Transferrable Graphon-Based Detection of Adversarial Samples in Power Grid Networks

Salma Aboelmagd*, Alyssa Traina†, Rachad Atat‡, Abdulrahman Takiddin†
*Department of Electrical & Computer Engineering, Florida State University, Tallahassee, Florida, USA
†Center for Advanced Power Systems, Florida State University, Tallahassee, Florida, USA
‡Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon
{saboelmagd, atraina, a.takiddin}@fsu.edu, rachad.atat@lau.edu.lb

***Abstract*—Modern grid infrastructures rely heavily on connected devices and real-time communication for efficient and secure operation. While traditional machine learning, deep learning, and graph-based methods have shown success in detecting classic false data injection attacks (FDIAs), which manipulate readings using simple perturbation strategies, their performance degrades significantly against evasion attacks where adversaries generate adversarial samples to fool detection models without requiring system knowledge. Moreover, traditional graph-based approaches suffer from high computational overhead when trained on large power system topologies. In this work, we propose implementing a graphon neural network (WNN)-based detector against evasion attacks, assuming a realistic threat scenario where the adversary has no prior system knowledge. Unlike graph neural networks, which require retraining when the topology changes, WNNs enable efficient and transferable learning by modeling families of graphs that share structural properties. Experimental results demonstrate that our WNN-based detector maintains robust detection against evasion attacks with a 1–2% deterioration in detection rate, compared to a 9–10% deterioration in benchmark models, while enhancing decision-making latency by 116% and reducing training time by 133% compared to benchmarks.**

***Index Terms*—Adversarial sample, smart grid communications, graphon neural network, machine learning, power grid, transfer learning.**

I. Introduction

Modern power grids are complex cyber-physical systems that increasingly rely on distributed sensors and real-time communication networks to support secure and efficient operation [1]. The integrity of sensor data exchanged across these networks is essential for reliable monitoring and decision-making. However, as grid infrastructure evolves into a highly connected environment, it becomes more vulnerable to cyber threats targeting communication grid devices and detection mechanisms. For instance, an attack targeting supervisory control and data acquisition (SCADA) systems disrupted communication with utility control centers in multiple states in the US [2]. Unlike classic false data injection attacks (FDIAs), which manipulate readings using simple attack functions, evasion attacks exploit vulnerabilities in detection models by dynamically adapting their attack patterns to bypass detection, ultimately misleading the SCADA system and resulting in inaccurate decisions by system operators [3]. Such threats critically undermine state estimation, disrupting grid operations and stability, requiring a robust and efficient detection framework adaptable to evolving attacks and configurations [4].

A. Related Work

Existing research on cyber attack detection in power grids primarily focuses on classic FDIA detection, but often overlooks adaptability to adversarial scenarios and computational efficiency. We group related works into two categories: detectors targeting classic FDIAs and evasion attacks.

1) Classic FDIA Detectors: A decision tree (DT)-based approach framed FDIA localization as a multi-label classification task using ensemble learning was proposed [5]. A random forest-based detection scheme was introduced to identify cyber attacks in the power grid [6]. A deep latent space clustering model for improved stealthy FDIA detection was proposed [7]. A graph-based FDIA detection model leveraging node and topology-level attention weights was introduced [8]. A graph autoencoder model capturing spatio-temporal correlations and generalizing across unseen topologies was presented [9]. A Hodge aggregation graph neural network (GNN)-based FDIA detector was proposed in [10]. A generalized graph autoencoder-based framework trained on topological variants to improve robustness across system sizes was presented in [11]. Graphon neural network (WNN)-based detection model that generalizes across graph sizes using graphon-based learning was proposed [12]. Other GNN-based methods were proposed for attack classification [13] as well as detection and localization [14].

The aforementioned classic FDIA detectors present at least one of the following limitations: (1) require retraining each time the grid configuration changes, hindering efficiency, (2) assume full attacker knowledge, or (3) overlook evasion attacks, which are designed to fool machine learning models.

2) Evasion Attack Detectors: A robust detection approach using an ensemble of attention-based autoencoders, long-short-term memory (LSTM), and feedforward neural networks (FNN) for electricity theft against evasion attacks was proposed [15]. A GNN-based model was evaluated under adversarial conditions where the attacker had full system knowledge was proposed [16]. A spatio-temporal GNN-based generation and detection of adversarial samples was proposed in [3].

While the aforementioned studies advance evasion attack detection, they are either limited by topology-specific designs, requiring full retraining for new graphs, or assume unrealistic attacker knowledge.

