Cyber Attack Detection in Distribution Networks with Topological Data Analytics aided Learning

Damilola R. Olojede*,1, Md Joshem Uddin†,1, Roshni Anna Jacob‡, Baris Coskunuzer†,2 and Jie Zhang*,‡,2,
*Department of Mechanical Engineering, The University of Texas at Dallas
†Department of Mathematical Sciences, The University of Texas at Dallas
‡Department of Electrical and Computer Engineering, The University of Texas at Dallas

2024 IEEE Kansas Power and Energy Conference (KPEC) | 979-8-3503-7240-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/KPEC61529.2024.10676194

***Abstract*—The integration of smart grid technologies has brought significant advancements to power systems, yet it has also increased its vulnerability to cyber threats. False data injection attacks (FDIAs) pose a substantial risk to grid data integrity, particularly in critical areas like voltage control and state estimation. This study centers on leveraging the latest advancement in topological data analysis (TDA), specifically multi-parameter persistent homology, which has shown remarkable effectiveness in graph representation learning in recent years. Our objective is to utilize this approach to bolster the detection of FDIA using data collected from voltage sensors. By integrating topological methods, our approach aims to fortify the resilience of power systems against cyber threats, thereby ensuring the reliability and security of smart grid operations.**

***Index Terms*—Distribution networks, cyber attacks, anomaly detection, topological data analysis, multiparameter persistence**

I. Introduction

The emergence of smart grid technologies has brought about a significant transformation in the power grid, providing avenues to enhance the efficiency and reliability in power supply. This evolution has been facilitated by the increased adoption of distributed energy resources, remote-controlled devices, and advanced monitoring and communication infrastructure within the power network. However, despite these benefits, the integration of smart grid technologies has also increased the vulnerability of the power network to cyber-physical attacks and threats.

The key security requirements for a well-functioning grid as per the National Institute of Standards and Technology (NIST) interoperability panel include data availability, integrity and confidentiality [1]. Transmitting the sensitive power data over public or private networks creates opportunities for the attackers to access this information, thereby exposing sensitive grid data. Specific areas of the power system often targeted during cyber attacks include automatic generation control, state estimation, load redistribution, voltage control, etc. The attacks can range from denial of service (DoS) attacks, which aim to delay or block the availability of critical operational data (through methods such as channel jamming or spoofing), to data integrity attacks which involve malicious modification of transmitted data to compromise the stability of the grid.

In this paper, we focus on false data injection attacks (FDIA), a type of data integrity attack known to circumvent traditional algorithms for bad measurement detection [2]. The attacker exploits the vulnerablilities in the communication system to inject false data into the voltage sensors within the power distribution system, thus providing incorrect system state information at the control centers. This could lead to unwarranted voltage control actions being implemented in the network, potentially resulting in either under or over voltage conditions.

In recent years, there has been considerable research on FDIA detection algorithms. The FDIA detection methods primarily fall into two categories: traditional state estimation-based and machine learning-based approaches [3]. The state estimation methods employed for FDIA detection utilize variations of the conventional least-squared methods [4]. In [5], the authors proposed a distributed state estimation method for detecting bad data, where the power network was divided into subsystems to perform state estimation. In such methods, the residual error exceeding the predefined thresholds indicate anomalies or bad data. However, manually setting the thresholds can result in false FDIA alarms. It is imperative to strike a balance between sensitivity to attacks and reliability in system operation. Hence, adopting machine learning or data driven methods may be suitable to minimize errors and false alarms.

The other class of methods for FDIA detection, employing machine learning techniques, range from unsupervised learning to different variations of deep learning [6]–[8]. However, these data-driven approaches have not considered the interdependence among node variables (such as voltages, power demand/generation, etc.) resulting from the underlying network connectivity. For instance, a surge in load demand at a specific bus (node) within the network may lead to voltage drops, affecting neighboring network buses. However, an attack on the voltage sensors would not yield the same behavior on network variables, since the cause of undervoltage is external and not intrinsic to the network. To effectively identify these patterns, learning techniques must integrate network topology into the detection model. This is evident in [9] where the use of Gated Graph Neural Networks significantly improved FDIA detection accuracy compared to alternative Euclidean data-driven methods.

