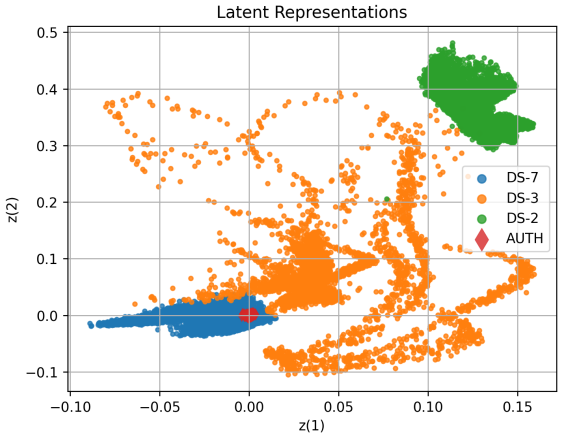

Fig. 2. Latent variables extracted by the VAE from TEXBAT datasets.

TABLE II
PROPOSED DETECTOR'S GPS SPOOFING DETECTION SCORES UNDER MULTIPLE FALSE POSITIVE RATES.

FPR/PFA	5%			1%			0.10%		
Test DS	ACC	PD	PFA	ACC	PD	PFA	ACC	PD	PFA
Authentic	94.32	-	5.68	98.93	-	1.07	99.90	-	0.10
DS-2	94.37	100.00	23.60	96.84	99.99	13.19	99.61	99.96	1.51
DS-3	96.96	100.00	10.92	98.95	100.00	3.77	99.75	100.00	0.89
DS-7	99.22	99.63	10.29	98.36	98.39	2.47	91.78	91.46	0.82

easily computable features from a generic GPS receiver. The PMUs share these features, along with their measurements, with the control center for training/testing models to detect GPS spoofers. Our study revealed the limitations of supervised ML models in handling high sampling rate time series data, as they tend to over-fit and require data from all possible attack scenarios for effective training. In contrast, our representation learning-based detector, employing a variational autoencoder trained solely on authentic datasets, outperformed supervised ML models, especially for the DS-7 attack scenario. A major advantage of our detector is its robustness in detecting spoofing without exhaustive coverage of all possible attack scenarios during training. This enables the control center to employ more advanced mitigation methods to counter spoofing effects before making smart grid control decisions.

REFERENCES

[1] M. N. Aman, K. Javed, B. Sikdar, and K. C. Chua, "Detecting data tampering attacks in synchrophasor networks using time hopping," in *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pp. 1–6, IEEE, 2016.

[2] G. Directorate, "NAVSTAR GPS space segment/user segment L1C interfaces," tech. rep., 2019.

[3] 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.

[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] R. Huang and Y. Li, "False phasor data detection under time synchronization attacks: A neural network approach," *IEEE Transactions on Smart Grid*, vol. 13, no. 6, pp. 4828–4836, 2022.

[6] M. Sabouri, S. Siamak, M. Dehghani, M. Mohammadi, and M. H. Asemani, "Intelligent GPS spoofing attack detection in power grid," in *2021 11th Smart Grid Conference (SGC)*, pp. 1–6, IEEE, 2021.

[7] F. Zhu, A. Youssef, and W. Hamouda, "Detection techniques for data-level spoofing in GPS-based phasor measurement units," in *2016 International Conference on Selected Topics in Mobile Wireless Networking (MoWNeT)*, pp. 1–8, IEEE, 2016.

[8] Y. Fan, Z. Zhang, M. Trinkle, A. D. Dimitrovski, J. B. Song, and H. Li, "A cross-layer defense mechanism against GPS spoofing attacks on PMUs in smart grids," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2659–2668, 2014.

[9] J. Xie and A. S. Meliopoulos, "Sensitive detection of GPS spoofing attack in phasor measurement units via quasi-dynamic state estimation," *Computer*, vol. 53, no. 5, pp. 63–72, 2020.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[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] G. Aissou, H. O. Slimane, S. Benouadah, and N. Kaabouch, "Tree-based supervised machine learning models for detecting gps spoofing attacks on uas," in *2021 IEEE 12th Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, pp. 0649–0653.

[17] X. Shang, F. Sun, B. Liu, L. Zhang, and X. Zhu, "Mitigation of GNSS time synchronization attacks in a multicorrelator receiver," *IEEE Access*, vol. 10, pp. 70383–70393, 2022.

[18] 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.

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

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

[21] 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.

[22] S. Theodoridis, *Machine learning: a Bayesian and optimization perspective*. Academic press, 2015.

[23] L. Yang, A. Finamore, F. Jun, and D. Rossi, "Deep learning and zero-day traffic classification: Lessons learned from a commercial-grade dataset," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4103–4118, 2021.

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

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.