B. Contributions

These limitations motivate the implementation of a scalable and transferable detection framework for adversarial threats in power grids. Thus, we propose a WNN that efficiently detects evasion attacks. Specifically, the contributions of this work are summarized as follows:

- We propose implementing a WNN-based model that learns by transferring parameters across varying power system topologies, enabling the model to detect across graphs of different sizes and structures without retraining while preserving spectral consistency. WNNs leverage the statistical convergence of graph families to enable efficient and scalable learning, requiring only small initial graphs and progressively transferring learned parameters as the system expands.
- We evaluate our WNN-based model against evasion attacks, under a realistic threat scenario where the adversary lacks prior system knowledge, using fast gradient sign method (FGSM) [3], projected gradient descent (PGD) [17], and elastic-net attack to deep neural networks (EAD) [18]. Our results demonstrate that our model remains robust, with a $1-2\%$ deterioration in detection rate (DR), significantly outperforming benchmarks that experience a deterioration of $9-10\%$ against evasion attacks compared to classic FDIAs.
- We leverage learning by transference using WNNs to enhance generalization and scalability in power grid security. Our WNN-based detector reduces decision-making time and training time by 116% and 133%, respectively, compared to benchmarks, highlighting its efficiency and practical applicability.

This paper is outlined as follows. Section II details the generation of spatio-temporal benign, FDIA, and evasion samples. Section III presents the WNN architecture. Section IV presents the experimental setup and simulation results. The paper is concluded in Section V.

II. Dataset Generation

This section outlines the process for generating spatio-temporal benign and malicious datasets, which serve as the foundation for training and assessing the performance of the proposed WNN-based detection model.

A. Spatial Data

a) GNN Graphs vs. Graphon Graphs: Traditional GNNs operate on fixed, discrete graphs, where the structure, including the number of nodes and edges, must remain static throughout training. Graph operations in GNNs rely on fixed adjacency or Laplacian matrices, making them inflexible in scenarios where the graph topology evolves. This poses a serious limitation in dynamic power systems, where substations or transmission lines may be added or removed over time. Since GNNs must be retrained entirely for each new graph configuration, this significantly increases computational costs and reduces practicality for real-time applications. Graphon-based graphs, in contrast, are generated from a continuous function that encodes probabilistic connectivity over the unit interval. Instead of treating each graph as a separate entity, WNNs view individual graphs as samples from a shared underlying structure—a graphon. This enables the model to generalize across graphs of different sizes and structures without retraining. By leveraging this property, WNNs support scalable and efficient learning across a family of related power system topologies, making them especially well-suited for dynamic environments where grid configurations frequently evolve.

b) Graphon: A graphon is a symmetric measurable function $\mathbf{W} : [0,1]^2 \to [0,1]$ that captures edge weights between node pairs in large-scale networks. Each node pair (u_i, u_j), where i and j are indices representing nodes, is assigned a connection probability $0 \leq \mathbf{W}(u_i, u_j) \leq 1$, enabling the construction of scalable and dense graph topologies. The resulting graph can be interpreted as a limit object of an infinitely large graph, where the weight of an edge between two nodes is given by $\mathbf{W}(u_i, u_j) = \mathbf{W}(u_j, u_i)$ in the case of undirected graphs [19].

c) Graphon-sampled Graphs: To sample graphs from the graphon, we can adopt deterministic and stochastic approaches. Graph operations are typically defined using the graph shift operator (GSO), denoted as $\mathbf{S} \in \mathbb{R}^{n \times n}$, which can take the form of either the adjacency matrix or the graph Laplacian. The structure of the graph G is reflected in the sparsity pattern of $\mathbf{S}$, where the element $s_{ij} \neq 0$ if there is an edge between nodes i and j [12]. In the deterministic setting, nodes are uniformly sampled as $u_i = \frac{i-1}{n}$ for $1 \leq i \leq n$ in $[0,1]$ for a graph of size n, and each GSO element $[\bar{\mathbf{S}}_n]_{ij}$ is set to $\mathbf{W}(u_i, u_j)$ for a deterministic graph G_n. In the stochastic case, edges between node pairs are sampled from a Bernoulli distribution, so that each element of the GSO becomes a Bernoulli random variable, $[\mathbf{S}_n]_{ij} \sim \text{Bernoulli}([\bar{\mathbf{S}}_n]_{ij})$ [12]. Since $\mathbf{S}$ is symmetric, it admits an eigen decomposition $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^H$, where Λ contains eigenvalues and $\mathbf{V}$ contains the eigenvectors forming the graph spectral basis. Comparable to graph signals, which assign values to nodes in a graph, a graphon signal X is defined over the graphon domain $L_2([0,1])$. These signals can be viewed as generating functions for signals on either deterministic or stochastic graphs, or as the limiting form of graph signals defined on sequences of graphs that converge to a graphon [20]. WNNs rely on the graphon shift operator (WSO), which serves as the infinite-dimensional counterpart to the GSO used in GNNs. Given a graphon $\mathbf{W}$, the WSO is defined as:

