

PowerBench: A Benchmark Suite for Monitoring and Security in Distribution Systems

Md Joshem Uddin*

MDJOSHEM.UDDIN@UTDALLAS.EDU

Dept. of Mathematical Sciences

University of Texas at Dallas

Richardson, Texas, USA

Roshni Anna Jacob*

ROSHNI.JACOB@UTDALLAS.EDU

Dept. of Mechanical Engineering

University of Texas at Dallas

Richardson, Texas, USA

Damilola R. Olojede

DAMILOLA.OLOJEDE@UTDALLAS.EDU

Dept. of Mechanical Engineering

University of Texas at Dallas

Richardson, Texas, USA

Nanhong Liu

NANHONG.LIU@UTDALLAS.EDU

Dept. of Mechanical Engineering

University of Texas at Dallas

Richardson, Texas, USA

Baris Coskunuzer†

COSKUNUZ@UTDALLAS.EDU

Dept. of Mathematical Sciences

University of Texas at Dallas

Richardson, Texas, USA

Jie Zhang†

JIEZHANG@UTDALLAS.EDU

Dept. of Mechanical Engineering;

Dept. of Electrical and Computer Engineering

University of Texas at Dallas

Richardson, Texas, USA

Reviewed on OpenReview: <https://openreview.net/forum?id=CHMiQ3aw9q>

Editor: Yi Zhou

Abstract

Modern power distribution networks face growing risks from physical faults and cyber intrusions, demanding robust data-driven monitoring for operational resilience. Yet, progress in this area is constrained by the lack of standardized, openly accessible datasets that reflect realistic grid behaviors. This paper presents PowerBench, a comprehensive benchmark suite designed to advance machine learning research for distribution network monitoring and security. PowerBench provides high-fidelity simulations of IEEE 34-, 123-, and 8500-bus feeders under diverse operating and attack scenarios, annotated with line status, cyber events,

*. Co-first authors.

†. Co-senior authors.

and system states. Each dataset includes fixed training–validation–test splits, task-specific metrics, and baseline implementations for outage detection and localization, cyberattack identification, and state estimation. Comparative experiments with representative graph neural networks and graph transformer models are reported to establish performance baselines. All datasets, simulation scripts, and evaluation tools are released as open-source resources to ensure transparency and facilitate community adoption. By unifying multiple diagnostic and estimation tasks within a realistic framework, PowerBench provides a common foundation for developing and benchmarking intelligent, trustworthy methods for smart grid applications.

Keywords: power grid, resilience, distribution networks, datasets, outage detection, state estimation, cyberattacks, and graph learning.

1 Introduction

Modern power grids have evolved into highly complex, interconnected cyber-physical systems that form the backbone of societal and economic activity (Amani and Jalili, 2021). However, it has increasingly become vulnerable to component failures and operational disruptions, driven by the rising frequency of extreme weather events and the escalating threat of cyber attacks (Younesi et al., 2022). Accordingly, enhancing grid resilience and robust security monitoring are crucial.

Distribution networks (DNs) have traditionally been regarded as passive infrastructures, characterized by limited observability due to sparse sensing and minimal high-fidelity monitoring (Majidi et al., 2014). Moreover, widespread deployment of meters across the DN introduces significant data traffic and communication bottlenecks. With the increasing penetration of distributed energy resources (DERs), smart inverters, and electric vehicles (EVs), DNs are being transformed into active, bidirectional cyber-physical infrastructures. This evolution necessitates real-time situational awareness, advanced state estimation, and threat detection strategies considering both cyber and physical uncertainties (Qi et al., 2016).

Traditional DN monitoring relies on model-based state estimation (SE) (e.g., weighted least squares), which assumes accurate network parameters and sufficiently dense measurements. In modern active DNs, DERs and EVs introduce bidirectional flows, nonlinear and unbalanced operation, and stochastic variability, often degrading SE accuracy (Primadianto and Lu, 2016; Dehghanpour et al., 2018; El Iaali et al., 2025). Moreover, SE is poorly suited to sparsely metered feeders and can be computationally intensive, initialization-sensitive, and difficult to scale for real-time monitoring (Primadianto and Lu, 2016; Dehghanpour et al., 2018). Finally, classical SE frameworks are not designed to withstand cyber-physical threats or detect stealthy data-driven attacks that exploit model dependencies (Qi et al., 2016).

