IEEE Open Access Journal of Power and Energy

Received 13 August 2024; revised 22 November 2024; accepted 10 January 2025. Date of publication 16 January 2025; date of current version 24 January 2025.

Digital Object Identifier 10.1109/OAJPE.2025.3530352

Graphon Neural Networks-Based Detection of False Data Injection Attacks in Dynamic Spatio-Temporal Power Systems

RACHAD ATAT[1] (Senior Member, IEEE), ABDULRAHMAN TAKIDDIN[2] (Member, IEEE), MUHAMMAD ISMAIL[3,4] (Senior Member, IEEE), AND ERCHIN SERPEDIN[5] (Fellow, IEEE)

[1]Department of Computer Science and Mathematics, Lebanese American University, Beirut 1102-2801, Lebanon
[2]Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering, Florida State University, Tallahassee, FL 32310 USA
[3]Cybersecurity Education, Research and Outreach Center (CEROC), Tennessee Technological University, Cookeville, TN 38501 USA
[4]Department of Computer Science, Tennessee Technological University, Cookeville, TN 38501 USA
[5]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA
CORRESPONDING AUTHOR: R. ATAT (rachad.atat@lau.edu.lb)

This work was supported by NSF Energy, Power, Control, and Networks (EPCN) under Award 2220346 and Award 2220347.

ABSTRACT Cyberattacks on power systems have doubled due to digitization, impacting healthcare, social, and economic sectors. False data injection attacks (FDIAs) are a significant threat, allowing attackers to manipulate power measurements and transfer malicious data to control centers. In this paper, we propose the use of graphon neural networks (WNNs) for detecting various FDIAs. Unlike existing graph neural network (GNN)-based detectors, WNNs are efficient as they make use of the non-parametric graph processing method known as graphon, which is a limiting object of a sequence of dense graphs, whose family members share similar characteristics. This allows to leverage the learning by transference on the graphs to address the computational complexity and environmental concerns of training on large-scale systems, and the dynamicity resulting from the spatio-temporal evolution of power systems. Through experimental simulations, we show that WNN significantly improves FDIAs detection, training time, and real-time decision making under topological reconfigurations and growing system size with generalization and scalability benefits compared to conventional GNNs.

INDEX TERMS Power grids, graphon neural networks, attacks detection, transfer learning, dynamic graphs.

I. INTRODUCTION AND MOTIVATION

CYBERATTACKS on power systems are on the rise. According to the 2022 Microsoft Digital Defense Report [1], the number of attacks on critical infrastructures around the world doubled from 20% to 40% compared to 2021. These threats are made easier by the digitization of power systems. In recent years, cyberattacks have wreaked havoc on the healthcare, social, and economic sectors, leading to devastating consequences. Of the 45 cybersecurity incidents mentioned targeting the energy sector since 2017, 13 of them occurred in 2022, according to a report by S&P Global Energy Security Sentinel [2]. These reports emphasize how crucial it is to protect these systems and reduce the damage impact. Smart meters are crucial components in modern smart grid systems. They provide real-time data collection, enable remote monitoring and control, enhance grid management, and engage consumers by providing detailed information about their energy usage. However, the power measurements collected by these smart meters are subject to a range of cyberattacks, including data spoofing and injection, denial of service, man-in-the-middle, false data injection attacks (FDIAs), etc. In particular, FDIAs pose the greatest threat to power systems and have gained popularity recently [3]. In FDIAs, an attacker can simply alter or manipulate the power measurements via field devices and transfer the malicious data to the control center. One recent FDIA example is the October 2022 Russian cyberattack on Ukraine's power grid, where hackers gained access to the network hosting a supervisory control and data acquisition (SCADA) instance, and then injected a malware [4].

Data-driven approaches for attacks detection are favored over model-driven approaches for several reasons. Data-driven deep learning methods offer a promising solution for modeling complex system interactions without explicit knowledge. They detect anomalies, identify patterns, and adapt to changes in behavior, making them ideal for dynamic environments. Conversely, model-based approaches detect FDIAs through the use of mathematical models, such as state estimation techniques, where attacks are detected based on the differences between the estimated states and the actual measurements in the system [5]. These methods present several shortcomings, one of which is that they need accurate knowledge of the properties of the underlying systems, which is not always possible [6]. In addition, capturing all the complex interactions among the power system components via equations is very challenging and often impractical.

Power systems can be modeled as graphs, where a node defines a bus, and an edge represents a power line [7], [8]. Such a graph representation can be used to model and analyze complex network topologies by capturing spatial dependencies among different nodes in the power grid, while handling temporal dependencies. Due to their features, graph neural networks (GNNs) have been employed in a variety of power systems applications including power system state estimation, fault analysis, load prediction, and cybersecurity. GNNs represent a powerful modeling framework for cyber attacks detection in large-scale power systems.

Utilizing data-driven techniques like GNNs for detecting FDIAs in power systems presents several challenges:

- **Limited dataset for training:** Research on power systems faces limitations due to restricted access to confidential data. In the U.S., data related to energy production, generation, transmission, and distribution is not available for research, and studies on actual power systems are non-disclosure agreements [9].
- **Assumptions on synthetic systems:** Synthetic power systems have been proposed [10], [11], [12] to mimic the features of actual power grids, but these models rely on assumptions like geographical area, region structure, topological and electrical statistics, and present limited test cases. Scaling to large-scale power systems is difficult, and research is conducted on size-specific synthetic test cases, resulting in case-dependent results.
- **Computational complexity:** The computational complexity of training and testing deep learning methods represents a critical issue, especially that power systems are massive systems that can span entire nations. For instance, the U.S. power grid connects thousands of power plants to 150 million customers via over five million miles of power lines and about 3,300 utilities [13].
- **Dynamicity of power systems:** Power systems are constantly growing to accommodate the increase in the population. Additionally, due to topological changes, seasonal reconfigurations such as integration of renewable energy sources, addition of new buses and lines, etc., power systems are constantly changing in both space and time.
- **Environmental concerns:** Large-scale data training models consume a considerable amount of power, which raises environmental concerns. For instance, large data centers that store and train massive data can consume more than 100 megawatts of power, which is equivalent to powering 80,000 U.S. households [14].

In this paper, we attempt to address all of the above challenges by developing an FDIAs detector using graphon neural networks (WNNs). The acronym WNN is consistent with the terminology found in the literature, referring to the use of graphons, usually represented by $\mathbf{W}$, in neural network architecture to model continuous density functions. Graphons belong to the family of non-parametric graph processing techniques and can be defined as limiting objects of a sequence of graphs with a large number of nodes, in which members of the same graph family share similar structures even if their corresponding number of nodes is different [15]. Since the number of parameters that describe the features and structure of the network does not have to be fixed or even finite, they are regarded as non-parametric models. Graphons offer a more versatile and expressive framework for simulating intricate graph structures because they support an infinite number of parameters. When working with complex real-world networks, where the underlying structures might be difficult to capture by a fixed set of parameters, the non-parametric nature of graphons is especially helpful. The graphon model allows to predict how power systems massively expand over space and time by taking the graphs to their limits—that is, to an infinite number of vertices and edges. This idea from graph theory makes it possible to produce both deterministic and random power graphs that match the observed real power grid. As a matter of fact, in our prior work [16], we estimated the graphon model of the IEEE 39-bus New England system and we showed that any graph sampled from the estimated graphon achieves an average similarity score of 88% with the actual system in terms of topological characteristics such as the eigenvalues spread, graph diameter, betweenness centrality, closeness centrality, nodal degree, and clustering coefficients.

By estimating the graphon model of power systems, we resort to WNNs to overcome the computational burden of training GNNs, since graph convolutions in GNNs involve large matrix operations. The computational efficiency of WNNs is controlled via learning by transference where the learnable parameters of one graph is transferred to another, as will be explained later in this paper.