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

where $\mathbf{T_W}$ is a Hilbert-Schmidt integral operator defining the shift transformation in the continuous graphon space, $X(u)$ is the signal function over the node domain, and $\mathbf{W}(u,v)$ represents the weighted connectivity between nodes u and v. The eigenvalue decomposition of the graphon is given by:

$$\mathbf{W}(u,v) = \sum_{i \in \mathbb{Z}\setminus 0} \lambda_i \varphi_i(u)\varphi_i(v), \tag{2}$$

where λ_i are eigenvalues ordered in decreasing magnitude, and φ_i are the corresponding eigenfunctions. As $i \to \infty$, eigenvalues accumulate near zero [21]. Graphon convolutions involve repeatedly applying the WSO and combining the result through a weighted sum. The convolution is defined as $Y = h *_{\mathbf{W}} X = \sum_{k=0}^{K-1} h_k \left(T_{\mathbf{W}}^{(k)} X\right)(v)$, where each recursive application of the shift operator is given by $\left(T_{\mathbf{W}}^{(k)} X\right)(v) = \int_0^1 \mathbf{W}(u, v) \left(T_{\mathbf{W}}^{(k-1)} X\right)(u)\, du$. Here, $h = [h_0, \ldots, h_{K-1}]$ denotes the filter coefficients, $*_{\mathbf{W}}$ represents convolution with the WSO, and $T_{\mathbf{W}}^{(0)} = I$ is the identity. Graphon estimation for power systems aims to model grid connectivity by processing the adjacency matrix through a graphon estimation technique. While several estimation methods exist, such as Universal Singular Value Thresholding (USVT) [22] and Stochastic Blockmodel Approximation (SBA) [23], we adopt the Sort and Smooth method [24] due to it yielding the lowest estimation error. This method applies degree sorting, filtering, and denoising to recover the graphon's structure. A synthetic graph is then sampled from the smoothed graphon, preserving key structural traits of the original grid.

We generate five different power system topologies using the Texas 2000-bus transmission system [12] by sampling graphs of increasing size from the same graphon model. Each topology represents a distinct power system configuration with a different number of nodes. The sampled graphs consist of 600, 900, 1200, 1500, and 2000 nodes. Each graph $G_{\mathrm{n}} = (V_{\mathrm{n}}, E_{\mathrm{n}}, \mathbf{W}_{\mathrm{n}})$ represents a snapshot of the power system, where V_{n} denotes the set of substations, E_{n} is the connecting transmission lines, and $\mathbf{W}_{\mathrm{n}} \in \mathbb{R}^{n \times n}$ is the weighted adjacency matrix derived from line impedances.

B. Temporal Data

We generate measurements for a series of dynamic power system graphs sampled from a graphon. To generate the temporal data, we use load profiles from the Electric Reliability Council of Texas (ERCOT) system [25], normalized to a zero-mean, unit-variance vector $f = [f_1, f_2, \ldots, f_T]$. For each timestamp t, dynamic scaling is applied to the base power flow using a random factor from a normal distribution with mean $1 + 0.025 f_t$ and standard deviation 0.01, introducing realistic variation in load behavior over time. Power flow calculations are performed using Newton's method via the MATPOWER toolbox [26], yielding time-varying active (P) and reactive (Q) power values. We generate 500 temporal snapshots per graph to simulate realistic grid dynamics for model training.

C. Benign Dataset Generation

Using the spatial and temporal structures described above, we generate benign datasets to represent normal power grid operation across all graph configurations. For each dynamic power system graph, time-series sensor readings are collected at every node, reflecting typical conditions. These measurements, which evolve over time and vary across network topologies, constitute the benign data samples used to train the WNN-based detector and the benchmark models. We denote these samples as $\mathbf{X^b} \in \mathbf{X}$, where $\mathbf{X}_{t,n}^{\mathbf{b}}$ represents the benign sample at time t and node n. This dataset captures the spatio-temporal dynamic of normal grid behavior and serves as the foundation for evaluating the ability of the models to distinguish benign from adversarial activity.

D. Malicious Dataset Generation

To assess detection performance, we evaluate two attack types: classic FDIAs and adversarial evasion attacks. While both aim to disrupt the power system or mislead the detection mechanism, they differ fundamentally in strategy. Classic FDIAs directly manipulate measurement values within the power grid to cause instability or mislead control decisions. In contrast, evasion attacks focus on subtly altering inputs to fool machine learning models into misclassifying them as benign. Hence, classic FDIAs are present in the training and testing sets of the models, where a classic FDIA sample $\mathbf{X}_{t,n}^{\mathbf{m}}$ is labeled as '1', whereas adversarial samples $\mathbf{X}_{t,n}^{\mathbf{e}}$, generated using evasion attack functions, are only present in the test set with a benign label of '0' to deceive the detector.