We aim to advance this by employing topological data analytics-aided learning models that will capture the temporal evolution of the topological signature and extract meaningful

[1]Co-First Authors; [2]Co-Senior Authors.

latent features for learning. The contributions of our paper are as follows:

- We generate a distribution network cyber attack dataset that is composed of FDIA scenarios for voltage magnitudes.
- We present a novel model to detect anomalies stemming from FDIA on voltage sensors.
- Our approach combines the latest tools in TDA, namely multipersistence, and various machine learning methods to achieve enhanced detection promptly.

II. Methodology

A. Problem Formulation

In this paper, our focus is on FDIAs targeting voltage measurements within distribution networks. These attacks manipulate voltage magnitudes, deceiving control centers by falsely indicating under- or over-load conditions at the buses. Table I is formulated from [10] and it presents the different FDIAs considered in this paper. The voltage magnitude, denoted by V, is represented in per unit (p.u.).

TABLE I
Anomalies in Voltage Magnitude

Anomaly Type	Expression
Interruption	$V < 0.1$
Undervoltage	$0.1 \leq V \leq 0.9$
Overvoltage	$1.1 \leq V \leq 1.8$

B. Dataset Generation

The test networks used in this paper are the IEEE 37- and 123-bus networks. We started the data generation process by first extracting the network's graph structure, with the buses represented by the graph nodes, and the branches (i.e., lines/transformers) represented by the edges. A scenario generation method is employed to compose the dataset, which is summarized in Fig. 1. Varying load shapes are simulated by the addition of a random noise signal to generate different loading conditions for each scenario. The dataset is composed of scenarios representing both normal operation and anomalous conditions. The scenarios are randomly selected to represent anomalies by inducing attacks on voltage measurements. In each attack scenario, buses are randomly chosen within the network, then FDIAs are added to the voltage signals acquired from the network model simulation.

The scaling attacks are constructed following the randomized attack template discussed in [11]. Building on this, we obtained:

$$V_t = \begin{cases} V_t & \text{for t} \notin \Gamma_a \\ (1+S) \times V_t & \text{for t} \in \Gamma_a \end{cases} \quad (1)$$

where V_t represents the voltage magnitude in per unit at time t, S is the scaling factor used to inject the anomaly, and Γ_a includes the time steps that fall within the duration when anomaly occurs. The start time and length of Γ_a are drawn from a uniform random distribution. Also, the scaling factor S is selected from a uniform distribution ranging from -1.0 to 0.5. The range for the scaling factor is appropriately selected to encompass the three different anomalies stated in Section II-A.

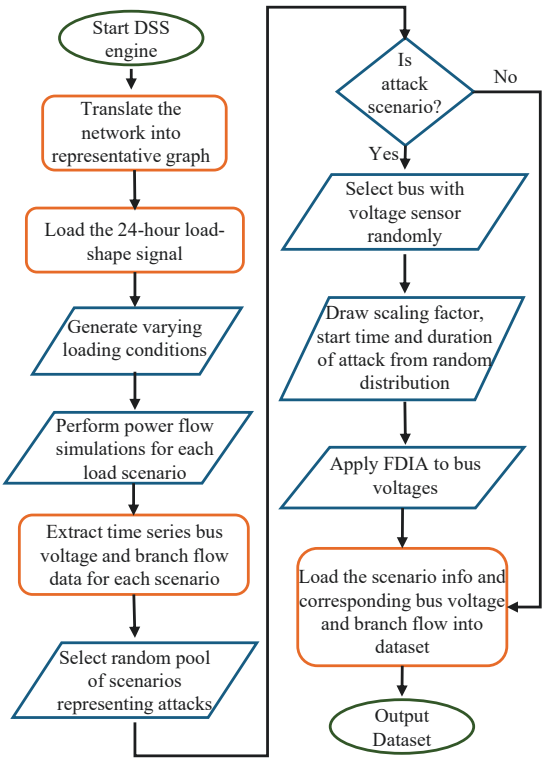

Fig. 1. Flow chart representing the scenario generation process used to construct the dataset. The dataset generated considers the variations in load patterns, attack site, type and the duration of attacks. Both normal operation and anomalous conditions are present in the dataset.

C. Persistent Homology

Persistent Homology (PH) acts as a mathematical framework for extracting hidden topological patterns within data, utilizing techniques from algebraic topology [12].