A. RELATED WORK

To the best of the authors' knowledge, the idea of using WNNs for attacks detection in power systems is novel and has not been investigated in literature. Next, we will summarize the recent papers that proposed attacks detectors using GNNs, and we list their common limitations.

In [17], we proposed a graph neural network-based, scalable, and real-time detector of FDIAs, which outperforms existing solutions in F1 score for standard IEEE testbeds. Moreover, in [18] and [19], we proposed a graph autoencoder (GAE)-based detector against FDIAs and data poisoning in power systems. The detector aimed at improving detection performance and generalization against unseen attacks by capturing spatio-temporal power system features on different topological configurations. However, all these works did not consider the dynamicity of power system topologies as well as the computational burden and environmental concerns of training large-scale systems. In [20], the authors proposed a canonical variate analysis-based detection method to defend against FDIAs in power systems. The method monitored variation of detection indicators before and after attacks, demonstrating effectiveness and accuracy. In [21], a novel unsupervised method for detecting FDIAs in power systems was proposed, combining the strengths of dual graph-convolutional autoencoder and generative adversarial network. In [22], the authors proposed a GNN-based detector that uses auto-regressive moving average type graph filters to detect and localize FDIA in power systems, making them more adaptable to spectral changes compared to polynomial type graph filters like Chebyshev. In [23], the authors introduced a novel online cyber attack situational awareness method by identifying and localizing active attack locations in Operational Technology networks in real-time using a hybrid graph convolutional Long Short-Term Memory (LSTM) and a deep convolutional network. In [24], the authors proposed a false data detection method using a GNN to extract spatial features of power grid topology information and operation data, using an attention mechanism to enhance node representation. In [25], the authors proposed a detection and defense model for load frequency control systems against invisible FDIAs, using attack-detection evolutionary game model and Kalman filtering algorithms. The model used support vector machines and K-Nearest neighbor detection algorithms for optimal control signal. In [26], the authors introduced a graph convolutional network (GCN) framework for detecting FDIAs, which analyzed fluctuating state estimation values and identified attack locations by means of power network graphical structures.

While effective, the mentioned works do not leverage the structural advantages of GNNs; in contrast, our work uses WNNs to enhance detection performance under dynamic topological reconfigurations. Our method leverages transfer learning on large-scale systems with thousands of buses, enhancing generalization and scalability. Additionally, our approach significantly reduces training time and energy consumption, making it more efficient and robust in real-time detection. By focusing on scalable WNN models, we address limited data availability and improve detection capabilities, ensuring improved generalization and energy efficiency.

Next, we summarize the related works to WNNs. In [27], the authors introduced WNNs as limit objects of GNNs. They proved a bound on the difference between GNN output and limit WNNs, which vanishes with increasing node count if the graph convolutional filters are bandlimited. In [28], the authors explored transferability in graphon analysis, proving that fixed GCNs with continuous filters are transferable under graphs that approximate the same graphon. They also proved the asymptotic transferability for graphs approximating unbounded graphon shift operators. In [29], the authors defined WNNs and analyzed their stability to graphon perturbations. They interpreted the WNN as a generating model for GNNs on deterministic and stochastic graphs. They showed that the stability bound decreased asymptotically with graph size. In [30], the authors proposed a new strategy for pooling and sampling on GNNs using graphons, preserving the spectral properties of the graph. By considering graph layers as elements of a sequence convergent to a graphon, node labeling remained consistent and signals were mapped without ambiguity. In [31], the authors demonstrated that the learning distance between GNN and WNN decreases asymptotically with graph size, and gradient descent followed the WNN's learning direction when training on growing graphs. The authors proposed a novel algorithm for learning GNNs on large-scale graphs, gradually increasing the graph size during training. In [32], the authors proposed a GNN training method using resampling from a graphon estimate from the underlying network data. This mitigated over-smoothing in GNNs, rendering the framework computationally efficient with minimal required tuning.

Existing literature employed GNNs for intrusion detection in power systems. The main issue with GNNs is that they involve large matrix operations. One distinctive feature of GNNs is that the number of parameters is independent of the number of nodes. This is so because graph shifts determine graph convolutions. Consequently, GNNs can be transferred from one graph to another since GNNs parametrization and graph size are independent. However, when the graph changes, particularly when moving to larger graphs, the performance of GNNs may not be preserved [27]. Furthermore, the scalability of large-scale convolutions is limited due to their high computational costs [27]. Moreover, GNNs may not be resilient when the size of the network changes over time due to nodes being added or removed. In addition, exisitng works did not consider dynamicity of system topologies and computational burden of training large-scale systems. To address all of these issues, recent research [27], [28], [29], [30], [31], [32], [33] has focused on WNNs, which are the limiting object of GNNs. WNNs can be learned on very large graphs by leveraging the limit object of a sequence of growing graphs, $\{G_n\}$, called the graphon. When training on large graph of size N, the graph convolutions of GNNs would require $O(N^2)$ matrix multiplications. With WNNs, this reduces to $O(n^2)$ computations, where $n < N$, without compromising optimality [33]. Thus, instead of considering the graph to be a fixed hyperparameter, we can think of it as a learnable parameter with weights.

Our work builds on the findings of [27], [28], [29], [30], [31], [32], and [33] by applying WNNs to power systems, demonstrating their practical benefits in FDIA detection and showing that WNNs improve generalization and scalability compared to GNNs. By enhancing FDIA detection under dynamic conditions, we improve the efficiency and robustness of our detection framework. Additionally, WNN training is significantly more efficient, consuming less energy while maintaining performance, thus enhancing overall detection performance and training efficiency.

B. CONTRIBUTIONS AND ORGANIZATION

The contributions of this paper are summarized as follows:

- Graphon model estimation and sampling: We present a novel method to estimate the graphon model from real power systems. This approach enables accurate sampling of graphs and the statistical assignment of electrical parameters, providing a robust framework for effectively representing power system topologies.
- Generation of training and attack samples: We propose a method to generate benign and FDIA samples using MATLAB's MATPOWER toolbox for power flow analysis and normalized ERCOT load data. Dynamic variations are introduced to time-series data using a normal distribution, modeling realistic system fluctuations. By leveraging topological and temporal variability, this approach creates diverse datasets that closely resemble real-world scenarios, enhancing the robustness and utility of deep learning models in power systems.
- Learning by transference with the WNN Model: We leverage the concept of learning by transference using the WNN model. Our results demonstrate that WNN significantly improves generalization and scalability compared to conventional GNNs. Specifically, WNN enhances FDIA detection performance under various topological reconfigurations and increasing system sizes. Moreover, WNN training time is notably reduced, consuming 60% less energy than GNN, showcasing its efficiency and practical applicability.

With these specific contributions in mind, our paper aims to provide significant advancements in the application of graphon neural networks for power systems, offering theoretical and practical enhancements to the existing approaches in terms of model accuracy and computational complexity.

The remainder of this paper is organized as follows. Section II describes the approach of estimating the graphon from an actual power system. Section III describes the benign and malicious dataset generation. Section IV presents the WNNs architecture. The experimental setup, simulation results, and discussions are provided in Section V. Finally, the paper is concluded in Section VI.

II. GRAPHON MODEL ESTIMATION

In this section, we describe what graphon is, how to estimate a graphon model from an actual power system to sample from it, and how to statistically assign electrical parameters for a convergent power flow.