1) Classic FDIA Samples: We generate five types of classic FDIAs, including a general attack, a random attack, and three variants of replay attacks. In each case, the modification to true measurements remains below a detection threshold to maintain stealth against traditional bad data detection mechanisms.

a) Random Attack: The random attack perturbs benign measurements by a small, randomly scaled factor, generated by the following equation $\mathbf{X}_{t,n}^{\mathbf{m}} = \mathbf{X}_{t,n}^{\mathbf{b}} + \alpha \cdot \mathbf{X}_{t,n}^{\mathbf{b}}$, where α is a random variable controlling the perturbation magnitude.

b) General Attack: The general attack introduces noise scaled by the range of observed measurements generated by $\mathbf{X}_{t,n}^{\mathbf{m}} = \mathbf{X}_{t,n}^{\mathbf{b}} + (-1)^{\beta}\, \zeta\, \gamma \cdot \mathrm{Range}(\mathbf{X}_{t,n}^{\mathbf{b}})$, where ζ and β are binary random variables representing the attack's magnitude and direction, respectively; γ is a continuous uniform random variable in $[0, 1]$; and $\mathrm{Range}(\mathbf{X}_{t,n}^{\mathbf{b}})$ is the range of the benign measurements at timestamp t and node n.

c) Replay Attacks: Such attacks include one-step, random, and interval replays. The one-step replay substitutes a measurement from the immediate past $t-1$, where $\mathbf{X}_{t,n}^{\mathbf{m}} = \mathbf{X}_{t-1,n}^{\mathbf{b}}$, or a from an arbitrary prior timestamp $\hat{t}$, where $\mathbf{X}_{t,n}^{\mathbf{m}} = \mathbf{X}_{t-\hat{t},n}^{\mathbf{b}}$. The interval replay replaces a sequence of current readings $[t_n, \ldots, t_m]$ with a corresponding sequence from earlier periods $[\hat{t}_n, \ldots, \hat{t}_m]$, making it more difficult to detect due to its temporal consistency with benign behavior. The interval replay is generated by $[\mathbf{X}_{t_n,n}^{\mathbf{m}}, \ldots, \mathbf{X}_{t_m,n}^{\mathbf{m}}] = [\mathbf{X}_{\hat{t}_n,n}^{\mathbf{b}}, \ldots, \mathbf{X}_{\hat{t}_m,n}^{\mathbf{b}}]$.

2) Adversarial Samples: We generate adversarial samples through three benchmark evasion attack functions, namely, FGSM, PGD, and EAD. To maintain practical attack scenarios, we assume the adversary has no knowledge of system parameters, network topology, or detection model.

a) FGSM Attack: This attack perturbs input data along the direction of the loss function to maximize misclassification. Given an electricity measurement, the adversarial sample is computed as:

$$\mathbf{X}_{t,n}^{\mathbf{e}} = \mathbf{X}_{t,n}^{\mathbf{b}} + \epsilon \cdot \mathrm{sign}\left(\nabla_{\mathbf{X}_{t,n}^{\mathbf{b}}} J(\phi, \mathbf{X}_{t,n}^{\mathbf{b}}, y)\right), \qquad (3)$$

where $\mathbf{X}^{\mathbf{e}}_{t,n}$ is the generated adversarial sample at time step t and node n, $X^{\mathbf{b}}_{t,n}$ is the original input reading, ϵ denotes the perturbation magnitude, sign is applying the signum function, J refers to the model's loss function, ϕ denotes the model parameters, and y is the true label.

b) PGD Attack: This attack is an iterative variant of FGSM, applying multiple small perturbation steps while ensuring the adversarial sample remains within a bounded region [27]. This adversarial sample is computed as:

$$\mathbf{X}^{\mathbf{e},k+1}_{t,n} = \Pi_{\mathcal{B}(\mathbf{X}^{\mathbf{b}}_{t,n},\epsilon)}\Big(X^{\mathbf{e},k}_{t,n} + \delta \cdot \text{sign}\left(\nabla_{\mathbf{X}^{\mathbf{b}}_{t,n}} J(\phi, \mathbf{X}^{\mathbf{e},k}_{t,n}, y)\right)\Big), \tag{4}$$

where $\mathbf{X}^{\mathbf{e},k}_{t,n}$ is the adversarial example at iteration k, the parameter δ represents the step size, and $\Pi_{\mathcal{B}(\cdot)}$ is the projection operator onto the ℓ_∞-ball of radius ϵ centered at the benign sample $\mathbf{X}^{\mathbf{b}}_{t,n}$.