The main idea behind PH is to extract a meaningful sequence of topological spaces, and record the evolution of topological features on this sequence. In particular, for a given graph $\mathcal{G}$, we construct a nested sequence of subgraphs $\mathcal{G}^1 \subseteq \ldots \subseteq \mathcal{G}^N = \mathcal{G}$. For each $\mathcal{G}^i$, we define an abstract simplicial complex $\widehat{\mathcal{G}}^i, 1 \leq i \leq N$, resulting in a nested sequence of simplicial complexes $\widehat{\mathcal{G}}^1 \subseteq \ldots \subseteq \widehat{\mathcal{G}}^N$, which is known as *filtration*. One of the most common choices for simplicial complex is the clique complex. This filtration step is a crucial aspect of PH as it allows for the incorporation of domain-specific information into the PH process.

In the context of an unweighted graph $\mathcal{G} = (\mathcal{V}, E)$, a common approach involves the use of a filtering function $f : \mathcal{V} \to \mathbf{R}$ alongside a set of thresholds $\mathcal{I} = \{\alpha_i\}$ where $\alpha_1 = \min_{v \in \mathcal{V}} f(v) < \alpha_2 < \ldots < \alpha_N = \max_{v \in \mathcal{V}} f(v)$. For each $\alpha_i \in \mathcal{I}$, let $\mathcal{V}_i = \{v_r \in \mathcal{V} \mid f(v_r) \leq \alpha_i\}$. Define $\mathcal{G}^i$ as the induced subgraph of $\mathcal{G}$ by $\mathcal{V}_i$, i.e. $\mathcal{G}^i = (\mathcal{V}_i, \mathcal{E}_i)$ where $\mathcal{E}_i = \{e_{rs} \in \mathcal{E} \mid v_r, v_s \in \mathcal{V}_i\}$. This procedure results in a nested sequence of subgraphs $\mathcal{G}^1 \subset \mathcal{G}^2 \subset \ldots \subset \mathcal{G}^N = \mathcal{G}$,

called *the sublevel filtration* induced by the filtering function f, as shown in Fig. 2. The selection of the function f is pivotal in this context, and often, f is derived from a significant characteristic within the domain. In power networks, typically the voltages measured at the different buses and the powerflow through the branches represent the system state. Therefore, these are considered as the filtering functions in the context of power grids as they aid in assessing the health of the network.

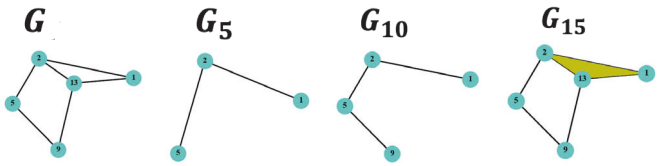

Fig. 2. **Single Persistence.** A simplified illustration of sublevel filtration, where node attributes determine threshold values of 5, 10, and 15. In this context, G_5 denotes the sub-simplicial complex of the graph G, consisting of nodes with values less than 5 and the corresponding edges between them. Similarly, G_{10} and G_{15} represent similar complexes based on nodes below their respective threshold values.

During this construction of sub-simplicial complexes, various topological features may arise and disappear over time. A k-dimensional topological feature, also known as a k-hole (σ), can represent different structures such as connected components (0-hole), loops (1-hole), or cavities (2-hole). Persistent homology systematically tracks the evolution of these topological patterns. When a topological feature emerges initially in $\widehat{\mathcal{G}}^{b_\sigma}$ and vanishes in $\widehat{\mathcal{G}}^{d_\sigma}$, we attribute a persistence value of $b_\sigma - d_\sigma$ to this feature. Alternatively, we can express this feature as a tuple (b_σ, d_σ), which we compile in a persistence diagram (PD), as seen in Fig. 3. Then the k^{th} *persistence diagram* is defined as

$$\mathrm{PD_k}(\mathcal{G}) = \{(b_\sigma, d_\sigma) \mid \sigma \in H_k(\widehat{\mathcal{G}}^i) \text{ for } b_\sigma \leq i < d_\sigma\}$$

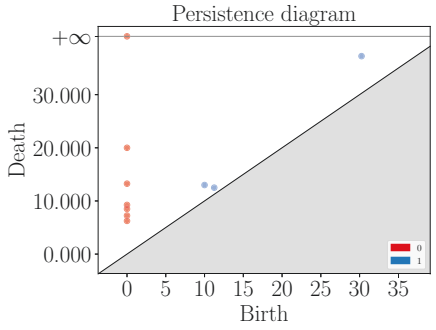

Fig. 3. An example of a persistence diagram (PD): in this diagram, red dots represent the tuple (birth, death) for connected components (0 holes), while blue dots represent the tuple (birth, death) for loops (1 holes).