A. WHAT IS A GRAPHON?

Large-scale graphs are often represented using graphons in the realm of graph theory. They make it possible to examine how graphs behave as they get larger by capturing the edge density between pairs of vertices. Graphons are useful to investigate and predict the features of complicated networks. A graphon is a symmetric measurable function $\mathbf{W} : [0, 1]^2 \to [0, 1]$ that maps the unit square to the unit interval. It is comparable to the weight matrix $\mathbf{W}$ of an infinitely large graph, where the weights of the edges are denoted by $\mathbf{W}(u_i, u_j) = \mathbf{W}(u_j, u_i)$ for undirected graphs [15], and the node variables are $(u_i, u_j) \in [0, 1]$. Each pair of nodes (u_i, u_j) is assigned a probability $0 \leq \mathbf{W}(u_i, u_j) \leq 1$ for connecting them on the large sampled graph. This becomes useful when creating large-scale dense graphs. Parameterizing graph operations on graph data is done through the graph shift operator (GSO), $\mathbf{S} \in \mathbb{R}^{n\times n}$, where GSO can represent the adjacency matrix or the graph Laplacian. Thus, the sparsity pattern of a graph G is encoded in GSO $[\mathbf{S}] = s_{ij} \neq 0$ when (i, j) belongs to the set of edges in G [29].

There are two approaches for sampling graphs from the graphon: i) deterministic and ii) stochastic. To sample a deterministic graph of size n, a regular partition $u_i = (i - 1)/n$ of $[0, 1]$ for $1 \leq i \leq n$ is built. Then, each point u_i is assigned to node i. Each GSO element $[\bar{\mathbf{S}}_n]_{ij}$ of a deterministic graph G_n is equal to $\mathbf{W}(u_i, u_j)$. As for sampling a stochastic graph, edges connecting a pair of nodes (u_i, u_j) are sampled from a Bernoulli distribution. Thus, each GSO element is a Bernoulli random variable as $[\mathbf{S}_n]_{ij} \sim \text{Bernoulli}([\bar{\mathbf{S}}_n]_{ij})$ [29]. Since $\mathbf{S}$ is symmetric, it can be diagonlized as $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$, where Λ is a diagonal matrix containing the graph eigenvalues and $\mathbf{V}$ stands for the graph eigenvectors or graph spectral basis.

In an analogy to a graph signal that associates a value to a node, we define a graphon signal X in the graphon domain $L_2([0, 1])$. Graphon signals can be interpreted as generating models for graph signals supported on either the deterministic graph or the stochastic graph, or can be thought of as limits of graph signals supported on graphs converging to a graphon [34]. Moreover, similar to GSO, we can define the graphon shift operator (WSO) as [29]

$$(T_{\mathbf{W}}X)(v) := \int_0^1 \mathbf{W}(u, v)X(u)du, \tag{1}$$

where $T_{\mathbf{W}}$ is a linear Hilbert-Schmidt operator that defines the stride of convolution or the interval between successive applications of the convolutional filter to the input signal. In the operator's spectral basis, we can use eigen decomposition of $\mathbf{W}$ to express it as $\mathbf{W}(u, v) = \sum_{i\in\mathbb{Z}\setminus\{0\}} \lambda_i\varphi_i(u)\varphi(v)$ and

$$(T_{\mathbf{W}}X)(v) = \sum_{i\in\mathbb{Z}\setminus\{0\}} \lambda_i\varphi_i(v) \int_0^1 \varphi(u)X(u)du, \tag{2}$$

where λ_i are the eigenvalues ordered in decreasing order with associated eigenfunctions φ_i. As $i \to \infty$, the eigenvalues of the graphon accumulate near zero [35, Theorem 3,Ch. 28].

Graphon convolutions refer to the process of repeatedly applying WSO and taking their weighted sum. The graphon convolution filter is expressed as $Y = h *_{\mathbf{W}} X = \sum_{k=0}^{K-1} h_k (T_{\mathbf{W}}^{(k)} X)(v)$, with $(T_{\mathbf{W}}^{(k)} X)(v) = \int_0^1 \mathbf{W}(u, v)(T_{\mathbf{W}}^{(k-1)} X)(u) du$, where $h = [h_0, \cdots, h_{K-1}]$ are the graphon filter convolution coefficients; $*_{\mathbf{W}}$ is the convolution operation with WSO $\mathbf{W}$; and $T_{\mathbf{W}}^{(0)} = I$ is the identity.

Any one of the consistent estimators proposed in the literature can be used to estimate the graphon model of a given power system: Stochastic blockmodel approximation (SBA) [36], Universal Singular Value Thresholding (USVT) concept [37], Sort and Smooth (SAS) [38], Largest Gap [39], and Matrix Completion [40].

Estimating a graphon for a power system involves representing the power grid's connectivity through graph theory, extracting meaningful structures, and smoothing them for sampling. The adjacency matrix of the power system graph is derived and processed via the SAS function, which uses empirical degree sorting, histogram filtering, and total variation denoising to estimate the graphon's structure. The resulting graphon matrix encapsulates the probabilistic connectivity pattern of the network. A sampled graph is then generated by creating a synthetic adjacency matrix based on the smoothed graphon and the desired sample size. Random permutations and component analysis are performed on the sample to study its structure, capturing the essential connectivity traits of the original grid. This approach combines statistical and graphical techniques to model and analyze complex network behavior.

B. STATISTICAL ASSIGNMENT OF ELECTRICAL PARAMETERS

After estimating the graphon, we can uniformly draw n samples from $[0, 1]$ representing node variables and map them to node labels. Next, edges are added between every pair of nodes, (u_i, u_j), based on the edge probability, $\mathbf{W}(u_i, u_j)$, that was determined from the estimated graphon. After obtaining the topological configuration of the sampled graph, we resort to a statistical approach to assign electrical parameters to nodes and edges. We use the exponential distribution [41] to generate a random set of active/reactive power values. At this stage, we have the normalized active/reactive power values and nodal degree for each node in the real and sampled systems. By comparing the probability mass function (PMF) of the two normalized variables in the sampled power system with the PMF of the normalized variables in the real power system, the load values are re-ordered to the appropriate nodes after the probability values in the two systems match. Next, based on their degrees, buses are assigned the actual unnormalized active/reactive power values. [41]. In addition, to provide line impedance values, we employ the empirical data from IEEE bus systems and the NYISO system in [42, Table 5], which models line impedance using different distributions depending on the system size. Next, a probabilistic match is carried out between the obtained values and the actual real values. Lastly, we run a continuation power flow that progressively increases the loading/generation to check if the system load surpasses the steady-state loading capacity. When the load surpasses the steady-state loading limit, we scale down all load values to obtain a convergent power flow solution.

C. SPATIO-TEMPORAL DYNAMIC POWER GRAPHS

The novelty of our approach lies in the integration of power system characteristics with WNNs, particularly in addressing the spatiotemporal dynamics of power systems. Power systems are continuously evolving in space and time due to topological modifications, seasonal reconfigurations, integration of renewable energy sources, and the addition of new buses and power lines. These dynamic features introduce new challenges that our proposed WNN method is specifically designed to handle.

The general power system graph representation that we developed by employing the graphon allows to address the dynamicity of power topologies. More specifically, to represent dynamic topological changes in power systems graphs over time $t_1, t_2, \cdots, t_f$, where t_i can represent months or years, we start by sampling a graph of size that matches the final representing size of the system. Then, moving backward in time from t_f down to t_1, each time we partition the topology into smaller subgraphs so that the resulting topology meets the required number of generator and load substations [43]. The graphon approach allows us to model the power system's spatiotemporal dynamics, which are crucial for accurately representing power system behavior over different time scales (e.g., months or years). This capability is essential for detecting FDIAs under varying conditions and configurations, which are common in real-world power systems.