c) EAD Attack: This attack is an optimization-based attack that generates perturbations by balancing sparsity and distortion. EAD introduces elastic-net regularization, which incorporates both ℓ_1-norm or ℓ_2-norm penalties. The optimization problem for EAD is formulated as:

$$\min_{\mathbf{X}^{\mathbf{e}}_{t,n}} \quad c \cdot J(\phi, \mathbf{X}^{\mathbf{e}}_{t,n}, y) + \eta \left\|\mathbf{X}^{\mathbf{e}}_{t,n} - \mathbf{X}^{\mathbf{b}}_{t,n}\right\|_1 + \left\|\mathbf{X}^{\mathbf{e}}_{t,n} - \mathbf{X}^{\mathbf{b}}_{t,n}\right\|_2^2, \tag{5}$$

where c is the loss term weighting, η controls the contribution of the ℓ_1-norm promoting sparsity in the perturbation, and the ℓ_2-norm ensures the overall perturbation remains small.

III. WNNs Architecture

This section presents the design of our WNN-based detector. We highlight the transference learning strategy used to progressively train on increasing graph sizes as well as the practical implementation of the detector, combining spatial and temporal feature extraction for robust detection.

A. Transference Learning

We employ learning by transference, an iterative training strategy that begins with small-scale graphs and progressively expands their size while preserving learned parameters. Initially, the model is trained on a small subset of the power system graph, capturing fundamental structural and feature relationships. As training progresses, the graph expands by introducing new nodes and edges at each epoch, reflecting the evolving system conditions. Throughout this process, the model's gradient updates remain aligned with the true gradient of the WNN learning problem, ensuring that the learned representations remain consistent across different graph sizes. Since the graphs are sampled from the same underlying graphon, they share structural properties. As a result, WNNs can transfer learned parameters across topologies without performance degradation. Unlike GNNs, which suffer from mismatched spectral properties when applied to different graphs, it has been mathematically proven that WNNs guarantee convergence and maintain accuracy as graph size grows [28].

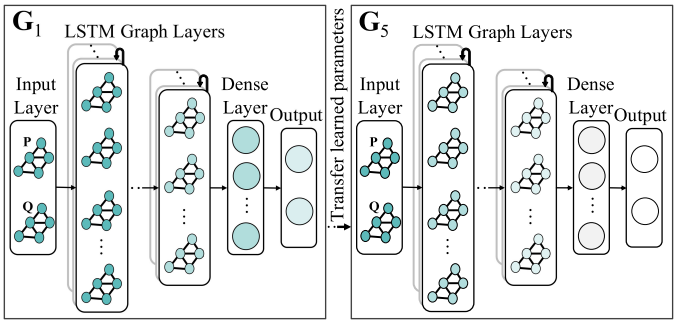

Fig. 1. Illustration of the proposed WNN-based detector.

B. WNN-Based Detector Design

To implement the transference learning approach in practice, we initiate training on smaller graph sizes using a GNN-LSTM model as shown in Fig. 1. The graph convolutional layers extract spatial features from each graph G_n, and the LSTM model captures temporal dependencies from active and reactive power measurements $[P_i, Q_i] \in \mathbb{R}^{n\times 2}$. After spatial and temporal features are processed, the output is passed through fully connected dense layers, which refine the learned representations before producing the final prediction through an output layer. Training begins on smaller graph instances, and after each learning phase, the learned parameters are transferred to the next, larger graph in the sequence. This progressive expansion allows the model to adapt to configuration changes in the power grid while significantly reducing computational overhead and improving efficiency. As a result, the model maintains high detection performance across dynamic topologies.

IV. Experimental Results

This section presents the experimental setup, evaluation metrics, and optimal hyperparameters, followed by a comparison of the proposed WNN model with benchmarks in terms of detection accuracy and efficiency.

A. Detection Performance Metrics

To evaluate the performance of attack detection, we report classification metrics detection rate DR $= \frac{\text{TP}}{\text{TP+FN}}$, F1-score $= \frac{2\cdot\text{TP}}{2\cdot\text{TP+FP+FN}}$, and false alarm rate FAR $= \frac{\text{FP}}{\text{FP+TN}}$, where TP, FN, FP, and TN represent true positives, false negatives, false positives, and true negatives, respectively.

B. Model Setup

We evaluate the WNN-based detector and benchmarks using a unified setup, dataset, attack scenarios, and sequential grid-search for hyperparameter tuning. For benchmarks, support vector machine (SVM) utilizes a Sigmoid kernel and automatic gamma selection. DTs employ the entropy criterion with the best splitter strategy. FNN utilizes 4 layers, 64 units, 0.2 dropout rate, Adam optimizer, and ReLU activation. LSTM utilizes 3 layers, 32 units, no dropout, SGD optimizer, and ReLU activation. GNN and WNN utilize 6 layers, 32 units, 0.4 dropout rate, Adam optimizer, and ReLU activation function.