Here, $H_k(\widehat{\mathcal{G}}^i)$ is the k^{th} homology group of $\widehat{\mathcal{G}}^i$ which keeps the information of the k-holes in the simplicial complex $\widehat{\mathcal{G}}^i$. We can also keep track of the evaluation of topological tensor by persistent barcode. Each bar in a persistent barcode represents a topological feature (e.g., a connected component, loop, or void) and its persistence - how long it lasts as the parameter changes.

The final step of PH is the vectorization process. While PH uncovers hidden shape patterns from data in the form of PDs, which consist of collections of points (birth times and death times) in $\mathbf{R}^2$, these diagrams are not inherently suitable for statistical and machine learning purposes. Instead, common techniques involve faithfully representing these PDs as kernels [13] or vectorizations [14]. Among the common single persistence (SP) vectorization methods are Persistence Images, Persistence Landscapes, Silhouettes, and various Persistence Curves (e.g., Betti number) [14]. The betti number is focused in this study, which describes the number of 'holes' of various dimensions in the space, as seen in Fig. 4. These vectorization methods typically transform PDs into single-variable functions or fixed-size vectors for use in various applications.

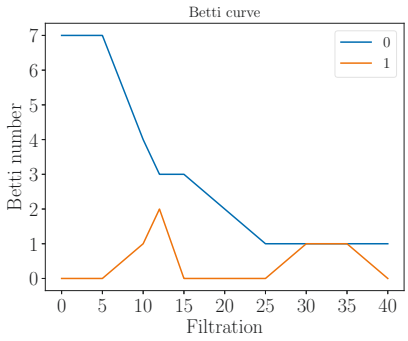

Fig. 4. Betti numbers represent the number of topological holes (connected components, loops, etc.) at each threshold value.

In this work, our objective is to expand the scope of SP vectorizations by extending them into their multidimensional counterparts. This expansion allows us to capture changes in the temporal dimension by treating power grids as dynamic networks and thereby gain deeper insights through the utilization of multipersistence approaches.

D. Multiparameter Persistence for Power Grids

Up to this point, our discussion has been focused solely on single-parameter persistence theory. The term "single" arises because we filter the data using only one function or parameter. The construction of the filtration method is crucial for detailed data analysis and capturing concealed patterns. However, in many applications, there are multiple natural domain functions available for analyzing the data. Utilizing these functions simultaneously would provide a more comprehensive understanding of the hidden patterns. With this insight in mind, multiparameter persistence (MP) theory emerges as a natural extension of single persistence (SP). So, if we utilize two or more functions, we gain the ability to examine the data in much finer detail. For instance, if we have two functions $f : \mathcal{V} \to \mathbf{R}$ and $g : \mathcal{V} \to \mathbf{R}$ with complementary information about the network, MP allows us to combine the insights from both functions into a unique topological fingerprint. These functions f and g induce a multivariate filtering function $F : \mathcal{V} \mapsto \mathbf{R}^2$ defined as $F(v) = (f(v), g(v))$. We then define non-decreasing thresholds $\{\alpha_i\}_1^m$ and $\{\beta_j\}_1^n$