An illustration of dynamic power system topologies is depicted in Fig. 1, where the graphon model of the 2000-bus transmission system geographically situated in the US state of Texas [10] was estimated from the SAS estimator. This power system model captures the intricate interconnections between thousands of buses and transmission lines, reflecting the real-world complexities of power grid infrastructure. Because of its extensive and complex power grid, it is an ideal testbed for anomaly detection methods, allowing exploration of dynamic power system topologies. Then, following real-world power systems evolution statistics [44], we obtained the different spatio-temporal topological configurations representing 10 different time periods such as years. To elaborate further on the decomposition process, we briefly present our approach as follows.

The decomposition process systematically partitions the power system topology into smaller, manageable subgraphs while preserving critical transmission interconnections. This is accomplished by ensuring that the graph's cut edges,

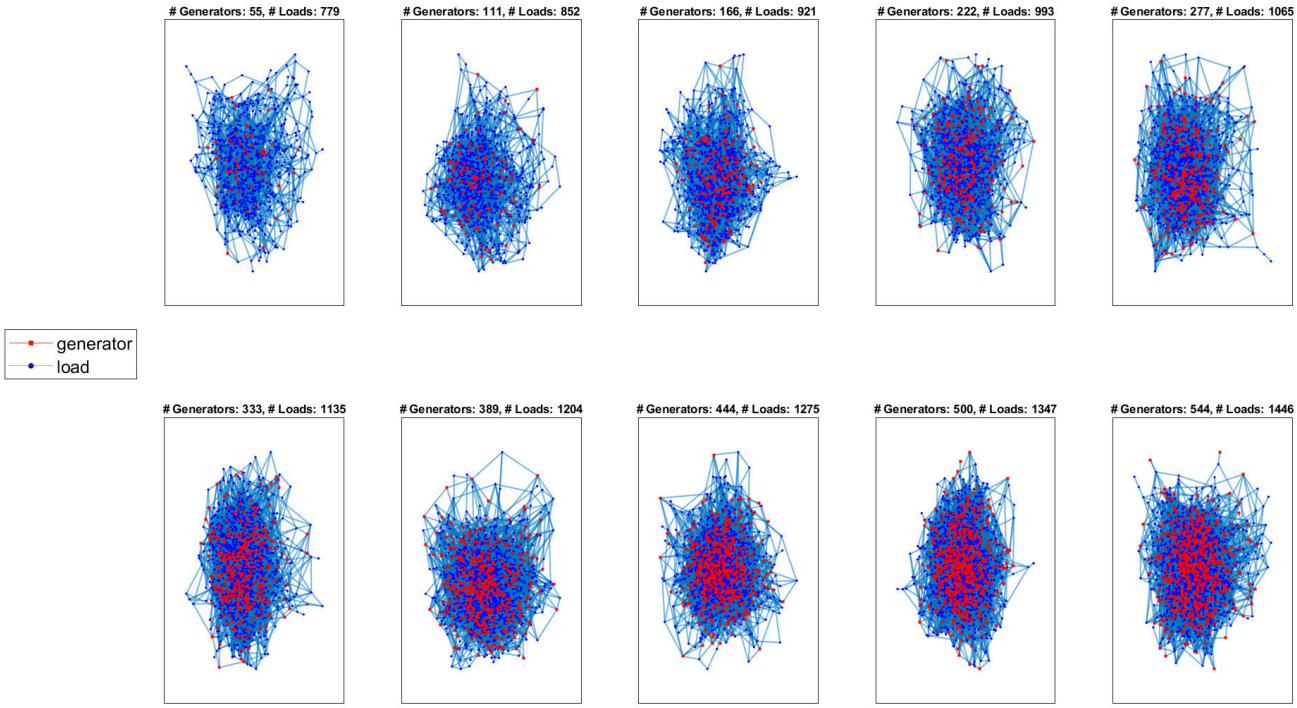

FIGURE 1. Graphon-based spatio-temporal dynamic topological configurations of 2000-bus system (upper left representing the power topology at t_1, while the lower right stands for the final topology at t_{10}).

representing key transmission lines, align with real-world operational requirements.

- Ensuring Connectivity: During the process, we iteratively refine the sampled graph to eliminate disconnected components, ensuring that all nodes belong to a single connected component. This guarantees that the graph is cohesive and suitable for further analysis, maintaining operational interconnectivity between critical elements.
- Power Flow Validation: Once a connected graph is obtained, we integrate MATPOWER to verify power flow feasibility. The process iterates until the generated subgraph satisfies generation and load balance constraints, ensuring the network is physically viable and consistent with realistic grid conditions. This validation step prevents the creation of subgraphs that fail operational or capacity requirements.
- Iterative Reduction Strategy: The reduction strategy begins with high-degree generator nodes and progressively constructs partitions that meet the following criteria:
  - The counts of generators and loads within each subgraph remain within acceptable operational ranges.
  - Subgraphs remain connected to preserve neighborhood relationships and functional coherence.
- Statistical Refinement: Generator and load statistics are computed for each subgraph, and the partitions are iteratively refined to ensure they fall within predefined tolerances. This reduction approach ensures the preservation of both functional and topological structures while balancing computational tractability and physical realism.

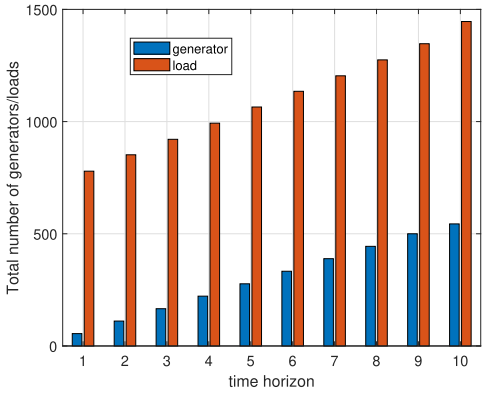

FIGURE 2. Number of loads and generator substations over 10 years.

These steps collectively create subgraphs that are operationally feasible and consistent with realistic power grid conditions. To shed further light on the dynamicity of the topologies, we plot in Fig. 2 a bar chart showing the number of generators and loads in each year. Moreover, Fig. 3 depicts the capacity evolution of loads and generators over the years.

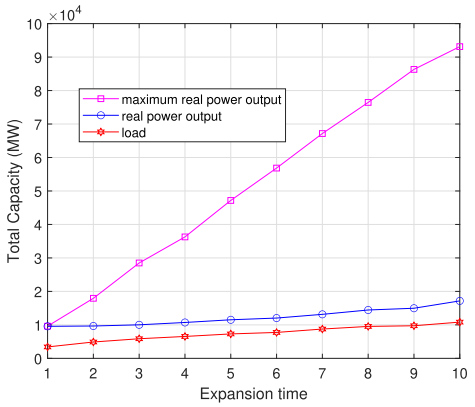

FIGURE 3. Power capacity of loads and generator substations over 10 years.

III. DATASET GENERATION

In this section, we describe the benign and malicious dataset generation, which will be used in the WNN model to develop an FDIAs detector.

A. BENIGN DATASET GENERATION

In the previous section, we described how to estimate the graphon from an actual power system, how to sample from the graphon, and how to assign electrical parameters to nodes and edges. We also showed how we obtained the spatio-temporal dynamic power system graphs belonging to the same graphon family, where each graph reflects the system's topological and power states for a specific period of time. In this section, we generate benign dataset measurements for each of the dynamic power system graphs for purposes of training the WNN.

Given that a power grid can be represented as a graph sampled from the graphon, the best FDIAs detection strategy can be created using WNN techniques. A graph $G_{\mathrm{P}} = (V_{\mathrm{P}}, E_{\mathrm{P}}, \mathbf{W}_{\mathrm{P}})$ can be used to represent the power system, with $|V_{\mathrm{P}}| = N$ representing the set of nodes, either load/demand substations or generator substations. The substations are connected by power lines (edges), which are represented by E_{P}. Power lines are defined by active/reactive power flows that are dictated by line impedance, whereas power nodes are characterized by distinct voltage angles and magnitudes. The weighted adjacency matrix, $\mathbf{W}_{\mathrm{P}} \in \mathbb{R}^{N\times N}$, is determined by the line impedance matrix. Each node consists of spatial and temporal features. The spatial features stem from the spatial distribution of nodes and their connectivities, while the temporal features emanate from the time-series data of active power (MW) and reactive power (MVAr) values.