TABLE I
DETECTION PERFORMANCE OF CLASSIC FDIAS (%)

Model	Metric	Topology				
		1	2	3	4	5
DTs	DR	74.4	75.6	77.8	78.9	80.0
	F1	74.1	74.9	77.0	78.3	79.6
	FAR	31.3	30.5	29.7	28.8	27.2
SVM	DR	75.9	77.8	78.3	80.1	81.0
	F1	75.0	77.1	77.9	79.7	80.7
	FAR	28	27.2	26.3	25.2	24.5
FNN	DR	78.1	79.4	81.4	82	83.7
	F1	77.7	78.9	80.7	81.6	83.0
	FAR	25.2	24.4	23.5	22.1	21.2
RNN	DR	80.2	81.7	83	84.5	85.4
	F1	79.7	81.2	82.8	83.7	84.7
	FAR	23.4	22.5	21.7	20.2	19
GNN	DR	86.3	89	91.8	93.1	94.1
	F1	85.8	88.7	91.6	93.3	94.1
	FAR	18.4	15.3	12.4	7.1	5.5
WNN	DR	86.3	90.4	93.1	96.7	98.2
	F1	86.2	90.3	92.8	96.6	98.1
	FAR	18.4	13.4	8.5	4.8	1.8

TABLE II
DETECTION PERFORMANCE OF EVASION ATTACKS (%).

Model	Metric	Topology				
		1	2	3	4	5
DTs	DR	60.6	61.7	63.4	64.3	66.4
	F1	60.1	61.0	62.8	63.9	65.8
	FAR	44.2	43.5	42.4	41.2	40.6
SVM	DR	62.7	64.9	65.6	67.6	68.9
	F1	62.1	64.2	65.2	67.2	68.3
	FAR	40.1	39.1	38.2	37.4	36.4
FNN	DR	67.5	68.4	70.4	71.2	72.4
	F1	66.9	67.8	69.7	70.8	71.8
	FAR	35.7	34.5	33.6	32.5	31.5
RNN	DR	70.8	71.7	73.7	74.8	75.9
	F1	70.1	71.2	73.1	74.2	75.2
	FAR	32.3	31.3	30.5	29.7	28.4
GNN	DR	84.4	86.5	88.8	89.7	90.8
	F1	84.0	81.0	84.1	85.6	86.8
	FAR	20.1	22.5	19.6	14.3	12.7
WNN	DR	84.4	88.7	91.5	94.9	96.5
	F1	84.0	88.2	90.9	94.4	96.0
	FAR	20.1	15.0	10.2	6.4	3.4

C. Detection Performance

Our experimental results, presented in Tables I and II, demonstrate that the WNN outperforms benchmark models in evasion attack detection. The WNN deteriorates by $1.6-1.9\%$ in DR across the 5 topologies compared to benchmark models which decline in DR by $9.78-10.2\%$ on average. This robustness is attributed to the ability of the WNN-based detector to generalize across structural variations using graphon-based representations, allowing it to maintain high accuracy with unseen topologies under evasion attacks.

1) Classic FDIA: Table I presents the detection performance of all models against classic FDIAs across 5 training topologies. The proposed WNN model outperforms shallow models, deep models, and graph models by $10.4-18.2\%$, $6.1-14.7\%$, and $1.3-4.1\%$ in DR, respectively. The WNN model also outperforms the shallow models, deep models, and graph model by $9.6-25.4\%$, $5.0-19.4\%$, and $1.9-3.9\%$ in FAR, respectively.

2) Evasion Attacks: Table II presents the detection performance of all models against evasion attacks across 5 training topologies. The proposed WNN model outperforms shallow models, deep models, and graph model by $21.7-30.6\%$, $13.6-24.1\%$, and $2.2-5.7\%$ in DR, respectively. The WNN model also outperforms the shallow models, deep models, and graph model by $20.0-37.2\%$, $12.2-28.1\%$, and $7.5-9.4\%$ in FAR, respectively.

3) Impact of Evasion Attacks vs. Classic FDIAs: Fig. 2 illustrates the performance deterioration in DR from classic FDIAs to evasion attacks across all models. The proposed WNN outperforms shallow models, deep models, and graph model by $13.2-13.8\%$, $9.4-10.6\%$, and 1.9%, for topology 1, respectively. The WNN model also outperforms the shallow models, deep models, and graph model by $12.1-13.6\%$, $9.5-11.3\%$, and 3.3%, for topology 5, respectively. The proposed WNN model has lower deterioration due to its generalization ability and scalability over multiple topologies, due to its parameter transfer learning ability.