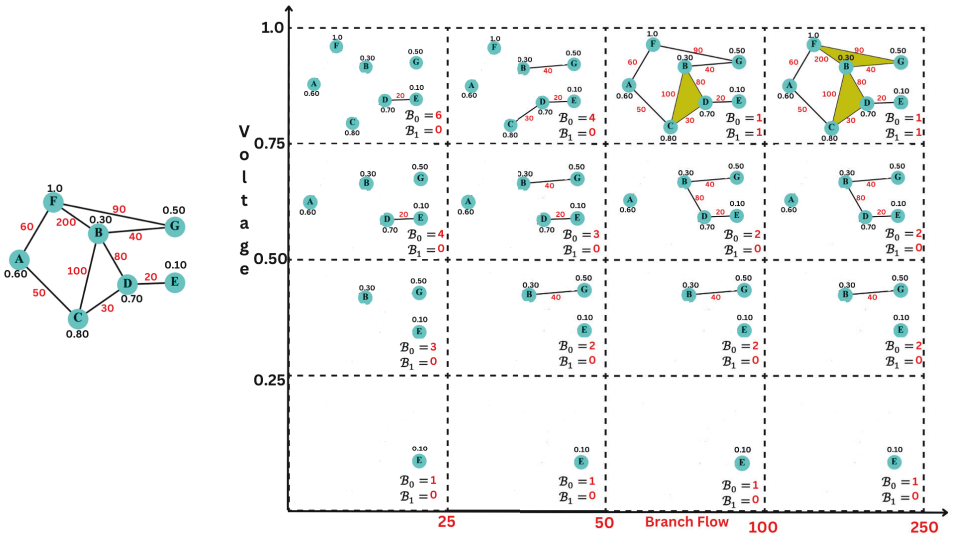

Fig. 5. **Multiparameter Persistence**. For a given power network, employing MultiPersistence allows us to extract substructures (subgraphs) dictated by both bus voltages and branch flows. The signature (e.g., β_0 and β_1) on each substructure induces an $m \times n$ tensor, which effectively captures the topological changes in the bifiltration, and helps to detect the anomalies in the network.

for f and g, respectively. Using these thresholds, we define sets $\mathcal{V}^{ij} = \{v_r \in V \mid f(v_r) \leq \alpha_i, g(v_r) \leq \beta_j\}$. Each $\mathcal{G}^{ij}$ represents the induced subgraph of $\mathcal{G}$ by $\mathcal{V}^{ij}$, capturing the hidden patterns in the data revealed by multipersistence (See Fig. 5). Note that the top row (or rightmost column) in the multipersistence grid represents single persistence with respect to the corresponding parameter. By constructing simplicial complexes from these subgraphs, we obtain a bifiltration of complexes $\{\widehat{\mathcal{G}}^{ij} \mid 1 \leq i \leq m, 1 \leq j \leq n\}$. Next, by computing the homology groups of these complexes, $\{H_k(\mathcal{G}^{ij})\}$, we obtain the induced bigraded persistence module, representing a rectangular grid of size $m \times n$. For more details on multipersistence, see [15].

The core principle guiding the multipersistence method is to extract vital descriptors from the meaningful substructures of the graph generated by employing multiple functions simultaneously. In simpler terms, functions f and g facilitate the organized decomposition of the entire graph into subgraphs, where the topological changes in specific subgraphs provide key signatures relevant to the downstream task. Over the recent years, multipersistence has demonstrated considerable efficacy in graph representation learning, surpassing numerous conventional methods and graph neural networks (GNNs) in various tasks [16]–[19].

Power grids, represented as the weighted (branch flow) directed network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, offer a diverse array of filtration functions for effectively applying TDA models. These functions can be divided into two main categories. The first category encompasses general functions applicable to any graph, such as degree, betweenness, and closeness. These functions capture fundamental graph properties and prove particularly valuable for tasks related to graph classification. The second category consists of *domain functions*, directly derived from the dataset's domain, in this case, power grids. These functions, e.g., bus voltages, residual capacity, and branch flow, offer insights tailored to the unique characteristics and behavior of power grids.

In this study, we have employed two types of filtration functions. The first type is filtration based on node features, where we utilize bus voltage as the node filtration parameter. Each observable node encompasses three voltage measurements for the three phases, and thus, we considered their average. Another filtration function we used is based on the edge weight, (w_{ij}) as the filtration parameter. In this approach, for sublevel filtration $\mathcal{G}_n = (\mathcal{V}_n, \mathcal{E}_n)$ is the subgraph generated by the edge set $\mathcal{E}_n = \{e_{ij} \in \mathcal{E} \mid \omega_{ij} \leq \alpha_n\}$, where $\mathcal{G}_n$ is the smallest subgraph in $\mathcal{G}$ containing the edges in $\mathcal{E}_n$. Consequently, $\mathcal{V}_n$ automatically comprises the set of endpoints of the edges in $\mathcal{E}_n$. Using these two filtration approaches, we constructed an $m \times n$ nested sequence of subgraphs, as mentioned earlier. Subsequently, we derived $m \times n$ topological features from the power grid network.