We generate the temporal features or time-series data needed to model the power flow in every graph topology. We perform the power flow analysis using Newton's method for each topology, using MATLAB's MATPOWER toolbox [45], to determine the active and reactive power flows. The load data profile from the Electric Reliability Council of Texas (ERCOT) [46] is first normalized to a zero mean and unit standard deviation scalar vector $f = [f_1, f_2, \cdots, f_T]$ so that it can be easily adapted to our test system, where f_t is the scalar value at timestamp t. This allows to generate the time-series active and reactive power values. To obtain a dynamic variation in the time-series values (P and Q) with respect to their static case (fixed values), we multiply the active P and reactive Q power values at the previous timestamp by a scaling sample taken from a normal distribution with $1 + 0.025 \times f_t$ mean and 0.01 standard deviation. Consequently, because of the characteristics of the normal distribution, a dynamic range of load values is produced. For each power system graph, we generated 500 power dynamics timestamps.

B. THREAT MODEL

The threat model that we are considering in this paper is FDIAs, where the attackers use malware, compromised devices, or unauthorized access to network infrastructure to manipulate the power measurements collected by the SCADA system for control decisions. By compromising the data integrity, FDIAs can affect voltage stability, imbalance power supply and demand, cause overloading, and potentially result in equipment malfunction [47]. The objective is to develop an FDIAs detector with generalization capability with respect to the system dynamicity and topological configuration. The system operators can use such a robust detector to identify FDIAs, which will enable them to make more informed decisions and enhance the stability of the power grid. Next, we describe the specifics of the malicious datasets and the procedure used to create attack samples that mimic the existence of FDIAs.

We consider five different types of FDIAs: three types of data replay attacks, a general attack, and a random attack. The difference between the altered and true measurement data is kept below a threshold value that is deemed acceptable yet effective to get around the traditional bad data detection of power systems, ensuring the stealthiness of such attacks while avoiding detection.

1) Random Attack: In this method, measurement data is altered by applying a small perturbation value α to benign samples, thereby affecting their integrity. The following equation generates a malicious sample, $X_{\mathrm{s}}(t, i)$, at timestamp t for bus i:

$$X_{\mathrm{s}}(t, i) = X_{\mathrm{b}}(t, i) + \alpha X_{\mathrm{b}}(t, i), \tag{3}$$

where α is a random variable that represents the amount of perturbation applied at random to a specific benign sample, $X_{\mathrm{b}}(t, i)$ at timestamp t and bus i. This attack's primary goal is to destabilize the measurement integrity without triggering detection algorithms. While the changes are minimal, the accumulation of such attacks can cause slight disruptions in system stability, potentially leading to errors in power flow calculations. 2) General attack: the malicious samples are generated as

$$X_{\mathrm{s}}(t, i) = X_{\mathrm{b}}(t, i) + (-1)^{\beta}\zeta\gamma\mathrm{Range}(X_{\mathrm{b}}(t, i)), \tag{4}$$

where ζ and β represent a binary random variable and the magnitude of the attack, respectively; γ is a uniform random variable between 0 and 1; and Range$(X_b(t, i))$ denotes the true measurements range at timestamp t and bus i. In this case, the attack is generated by applying a random perturbation with a specific magnitude, either adding or subtracting from the benign measurement. This attack is designed to introduce more controlled deviations. This type of attack can create significant measurement discrepancies that affect power flow and voltage stability. The system may experience imbalances in supply and demand or incorrect load balancing, causing inefficiencies in power distribution.

3) Replay Attacks: three instances of replay attacks are examined. One-step and random replay attacks are the first two types of attacks that call for choosing a benign sample and substituting a true measurement value from a prior timestamp for it. The one-step replay attack involves repeating data from a previous timestamp $t-1$, where $X_s(t, i)$ is expressed as

$$X_s(t, i) = X_b(t-1, i). \tag{5}$$

The main goal is to manipulate the system by repeating old data, potentially leading to control decisions based on outdated information. This can cause transient instability as the system reacts to "stale" inputs, creating delays in adjusting to real-time conditions.

Data from a randomly chosen prior timestamp $\hat{t}$ is repeated in the random replay attack, where $X_s(t, i)$ is expressed as

$$X_s(t, i) = X_b(t-\hat{t}, i). \tag{6}$$

This attack can be harder to predict and its impact can vary depending on the data reused but may lead to unexpected fluctuations in system performance, affecting load balancing and voltage regulation.

Repeating a single attack sample at a time was the goal of the first two replay attacks. The last type of replay attack is interval replay attack, which introduces a sequence of consecutive attack samples that adhere to benign patterns, and therefore regarded as stealthier because they repeat a series of benign readings. In particular, a sequence of consecutive benign samples $[X_b(t_n, i) \cdots X_b(t_m, i)]$ within time interval $[t_n, \cdots, t_m]$ are replaced with a sequence of consecutive true measurement values from earlier timestamps $[\hat{t}_n, \cdots, \hat{t}_m]$. Mathematically, we can express interval replay attack as

$$[X_s(t_n, i) \cdots X_s(t_m, i)] = [X_b(\hat{t}_n, i) \cdots X_b(\hat{t}_m, i)]. \tag{7}$$

Because this attack mimics benign patterns, it is stealthier and more difficult to detect. However, it can cause long-term disruptions in system performance, particularly in dynamic environments where real-time adjustments are critical.

IV. GRAPHON NEURAL NETWORKS

One distinctive feature of GNNs is that the number of parameters is independent of the graph's size due to graph convolutions being determined by graph shifts. Consequently, GNNs can be transferred from one graph to another since GNN parametrization and graph size are independent. However, when the graph changes, particularly when moving to larger graphs, the performance of GNNs might not be maintained [27]. Furthermore, the scalability of large-scale convolutions is limited due to their high computational costs [27]. Moreover, GNNs might not be resilient if the graph size fluctuates over time as a result of new or deleted nodes. Recent works [27], [28], [29], [30], [31], [32] have focused on WNNs, the limiting object of GNNs, to address all these issues. By utilizing the graphon, which is the limit object of a sequence of growing graphs $\{G_n\}$, WNNs can be trained on extremely large graphs.

WNNs can be transferred because the underlying graphs have similar structural properties. A fixed error term associated with training performance and a transferability constant are used to quantify WNN's performance guarantee [27]. The graphon variability, width, depth, and convolution filter parameters all affect the transferability constant. For filters to be transferrable, GNN filters need to match the eigenvalues of the source and target graphs. The eigenvalues of every sampled graph G_n converge to the graphon eigenvalues as the number of nodes gets closer to infinity. For small eigenvalues, matching becomes challenging. This means that some spectral components cannot be transferred, even for large graphs, because the graphon's eigenvalues accumulate near zero. Asymptotically, the transferability constant rises for filters with narrow passing bands. In [27], it was demonstrated that the smaller graph dominates the transferability bound and that the fixed error term and transferability constant decay with $O(1/\sqrt{\min(n_1, n_2)})$, where n_1 and n_2 are the sizes of the transferable learning graphs.