4) Training and Decision Time: Figs. 3 and 4 compare model complexity in terms of training time and decision time. The shallow models outperform the proposed WNN model in terms of training time by $2.4-2.8$ hours (hrs), however, they perform lower in terms of DR and FAR compared to the proposed WNN model. The proposed WNN model outperforms the deep models and graph model by $1.7-2.4$ hrs and 4.7 hrs, respectively. Moreover, the WNN model outperforms the shallow models, deep models, and graph model by $1.2-1.5$ milliseconds (ms), 1.9 ms, and 1.8 ms, respectively.

V. CONCLUSIONS

This paper investigated the implementation of our WNN-based model for detecting evasion attacks in dynamic power systems. Experimental results showed that the WNN-based model maintained robust detection performance under evasion attacks, with only a $1-2\%$ deterioration in detection rate, significantly outperforming benchmark models, which exhibited a $9-10\%$ deterioration. Additionally, the WNN model enhanced decision-making time by 116% and reduced training time by 133%, offering a scalable and efficient solution for real-time power grid monitoring. These findings highlight the potential of graphon-based learning in addressing the challenges of adversarial robustness and system adaptability.

REFERENCES

[1] M. R. Uddin *et al.*, "False data injection attack detection in edge-based smart metering networks with federated learning," in *IEEE Consumer Communications and Networking Conference*. Las Vegas, NV, USA, 10 –13 Jan. 2025, pp. 1–6.

[2] A. Bamigbade *et al.*, "Cyberattack on phase-locked loops in inverter-based energy resources," *IEEE Transactions on Smart Grid*, vol. 15, no. 1, pp. 821–833, Jan. 2024.

[3] A. Takiddin *et al.*, "Spatio-temporal graph-based generation and detection of adversarial false data injection evasion attacks in smart grids," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 12, pp. 6601–6616, Dec. 2024.

[4] A. T. El-Toukhy *et al.*, "Evasion attacks in smart power grids: A deep reinforcement learning approach," in *IEEE Consumer Communications and Networking Conference*. Las Vegas, NV, USA, 6 – 9 Jan. 2024, pp. 708–713.

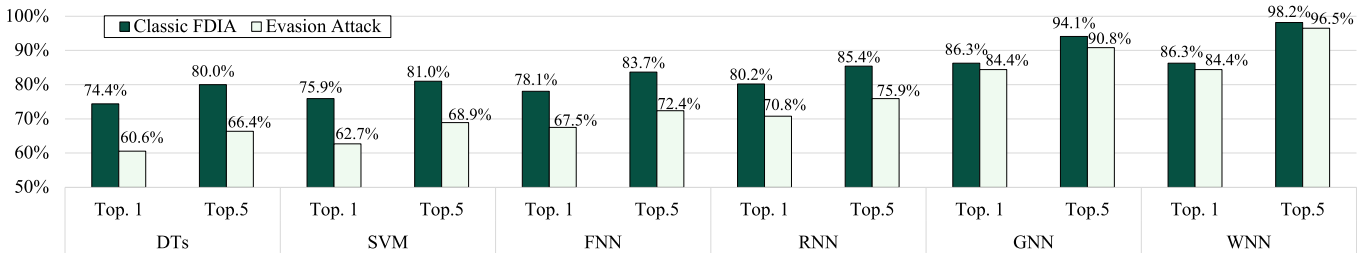

Fig. 2. Model DR against classic FDIAs and evasion attacks.

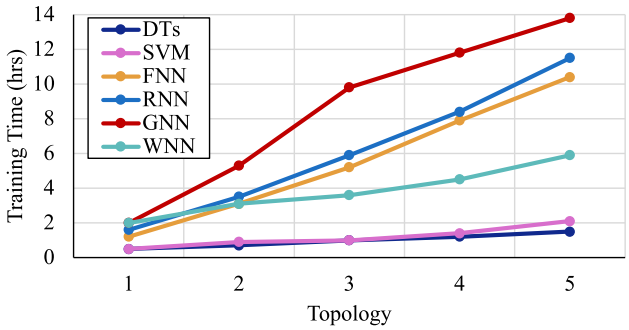

Fig. 3. Training time comparison between WNN and benchmarks.

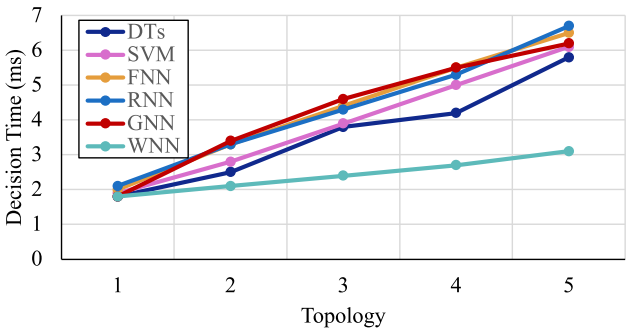

Fig. 4. Decision time comparison between WNN and benchmarks.