III. Simulation and Result Analysis

A. Distribution Network Simulation

We use the open source distribution system simulator (OpenDSS) [20] to simulate the power distribution network state. A time-series power flow is run using the daily simulation mode with a 1-hour resolution load-shape in OpenDSS. A representative sample from the dataset demonstrating an attack on the bus '799' in the 37-bus network is shown in Figs. 7 and 8. Specifically, Fig. 7 represents the time series voltage signals for the three phases at bus '799' during normal operation for a particular load shape. Natural voltage drops below the desired limit of 0.9 per unit is observed on phases 2 and 3 of bus '799'. This is attributed to the loading condition on the 37-bus network. Fig. 8, on the other hand, represents the three phase voltage signals at bus '799' for the same loading conditions with FDIA. In this scenario, the scaling factor is -0.407 (Eq. 1),

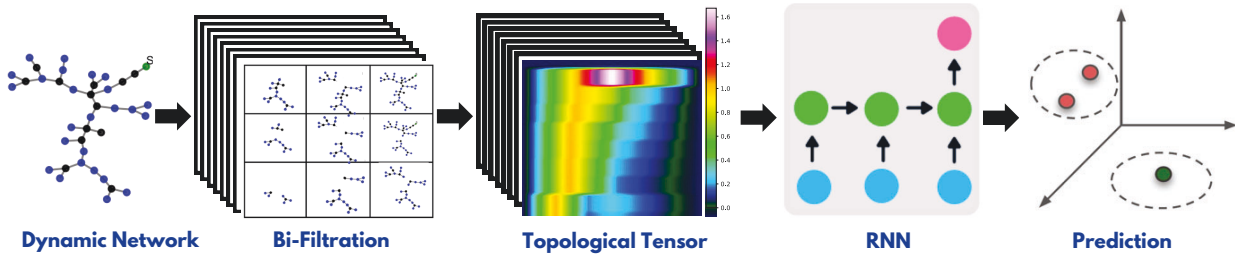

Fig. 6. Our model pipeline involves the following steps: Within a dynamic network, we generate an $m \times n$ multi-persistence tensor at each time step. Next, we extract a topological tensor from this substructure and input it into the recurrent neural network for classification purposes.

thereby introducing an under-voltage attack for 5 hours from the 7th to the 12th hour of operation.

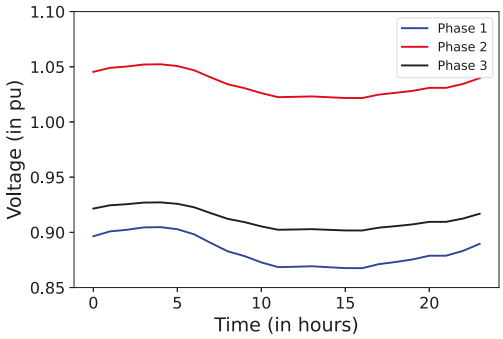

Fig. 7. Voltage plot for bus '799' in the 37-bus network during normal operation for specific loading conditions. Regular non-anomalous under-voltage values (< 0.90 p.u.) that could occur with normal system operations are observed.

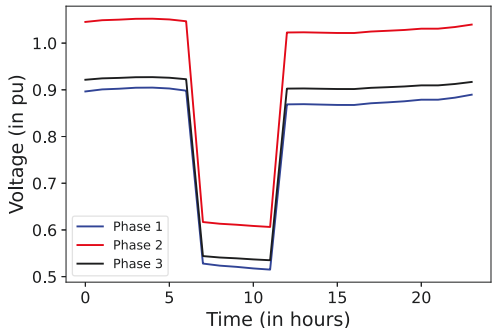

Fig. 8. Voltage plot for bus '799' in the 37-bus network for the same loading conditions as in Fig. 7 with FDIA. An under-voltage attack has been injected for 5 hours starting from the 7th hour of the operation.

B. Experimental Setup

We conducted our experiments on a machine equipped with the Apple M2 chip, which includes an 8-core CPU, a 10-core GPU, and a 6-core Neural Engine, along with 16GB of RAM. Our code is implemented in Python version 3.11.4. We present the outcomes derived from single persistence, employing bus voltage data, alongside multi-persistence, integrating node voltage and branch flow information. For single persistence, we employed 18 threshold values for voltage to construct sub-graph complexes. Additionally, for multi-persistence analysis, we incorporated 8 dimension thresholds for branch flow. For computing the number of k^{th} homology groups for each sub-simplicial complex, we utilized the Pyflagser package.