Machine learning, especially graph learning, can complement model-based DN monitoring by learning nonlinear dependencies in network-structured data and enabling fast inference under partial observability (Akhtar et al., 2023; Jacob et al., 2024; Chen et al., 2023). Despite growing interest (El Iaali et al., 2025), progress is hindered by the lack of standardized, physics-informed datasets that capture diverse operating conditions and adversarial/failure scenarios. Real utility data are difficult to obtain due to privacy and limited accessibility (Altamimi et al., 2024), and outages or cyberattacks are rare events that yield scarce labeled examples (Fahim et al., 2024). Consequently, building reliable and reproducible benchmarks for resilience-oriented learning models remains challenging.

Given these limitations, it is standard practice in the power systems community to employ synthetic or benchmark test networks for developing and evaluating new algorithms. Such networks (e.g., IEEE 34-, 123-, and 8500-node systems) provide controlled, reproducible environments for validating analytical and learning-based approaches. As highlighted in (Altamimi et al., 2024), synthetic datasets play a crucial role in advancing smart grid analytics by enabling systematic evaluation, reproducibility, benchmarking, and broader accessibility for both experts and non-experts in the domain. Several recent datasets have advanced ML for power system applications, but none specifically target the security and monitoring of distribution networks to support resilience and reliability through unified graph-based tasks. For example, PowerGraph (Varbella et al., 2024) introduces a graph neural network (GNN) tailored benchmark suite for power flow, optimal power flow, and cascading failure analysis, but focuses on transmission grids. Similarly, SafePowerGraph (Ghamizi et al., 2024) focuses on safety-aware evaluation of GNNs for power flow and optimal power flow problems under line outage and price variation scenarios. While both datasets enable the emulation of traditional power flow (PF) and optimal power flow (OPF) solvers, they do not support situational awareness tasks that are critical for enhancing grid resilience and operational reliability. Nauck et al. (Nauck et al., 2023) introduce datasets simulated for the dynamic stability assessment of synthetic power grids using GNNs, focusing on the inertia and synchronization dynamics of coupled oscillatory systems. Although an important step toward data-driven stability analysis, this work does not address DN resilience problems. Another work, GridTracer (Huang et al., 2021), introduces an overhead imagery dataset for tower detection and power line connectivity mapping, enabling graph-based reconstruction of grid infrastructure from remote sensing data. However, it is primarily designed for infrastructure mapping and visualization, rather than for operational monitoring or power system analysis.

While these resources have advanced ML research for transmission systems and infrastructure mapping, none systematically addresses distribution-level challenges such as line outage detection, cyberattack identification, or state estimation under partial observability. PowerBench fills this gap by transforming standard IEEE feeder models (34-bus, 123-bus, and 8500-node) (Kersting, 2001; Arritt and Dugan, 2010) into graph-structured datasets, annotated with line status, adversarial events, and operating conditions for key DN tasks, enabling systematic evaluation of learning models under realistic operational scenarios. Each dataset supports supervised and semi-supervised learning with predefined data splits and standard evaluation metrics. Performance baselines are established through the evaluation of models, spanning from classical graph neural networks (Wu et al., 2020) to modern graph transformers (Ying et al., 2021; Yuan et al., 2025), and the strengths and limitations of each approach are highlighted.

Our key contributions are as follows:

- *Diverse, multi-scale and realistic dataset generation:* Over two million labeled snapshots encompassing diverse load/DER profiles, single- and multi-phase faults, varied cyberattack strategies, and heterogeneous operating conditions on different scale distribution networks.
- *Baseline benchmarking:* A comprehensive evaluation of recent graph neural networks, graph transformers, and topology-aware models, supported by task-specific data splits, standardized metrics, and reproducible protocols.

- *Open-source, extensible framework and datasets:* Publicly released datasets and code-base for data generation, visualization, and model evaluation, enabling easy replication, benchmarking, and extension to new feeders, scenarios, and tasks.

By unifying these tasks in one framework *PowerBench* provides a solid foundation for developing and comparing reliable machine learning models that improve grid resilience, an area of growing importance.