With learning by transference, the transfer learning approach involves expanding the graph size every epoch. The learning direction (gradients) on the graph and on the graphon are aligned if GNNs have a large number of nodes [31], [33]. We aim for a small discretization of the gradient in order to follow the gradient direction of the graphon learning problem. We begin with a small graph size and increase it step by step until it reaches the total number of nodes; the upper bound on the number of nodes to be added is set by the data that is currently available. A term that asymptotically decreases with graph size bounds the distance between gradient descent steps on the WNNs and on the trained family of GNNs. The iteration on the graphon learning problem will follow the true gradient if the expected difference between gradients is small. Then, we keep increasing the number of nodes until the norm of the GNN gradient is smaller than the non-transferrable constant.

Using the learning by transference characteristic of WNNs, we will start by training smaller graphs using a hybrid GNN and LSTM and then progressively increasing their sizes as shown in Fig. 4. For each GNN-LSTM, the graph convolutional layers extract the spatial features from the graph G_n, while the temporal data measurement values $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$ are captured by the LSTM module. Each time the model is trained, the learnable parameters are passed to the next

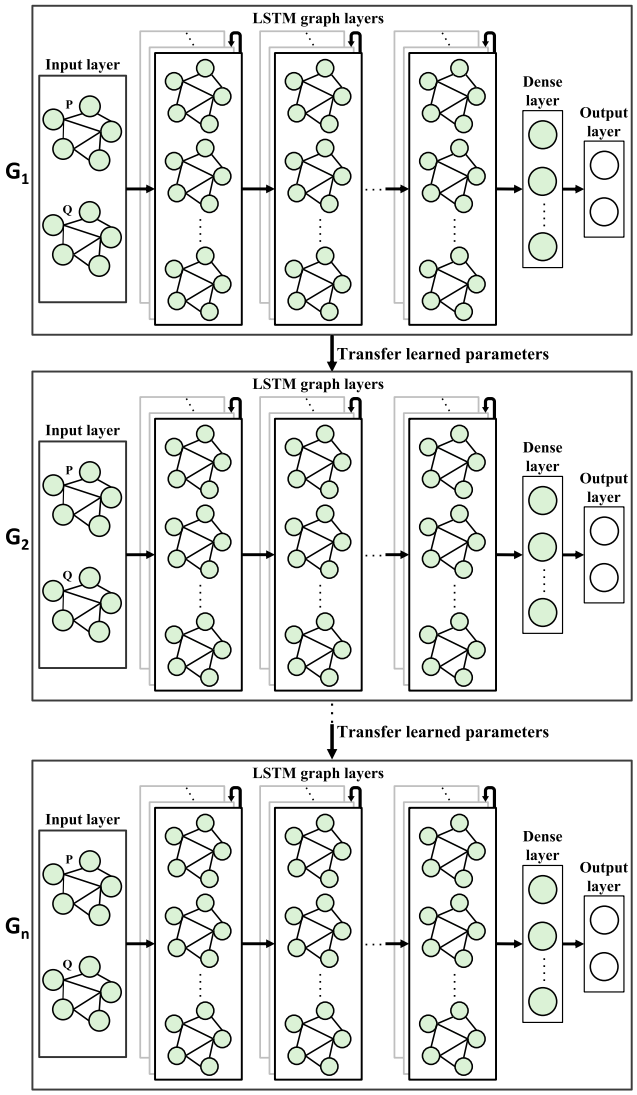

FIGURE 4. **Illustration of the proposed WNN detection.**

larger graph. This iterative approach allows i) to reduce the computation burden of conventional GNNs, ii) to address the dynamicity of topological reconfigurations as new nodes and edges are added in the system, and iii) to deal with the scalability issue of large-scale complex systems.

V. EXPERIMENTAL RESULTS

This section presents the experimental setup and the used metrics to evaluate the models along with the optimal hyperparameters. Then, a comparison between the benchmark GNN-LSTM (simply referred to as GNN) and the proposed WNN model is provided in terms of detection performance, generalization abilities, scalability, and computational complexity.

A. EXPERIMENTAL SETUP

We compare the performance of the proposed WNN model with a traditional convolutional GNN model as a benchmark. Both models are trained, validated, and tested on all the ten topological configurations. For each configuration, 70%, 10%, and 20% of the data is used for training, validation, and testing, respectively. To ensure results generalization, k-fold cross-validation, with $k = 10$, is carried out for each topological configuration. To avoid bias, the number of benign and malicious samples is equal and samples from all attack types are equally injected. For the WNN model, unlike traditional GNN, the learned parameters are carried out from one graph to the other, which results in boosting the detection performance and model efficiency, as will be seen in Sections V-D and V-E.

B. DETECTION PERFORMANCE METRICS

To evaluate the models comprehensively, we use the detection rate (DR), which reflects how well the model identifies attack samples. Additionally, we report the false alarm rate (FAR) to determine the proportion of benign instances that are incorrectly identified as attack instances. We also report the accuracy (ACC) of the models to reflect how well they identify each sample type [48].

C. OPTIMAL HYPERPARAMETERS

To determine the optimal hyperparameter settings for the GNN and WNN models, we use a sequential grid-search hyperparameter selection algorithm [49]. Each hyperparameter is selected in a stage from a predefined list of possible hyperparameter values. The hyperparameter value that provides the highest DR against the validation set is selected [19]. Specifically, the optimal number of layers for the models is 6, which is selected from $\{4, 5, 6, 8\}$. The optimal number of units is 32, which is selected from $\{16, 32, 64\}$. The optimal dropout rate is 0.4, which is selected from $\{0, 0.2, 0.4\}$. The optimal optimizer is Adam, which is selected from {SGD, Adam, Adagrad, Rmsprop}. The optimal activation function is Relu, which is selected from {Relu, Sigmoid, ELU, Tanh}.

D. DETECTION PERFORMANCE

Table 1 reports the detection performance of the GNN and WNN models against the ten topological configurations. Due to the parameter transfer learning capabilities of the WNN model, it is able to provide improved detection performance by up to 4.1%, 3.7%, and 4.7% in DR, FAR, and ACC, respectively, compared to the GNN model. Fig. 5 visualizes the generalization ability and scalability of the WNN model compared to the GNN model. The WNN model generalizes well over multiple graph topological configurations while offering improved detection performance. The WNN model is capable of maintaining consistent detection accuracy and reliability, even in the presence of diverse topological configurations. This ability underscores the model's reliability in adapting to different network environments and reinforces its effectiveness in detecting anomalies under varying conditions.

TABLE 1. Detection performance of WNN compared to GNN.

Metric	Model	Topology									
		1	2	3	4	5	6	7	8	9	10
DR (%)	GNN	86.3	87.2	89	91.4	91.8	92.5	93.1	93.9	94.1	94.1
	WNN	86.3	87.5	90.4	92.8	93.1	94.6	96.7	97	97.7	98.2
FAR (%)	GNN	18.4	16.2	15.3	13.8	12.4	10.8	7.1	5.6	5.5	5.5
	WNN	18.4	15.1	13.4	10.2	8.5	7.2	4.8	2.3	2	1.8
ACC (%)	GNN	85.1	86.1	88.4	90.3	90.9	92.1	92.8	93.2	93.2	93.2
	WNN	85.1	86.7	89.7	91.8	93	94.2	95.8	96.8	97	97.9

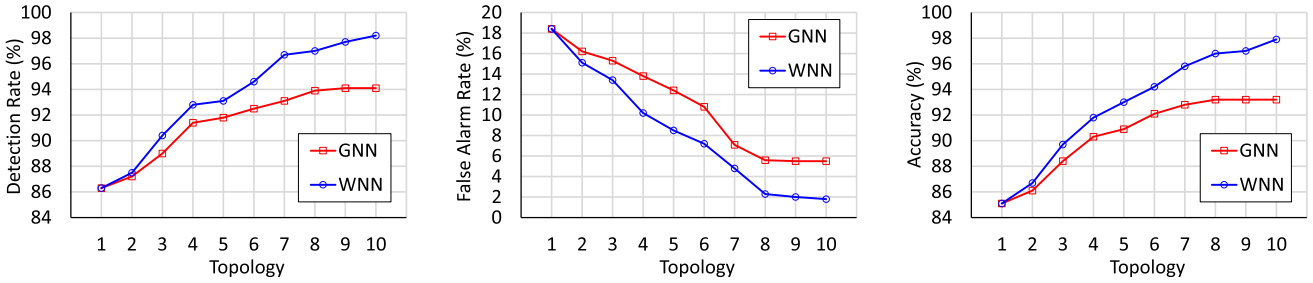

FIGURE 5. Detection performance of the WNN and GNN models.