Given the network's nature, we concentrated solely on the 0^{th} homology group, denoted as β_0, which signifies the number of connected components in the sub-simplicial complexes. Each time series is labeled as either 1 (indicating an anomaly at some time step) or 0 (indicating no anomaly). Therefore, we treat the anomaly detection as a binary time series classification task. We use the extracted topological features as input for machine learning classifiers, specifically XGBoost, RandomForest (RF), Multi-Layer Perceptron (MLP), and Long Short-Term Memory (LSTM) based Recurrent Neural Networks (RNNs). For each sample, we possess 24 sets of data spanning 24 time steps, resulting in 24 18-dimensional topological tensor for single persistence and 24 18×8 dimensional tensor in multi-persistence. In our machine learning model, we perform element-wise addition of all these 24 vectors to serve as input. In the case of the LSTM model, the vector corresponding to each time step is utilized as input.

Both XGBoost and RF classifier are trained with 100 boosting strategies, a learning rate set to 0.1, and a maximum tree depth of 10 to avoid overfitting. For the MLP, we optimize it using the 'adam' optimizer with a sigmoid activation function and set a maximum of 1000 iterations. We optimize the LSTM model using the 'adam' optimizer, employing a tanh activation function for the main layer and a sigmoid activation function for the recurrent connections. The pipeline of our model is shown in Fig. 6.

To evaluate performance and address overfitting concerns, we report the average score and standard deviation from a 10-fold cross-validation. In the 10-fold cross-validation, the dataset is divided into 10 equal parts or folds. The model is trained and evaluated 10 times, each time using a different fold as the validation set and the remaining 9 folds as the training set. This allows us to assess the effectiveness and versatility of these topological features with different machine learning models. Our code is accessible here[1].

C. Results

The results presented in Table II demonstrate that our TDA augmented learning model achieves good performance on both the IEEE 37-bus and IEEE 123-bus datasets. A major

[1]https://anonymous.4open.science/r/Cyber_Attack-1735/README.md

TABLE II
PERFORMANCE EVALUATION: CLASSIFICATION ACCURACY (%) AND STANDARD DEVIATION OF DIFFERENT METHODS

Dataset	ML Model	Single Persistence	MultiPersistence
IEEE 37-bus	XGBoost	84.00±6.67	**86.67±6.08**
	MLP	83.20±4.73	**86.67±3.51**
	RandomForest	84.10±4.18	**86.00±4.66**
	LSTM	79.39±8.11	**88.66±5.02**
IEEE 123-bus	XGBoost	84.80±5.18	**91.00±5.68**
	MLP	83.00±4.02	**92.00±4.76**
	RandomForest	84.40±3.62	**89.66±4.57**
	LSTM	80.39±4.97	**91.66±5.40**

challenge here is to consider the time evolution of the voltage signal and convert the anomaly indicator attributed to a time series signal to a label at each time step. By extracting the topological tensor of the network at each time step, we can create a suitable input vector for each time series for a machine learning model. This approach proves the effectiveness of topological features in the context of power grid analysis. In our model, we employed both the traditional method in TDA, single persistence, and the latest and improved version of it, multipersistence, to extract the topological signatures for cyber attack detection in power grids. We observe that multi-persistence approach is able to extract finer topological features from the network, leading to significantly superior performance compared to using single persistence.

In particular, our model employing multi-persistence demonstrates substantial improvements, achieving an average gain of 4% for the IEEE 37-bus network, and 8% for the IEEE 123-bus network, using various machine learning classifiers. This highlights the effectiveness of multi-persistence in the context of power distribution threat analytics.

IV. CONCLUSION AND FUTURE WORK

In this paper, we introduced a novel approach for detecting anomalies caused by cyber attacks targeting voltage sensors within power distribution networks. We focused on false data injection attacks, where attackers manipulate voltage measurements to deceive operators by fabricating under or overvoltage conditions at network buses. A learning-based approach has been developed for anomaly detection incorporating the latest topological data analysis methods to extract evolving topological signatures over time. The proposed model has been validated on the IEEE 37-bus and 123-bus networks, where single and multi-parameter persistence using voltage and branch-flow data yields significant results.