2 PowerBench Overview

PowerBench supports a wide range of learning tasks central to DN monitoring, fault detection, and state estimation:

- *Line outage detection and localization:* This task enables the detection and classification of outages caused by line failures, one of the most common disruptions in power systems (Campbell and Lowry, 2012). Outages may stem from weather, equipment faults, or vegetation interference. Under sparse sensing (about 40% sensor coverage), the task involves binary classification (no outage vs. at least one outage) and can be extended to edge-level fault localization or identifying unsupplied loads. These formulations cast as edge or node classification tasks, can be especially challenging under partial observability (Yuan et al., 2020). In this work, we address the fault localization problem by adopting a region-level localization strategy by considering electrically and geographically coherent operational zones in the feeder. This formulation aligns with real-world utility practice, where feeders are divided into protection zones and coordination sections. Each region is assigned a binary label indicating whether any line failure within that zone has failed, and multiple regions may be simultaneously affected.
- *Cyberattack detection and localization:* Three categories of cyber-physical attack scenarios are modeled. Each of these datasets supports graph-level anomaly detection, localization of compromised components, and time-series anomaly detection.
 - Voltage sensor attacks involve false data injection attacks (FDIAs) that alter reported voltage measurements without any physical anomaly (Liang et al., 2016). Though subtle, such attacks affect spatial and temporal signal patterns that ML models can learn to detect (Olojede et al., 2024).
 - PV attacks alter control settings or output readings, misleading operators about real power injections and causing voltage violations (Ye et al., 2021).
 - EV charging stations (EVCS) attacks involve adversarial manipulation of electric vehicle charging profiles, which can mislead control actions or destabilize the grid, particularly during peak hours (Jahangir et al., 2024; Acharya et al., 2024).
- *State estimation:* This task focuses on estimating unmeasured system states, especially bus voltages, from limited sensor data. Such inference is critical for reactive power control, voltage regulation, and grid stability, particularly when only a subset of buses and lines are instrumented (Dahale et al., 2020; Primadianto and Lu, 2016). Tasks include node-level voltage regression and graph completion under partial observations.

Each dataset models the distribution feeder as a graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes ($\mathcal{V}$) represent buses, and edges ($\mathcal{E}$) represent lines or inline transformers. Nodes are annotated with operational roles (e.g., substation, load, generator), and edges include physical meta-data. Features include node voltages, power flows, and derived electrical quantities, with scenarios generated through realistic power flow simulations. Figure 1 illustrates how physical grid components, such as buses, lines, and PV units, are mapped to graph elements and associated node/edge features in PowerBench.

The dataset includes both static snapshots (single network states) and time-series sequences (multi-step trajectories). Each instance is task-labeled, for example, for outages, attacks, or voltage targets, and is accompanied by preprocessing routines, consistent graph representations, and data-loading utilities designed for integration with standard GNN toolkits.

In terms of scalability and diversity, PowerBench encompasses multiple dimensions:

- Feeder variety (IEEE 34-bus, 123-bus, and 8500-node networks with diverse topologies, voltage levels, and DER configurations)
- Operational diversity (normal operation, variable load and DER profiles, and multiple fault types)
- Adversarial settings (voltage sensor false data injection attacks, PV curtailment spoofing, and EV charging manipulation)
- Sensing regimes (full and partial observability under varying sensor placements)
- Static/temporal modeling (snapshot or time-series data supporting event progression analysis).

This diversity supports the evaluation of ML models on generalization and scalability under realistic and challenging conditions. The benchmark is modular and can be extended with new feeders, attacks, and tasks as needed.

3 Dataset Generation and Annotations

The datasets in PowerBench are synthetic datasets generated from simulations of power distribution network operations under diverse scenarios, both with and without events, where

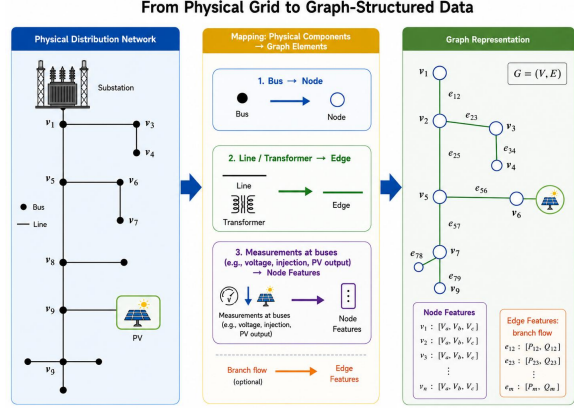


Figure 1: Illustration of how physical distribution network components are mapped to graph-structured data in PowerBench. Buses are represented as nodes, lines or inline transformers as edges, and electrical measurements such as bus voltages, PV outputs, and branch flows are incorporated as node and edge features.

Table 1: **PowerBench dataset overview.** We list dataset size $|\mathcal{G}|$, graph scale $|\mathcal{V}|, |\mathcal{E}|$, temporal length $|t|$, node feature dimension, number of tasks, and the associated learning setup and label type for each benchmark.