Moreover, the WNN model offers scalability to larger systems since detection performance improves as the system size increases. Specifically, with the largest system size, the WNN model offers enhanced detection performance by 11.9%, 16.6%, and 12.8% in DR, FAR, and ACC, respectively, compared to the smallest system. This improvement underscores the model's ability to effectively handle the increased complexity and size of larger power grid networks, demonstrating its scalability and suitability for real-world deployment.

In addition, it is worth mentioning that WNN offers enhanced generalization ability and scalability compared to the GNN model since the detection performance of the former keeps improving steeply in the presence of topological reconfigurations and increase in the system size. However, the detection performance of the GNN model saturates (reaches its maximum detection performance) after the seventh topology, while the detection performance of the WNN model keeps improving as system size grows. This distinction underscores the WNN model's superior ability to adapt to changing network conditions and its potential for continued improvement as system size grows. These capabilities and the improvement in the WNN model's detection performance are caused by the model's ability to transfer parameters between graphs that are all members of the same graphon family as the system's size grows. This mechanism enables the model to leverage knowledge learned from smaller networks to improve detection accuracy in larger ones, highlighting the importance of parameter sharing and knowledge transfer in achieving scalability and reliability.

To summarize, the primary advantage of the proposed WNN model lies in its ability to transfer learned parameters across different topological configurations, which significantly enhances its generalization across various system sizes and topologies. Unlike traditional neural networks, such as GNNs, which may struggle with adaptation to new topological changes, the WNN model leverages knowledge transfer from one graph to another within the same graphon family. This transfer mechanism allows the WNN model to maintain high detection performance as the system size increases or as the network topology changes, making it more scalable and adaptable in real-world applications. As a result, the WNN model outperforms existing neural networks in terms of both detection accuracy and computational efficiency, especially in large and dynamic network environments.

E. COMPUTATIONAL COMPLEXITY

Training and testing the models are carried out on a personal machine with an RTX 2080 hardware accelerator using Python. Therefore, the numerical results presented may vary depending on the used computational resources while offering similar trends. The offline training and real-time decision (testing) time with the energy consumption are discussed next.

1) TRAINING AND DECISION TIME

In Fig. 6, we plot the offline training time in hours (hr) along with the online (real-time) decision time in milliseconds (ms). For the GNN model, training time of the largest system takes 7 times more hours than training the smallest system, whereas it takes 3 times for the WNN model since parameter knowledge is being transferred among the different graphs. Specifically, training the WNN model requires significantly less number of hours, around 60% less compared to the GNN model with the largest system. For the WNN model, it takes around 3 ms (which satisfies the power systems latency

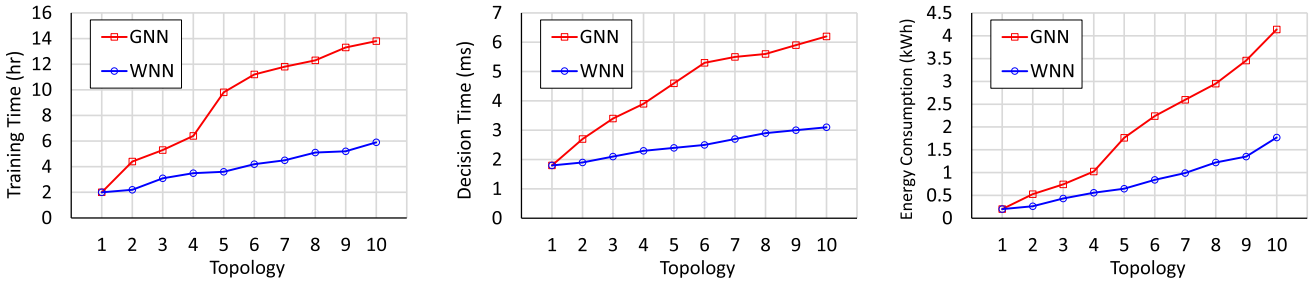

FIGURE 6. **Computational comparisons between the WNN and GNN models.**

requirements [19]) to provide a decision on a sample, which is half the amount of time that takes the GNN model.

In our analysis, we observed that the WNN model significantly outperforms the GNN model in terms of training time, particularly for larger systems. WNN model's reduced training time directly translates to improved scalability compared to the GNN model. This improvement is attributed to WNN's parameter transfer mechanism, which allows the model to efficiently handle increasing system sizes while maintaining computational efficiency.

2) ENERGY CONSUMPTION

Transferring the parameters among the different topological configurations not only leads to enhanced detection performance, but also enhances the training efficiency in terms of the energy consumption during the training stage. Fig. 6 also plots the energy consumed in kiloWatt hours (kWh) during the training stage of the WNN and GNN models. For the GNN model, training the largest system requires 20 times more energy than the smallest system, whereas for the WNN model, it requires 8 times to train the largest system compared to the smallest one. This means that training the WNN model consumes around $(20 - 8)/20 = 0.6$ or 60% less energy compared to training the GNN model with the largest graph size. However, the energy consumption here is estimated indirectly through the computation time, which serves as a proxy for energy usage. We inferred energy consumption from processing time on the hardware, assuming a consistent system configuration and a direct relationship between computation time and energy use. These energy savings are approximations based on this assumption; however, direct energy measurements will be necessary in future studies to validate these comparisons.

VI. CONCLUSION

In this paper, we proposed the use of WNN to exploit the learning by transference on the sequence of growing graphs belonging to the same graphon family. We showed that when compared to conventional GNN, WNN significantly improves FDIAs detection under topological reconfigurations and growing system size with benefits in generalization and scalability. Specifically, the training and real-time decision making were cut by more than half compared to GNN. Moreover, training on a WNN was found to be much more energy efficient than GNN as it consumed 60% less energy. These findings shed light on the promising potential of WNNs as a viable alternative to GNNs in various applications. Further research and development in this area could lead to the wide adoption of WNNs, revolutionizing the field of artificial intelligence.

REFERENCES

[1] T. Burt. (Nov. 2022). *Nation-state Cyberattacks Become More Brazen As Authoritarian Leaders Ramp Up Aggression*. [Online]. Available: https://blogs.microsoft.com/on-the-issues/2022/11/04/microsoft-digital-defense-report-2022-ukraine/

[2] L. James. (May 2023). *Energy Sector: More Cyber Attacks in 2022 Than Ever Before*. [Online]. Available: https://www.power-and-beyond.com/energy-sector-more-cyber-attacks-in-2022-than-ever-before-a-a53dfeb9e1a85d8a0710a010c7a7e7d3/

[3] A. A. Habib, M. K. Hasan, A. Alkhayyat, S. Islam, R. Sharma, and L. M. Alkwai, "False data injection attack in smart grid cyber physical system: Issues, challenges, and future direction," *Comput. Electr. Eng.*, vol. 107, Apr. 2023, Art. no. 108638.