The future scope of this work involves leveraging the developed framework, which integrates persistent homology and deep learning frameworks, to detect increasingly sophisticated attacks, including those targeting distributed energy resources and voltage regulators.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation under grants DMS-2202584, DMS-2220613, and DMS-2229417, Simons Foundation under grant # 579977, and Office of Naval Research under ONR award number N00014-21-1-2530.

REFERENCES

[1] K. Chatterjee, V. Padmini, and S. A. Khaparde, "Review of cyber attacks on power system operations," in *2017 IEEE Region 10 Symposium (TENSYMP)*, 2017, pp. 1–6.

[2] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, pp. 1–33, 2011.

[3] J. Cao, D. Wang, Z. Qu, M. Cui, P. Xu, K. Xue, and K. Hu, "A novel false data injection attack detection model of the cyber-physical power system," *IEEE Access*, vol. 8, pp. 95 109–95 125, 2020.

[4] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *Preprints of the first workshop on secure control systems, CPSWEEK*, vol. 2010. Stockholm, Sweden, 2010.

[5] Y. Gu, T. Liu, D. Wang, X. Guan, and Z. Xu, "Bad data detection method for smart grids based on distributed state estimation," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 4483–4487.

[6] M. Mohammadpourfard, A. Sami, and A. R. Seifi, "A statistical unsupervised method against false data injection attacks: A visualization-based approach," *Expert Systems with Applications*, vol. 84, pp. 242–261, 2017.

[7] D. Xue, X. Jing, and H. Liu, "Detection of false data injection attacks in smart grid utilizing elm-based ocon framework," *IEEE Access*, vol. 7, pp. 31 762–31 773, 2019.

[8] J. James, Y. Hou, and V. O. Li, "Online false data injection attack detection with wavelet transform and deep neural networks," *IEEE Transactions on Industrial Informatics*, pp. 3271–3280, 2018.

[9] X. Li, Y. Wang, and Z. Lu, "Graph-based detection for false data injection attacks in power grid," *Energy*, vol. 263, p. 125865, 2023.

[10] M. Jamei, A. Scaglione, C. Roberts, E. Stewart, S. Peisert, C. McParland, and A. McEachern, "Anomaly detection using optimally placed μPMU sensors in distribution grids," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3611–3623, 2017.

[11] M. Sun, L. He, and J. Zhang, "Deep learning-based probabilistic anomaly detection for solar forecasting under cyberattacks," *Int. Journal of Electrical Power & Energy Systems*, p. 107752, 2022.

[12] T. K. Dey and Y. Wang, *Computational Topology for Data Analysis*. Cambridge University Press, 2022.

[13] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels," *Applied Network Science*, vol. 5, no. 1, pp. 1–42, 2020.

[14] D. Ali, A. Asaad, M.-J. Jimenez, V. Nanda, E. Paluzo-Hidalgo, and M. Soriano-Trigueros, "A survey of vectorization methods in topological data analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[15] M. B. Botnan and M. Lesnick, "An introduction to multiparameter persistence," *arXiv preprint arXiv:2203.14289*, 2022.

[16] O. Vipond, J. A. Bull, P. S. Macklin, U. Tillmann, C. W. Pugh, H. M. Byrne, and H. A. Harrington, "Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors," *Proceedings of the National Academy of Sciences*, vol. 118, no. 41, p. e2102166118, 2021.

[17] M. Carriere and A. Blumberg, "Multiparameter persistence image for topological machine learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 432–22 444, 2020.

[18] D. Loiseaux, L. Scoccola, M. Carrière, M. B. Botnan, and S. Oudot, "Stable vectorization of multiparameter persistent homology using signed barcodes as measures," *NeurIPS*, 2023.

[19] C. Xin, S. Mukherjee, S. N. Samaga, and T. K. Dey, "GRIL: A 2-parameter persistence based vectorization for machine learning," in *Topological, Algebraic and Geometric Learning Workshops 2023*. PMLR, 2023, pp. 313–333.

[20] D. Montenegro, M. Hernandez, and G. Ramos, "Real time opendss framework for distribution systems simulation and analysis," in *2012 Sixth IEEE/PES Transmission and Distribution: Latin America Conference and Exposition (T&D-LA)*. IEEE, 2012, pp. 1–5.