[5] X. Lu *et al.*, "False data injection attack location detection based on classification method in smart grid," in *International Conference on Artificial Intelligence and Advanced Manufacturing*. Manchester, United Kingdom, 15–17 Oct. 2020, pp. 133–136.

[6] D. Wang *et al.*, "Detection of power grid disturbances and cyber-attacks based on machine learning," *Journal of Information Security and Applications*, vol. 46, pp. 42–52, June 2019.

[7] A. Bhattacharjee *et al.*, "Deep latent space clustering for detection of stealthy false data injection attacks against ac state estimation in power systems," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2338–2351, May 2023.

[8] X. Su *et al.*, "Damgat-based interpretable detection of false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 15, no. 4, pp. 4182–4195, July 2024.

[9] A. Takiddin *et al.*, "Robust graph autoencoder-based detection of false data injection attacks against data poisoning in smart grids," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 3, pp. 1287–1301, March 2024.

[10] W. Xia *et al.*, "Locational detection of false data injection attacks in the edge space via hodge graph neural network for smart grids," *IEEE Transactions on Smart Grid*, vol. 15, no. 5, pp. 5102–5114, Sep. 2024.

[11] A. Takiddin *et al.*, "Generalized graph neural network-based detection of false data injection attacks in smart grids," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 7, no. 3, pp. 618–630, June 2023.

[12] R. Atat *et al.*, "Graphon neural networks-based detection of false data injection attacks in dynamic spatio-temporal power systems," *IEEE Open Access Journal of Power and Energy*, vol. 12, pp. 24–35, Jan. 2025.

[13] J. Sweeten and A. Elshazly, "Cyber-physical fusion for gnn-based attack detection in smart power grids," *IEEE Open Access Journal of Power and Energy*, vol. 12, pp. 515–528, July 2025.

[14] H. Keller *et al.*, "Multi-task graph-based attack detection and localization in cyber-physical power systems," in *Proceedings of the European Signal Processing Conference*. Palermo, Italy, 2–6 Sep. 2025, pp. 1752–1756.

[15] A. Takiddin and M. Ismail, "Robust detection of electricity theft against evasion attacks in smart grids," in *IEEE International Conference on Communications*. Montreal, QC, Canada, 14–23 June 2021, pp. 1–6.

[16] I. Elgarhy *et al.*, "Securing smart grid false data detectors against white-box evasion attacks without sacrificing accuracy," *IEEE Internet of Things Journal*, vol. 11, no. 20, pp. 33 873–33 889, Oct. 2024.

[17] L. Tong *et al.*, "Adversarial sample detection framework based on autoencoder," in *International Conference on Big Data, Artificial Intelligence Software Engineering (ICBASE)*, Chengdu, China, 30 Oct.–1 Nov. 2020, pp. 241–245.

[18] P.-Y. Chen *et al.*, "Ead: Elastic-net attacks to deep neural networks via adversarial examples," in *AAAI Conference on Artificial Intelligence*, New Orleans, USA, 2 – 7 Feb. 2018, pp. 10–17.

[19] J. Eldridge *et al.*, "Graphons, mergeons, and so on!" *Advances in Neural Information Processing Systems*, vol. 29, Dec. 2016.

[20] L. Ruiz *et al.*, "The graphon fourier transform," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4 – 8 May 2020, pp. 5660–5664.

[21] L. V. Kantorovich and G. P. Akilov, *Functional Analysis*, 2nd ed. Elsevier, Feb. 2016.

[22] S. Chatterjee, "Matrix estimation by universal singular value thresholding," *The Annals of Statistics*, vol. 43, no. 1, pp. 177–214, Feb. 2015.

[23] E. Airoldi *et al.*, "Stochastic blockmodel approximation of a graphon: Theory and consistent estimation," *Advances in Neural Information Processing Systems*, Nov. 2013.

[24] S. Chan and E. Airoldi, "A consistent histogram estimator for exchangeable graph models," in *International Conference on Machine Learning*, 21 – 26 June 2014, pp. 208–216.

[25] "Electric Reliability Council of Texas." [Online]. Available: https://www.ercot.com/mktinfo/loadprofile/alp

[26] R. D. Zimmerman *et al.*, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, Feb. 2011.

[27] A. Madry *et al.*, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 30 Apr.–3 May 2018.

[28] L. Ruiz *et al.*, "Graphon neural networks and the transferability of graph neural networks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1702–1712, Dec. 2020.