[4] D. Antoniuk. (Nov. 2023). *Ukraine Energy Facility Took Unique Sandworm Hit on Day of Missile Strikes, Report Says*. [Online]. Available: https://therecord.media/sandworm-attack-ukraine-energy-facility-missile-strikes

[5] H. Long, Z. Wu, C. Fang, W. Gu, X. C. Wei, and H. Y. Zhan, "Cyber-attack detection strategy based on distribution system state estimation," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 4, pp. 669–678, Jul. 2020.

[6] F. Mohammadi, "Emerging challenges in smart grid cybersecurity enhancement: A review," *Energies*, vol. 14, no. 5, p. 1380, Mar. 2021.

[7] T. Ishizaki, A. Chakrabortty, and J.-I. Imura, "Graph-theoretic analysis of power systems," *Proc. IEEE*, vol. 106, no. 5, pp. 931–952, May 2018.

[8] T. Werho, V. Vittal, S. Kolluri, and S. M. Wong, "Power system connectivity monitoring using a graph theory network flow algorithm," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4945–4952, Nov. 2016.

[9] Federal Energy Regulatory Commission. *Critical Energy/Electric Infrastructure Information (CEII)*. Accessed: Jan. 27, 2024. [Online]. Available: https://www.ferc.gov/ceii

[10] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye, and T. J. Overbye, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258–3265, Jul. 2017.

[11] R. Atat, M. Ismail, M. F. Shaaban, E. Serpedin, and T. Overbye, "Stochastic geometry-based model for dynamic allocation of metering equipment in spatio-temporal expanding power grids," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2080–2091, May 2020.

[12] D. Deka, S. Vishwanath, and R. Baldick, "Analytical models for power networks: The case of the western US and ERCOT grids," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2794–2802, Nov. 2017.

[13] C. Lo. (Mar. 2022). *Upgrading the Us Power Grid for the 21st Century*. [Online]. Available: https://www.power-technology.com/features/featureupgrading-the-us-power-grid-for-the-21st-century-4866973/?cf-view

[14] S. Marcacci. (Mar. 2020). *How Much Energy Do Data Centers Really Use?* [Online]. Available: https://energyinnovation.org/2020/03/17/how-much-energy-do-data-centers-really-use/

[15] J. Eldridge, M. Belkin, and Y. Wang, "Graphons, mergeons, and so on!" in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[16] R. Atat, M. Ismail, and E. Serpedin, "Graphon-based synthetic power system model and its application in system risk analysis," in *Proc. IEEE Int. Smart Cities Conf. (ISC)*, Sep. 2023, pp. 1–6.

[17] O. Boyaci et al., "Graph neural networks based detection of stealth false data injection attacks in smart grids," *IEEE Syst. J.*, vol. 16, no. 2, pp. 2946–2957, Jun. 2022.

[18] A. Takiddin, R. Atat, M. Ismail, O. Boyaci, K. R. Davis, and E. Serpedin, "Generalized graph neural network-based detection of false data injection attacks in smart grids," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 618–630, Jun. 2023.

[19] A. Takiddin, M. Ismail, R. Atat, K. R. Davis, and E. Serpedin, "Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids," *IEEE Trans. Artif. Intell.*, vol. 5, no. 3, pp. 1287–1301, Mar. 2024.

[20] C. Pei, Y. Xiao, W. Liang, and X. Han, "Detecting false data injection attacks using canonical variate analysis in power grid," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 971–983, Apr. 2021.

[21] H. Feng, Y. Han, F. Si, and Q. Zhao, "Detection of false data injection attacks in cyber-physical power systems: An adaptive adversarial dual autoencoder with graph representation learning approach," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.

[22] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 807–819, Jan. 2022.

[23] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sep. 2023.

[24] X. Li, Y. Wang, and Z. Lu, "Graph-based detection for false data injection attacks in power grid," *Energy*, vol. 263, Jan. 2023, Art. no. 125865.

[25] Z. Zhang, J. Hu, J. Lu, J. Cao, and F. E. Alsaadi, "Preventing false data injection attacks in LFC system via the attack-detection evolutionary game model and KF algorithm," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 6, pp. 4349–4362, Nov. 2022.

[26] E. Vincent, M. Korki, M. Seyedmahmoudian, A. Stojcevski, and S. Mekhilef, "Detection of false data injection attacks in cyber–physical systems using graph convolutional network," *Electric Power Syst. Res.*, vol. 217, Apr. 2023, Art. no. 109118.

[27] L. Ruiz, L. Chamon, and A. Ribeiro, "Graphon neural networks and the transferability of graph neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1702–1712.

[28] S. Maskey, R. Levie, and G. Kutyniok, "Transferability of graph neural networks: An extended graphon approach," *Appl. Comput. Harmon. Anal.*, vol. 63, pp. 48–83, Mar. 2023.

[29] L. Ruiz, Z. Wang, and A. Ribeiro, "Graphon and graph neural network stability," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 5255–5259.

[30] A. Parada-Mayorga, L. Ruiz, and A. Ribeiro, "Graphon pooling in graph neural networks," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Jan. 2021, pp. 860–864.

[31] J. Cerviño, L. Ruiz, and A. Ribeiro, "Learning by transference: Training graph neural networks on growing graphs," *IEEE Trans. Signal Process.*, vol. 71, pp. 233–247, 2023.

[32] Z. Hu, Y. Fang, and L. Lin, "Training graph neural networks by graphon estimation," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2021, pp. 5153–5162.

[33] J. Cerviño, L. Ruiz, and A. Ribeiro, "Training graph neural networks on growing stochastic graphs," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2023, pp. 1–5.

[34] L. Ruiz, L. F. O. Chamon, and A. Ribeiro, "The graphon Fourier transform," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5660–5664.

[35] L. V. Kantorovich and G. P. Akilov, *Functional Analysis*. Amsterdam, The Netherlands: Elsevier, 2016.

[36] E. M. Airoldi, T. B. Costa, and S. H. Chan, "Stochastic blockmodel approximation of a graphon: Theory and consistent estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, Nov. 2013, pp. 1–9.

[37] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *Ann. Statist.*, vol. 43, no. 1, pp. 177–214, Feb. 2015.

[38] S. H. Chan and E. M. Airoldi, "A consistent histogram estimator for exchangeable graph models," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2014, pp. 208–216.

[39] A. Channarond, J.-J. Daudin, and S. Robin, "Classification and estimation in the stochastic blockmodel based on the empirical degrees," *Electron. J. Statist.*, vol. 6, pp. 2574–2601, Jan. 2012.

[40] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[41] S. H. Elyas, Z. Wang, and R. J. Thomas, "On the statistical settings of generation and load in a synthetic grid modeling," in *Proc. 10th Bulk Power Syst. Dyn. Control Symp.*, Sep. 2017, pp. 1–7.

[42] Z. Wang, A. Scaglione, and R. J. Thomas, "Generating statistically correct random topologies for testing smart grid communication and control networks," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 28–39, Jun. 2010.

[43] H. K. Temraz, M. M. A. Salama, and V. H. Quintana, "Application of partitioning techniques for decomposing large-scale electric power networks," *Int. J. Electr. Power Energy Syst.*, vol. 16, no. 5, pp. 301–309, Oct. 1994.

[44] KAHRAMAA. *Qatar General Electricity and Water Corporation*. Qatar Gen. Electr. Water Corp. Accessed: Jan. 27, 2024. [Online]. Available: https://www.km.qa/AboutUs/Pages/ElectricitySector.aspx

[45] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[46] *Electric Reliability Council of Texas*. Accessed: Jan. 27, 2024. [Online]. Available: https://www.ercot.com/mktinfo/loadprofile/alp

[47] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–33, May 2011.

[48] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4106–4117, Sep. 2022.

[49] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Robust electricity theft detection against data poisoning attacks in smart grids," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2675–2684, May 2021.