Setting	Problem Type	Dataset	$ \mathcal{G} $	$ \mathcal{V} $	$ \mathcal{E} $	$ t $	Node Feat.	#Tasks	Learning Task	Classification Type
Static	Line Failure Detection	34-Bus	10000	37	36	-	3	1	Graph Classification	Binary
		123-Bus	15000	132	131	-	3	1	Graph Classification	Binary
		8500-Node	20000	4876	4874	-	3	1	Graph Classification	Binary
	Outage Localization	34-Bus	10000	37	36	-	3	6	Graph Classification	Multi-label
		123-Bus	5000	132	131	-	3	8	Graph Classification	Multi-label
	State Estimation	34-Bus	10000	37	36	-	3	-	Graph Regression	-
		123-Bus	15000	132	131	-	3	-	Graph Regression	-
		8500-Node	20000	4876	4874	-	3	-	Graph Regression	-
	Time Series	PV Attack Detection/Local.	34-Bus	2000	37	36	48	3	1/3	Temporal Graph Class.
123-Bus			4000	132	131	48	3	1/5	Temporal Graph Class.	Binary/Multi-label
8500-Node			2000	4876	4874	48	3	1/7	Temporal Graph Class.	Binary/Multi-label
EV Attack Detection		34-Bus	2000	37	36	288	3	1/3	Temporal Graph Class.	Binary/Multi-label
		123-Bus	4000	132	131	288	3	1/5	Temporal Graph Class.	Binary/Multi-label
		8500-Node	1000	4876	4874	288	3	1/7	Temporal Graph Class.	Binary/Multi-label
Sensor Attack Detection		34-Bus	2000	37	36	48	3	1/37	Temporal Graph Class.	Binary/Multi-label
		123-Bus	4000	132	131	48	3	1/132	Temporal Graph Class.	Binary/Multi-label
		8500-Node	4000	4876	4874	48	3	1/-	Temporal Graph Class.	Binary/Multi-label

system states are extracted after solving the network power flow. The state describes the operating conditions of the network using variables such as bus voltages, branch power flows, and power injections at buses, considering load and generation.

3.1 Simulation Environment

To construct the datasets in PowerBench, open-source distribution system simulator OpenDSS (Dugan and McDermott, 2011) interfaced through OpenDSSDirect.py (Krishnamurthy and Meira, 2024), a Python API enabling direct circuit manipulation and I/O operations. OpenDSS scripts for standard power distribution test networks, namely the IEEE 34-bus, 123-bus, and 8500-node test networks (Kersting, 2001; Arritt and Dugan, 2010), are employed with circuit manipulations performed as required for the specific tasks.

Utility-scale photovoltaic (PV) systems are modeled by integrating the PVSystem element (in OpenDSS) to the standard IEEE test networks. This element models the combined behavior of the PV array and its associated inverter. We selected deployment at 3, 5, and 7 locations for the IEEE 34-bus, 123-bus, and 8500-bus networks, respectively. The irradiance input is based on 5-minute resolution Global Horizontal Irradiance (GHI) data obtained from the National Solar Radiation Database (NSRDB) (National Renewable Energy Laboratory (NREL), 2020). Further, to capture seasonal and temporal demand variability, annual representative load profiles with hourly resolution are utilized within OpenDSS. The load shape is applied to all network loads using time-dependent multipliers on the base demand, effectively simulating realistic operating conditions across different times of the

year. For cases with EVCS, the storage element model available in OpenDSS is configured with charging-only mode. Time-varying charging profiles, derived from real-world EVCS usage data in Texas, are used as the power input to these elements. This setup allows for a realistic simulation of spatial and temporal EV charging demand.

The simulation environment is set up to perform quasi-static time series power flow analysis by iterating over time steps (e.g., 5 or 30 minutes) for the time series tasks and to run in snapshot mode for static tasks. For each state update, the power flow is solved, and the resulting system variables are retrieved and stored as structured data. The entire pipeline embeds OpenDSS within Python to inject events (e.g., line outages, cyberattacks) and generate labeled data. The network interconnectivity is also abstracted as a NetworkX graph, where buses are treated as nodes and branches as edges. Larger datasets (e.g., 123-bus and 8500-node) are generated using high-performance computing resources (High Performance Computing at UTD). The simulation set up and the scenario generation modes are illustrated in Figure 2.

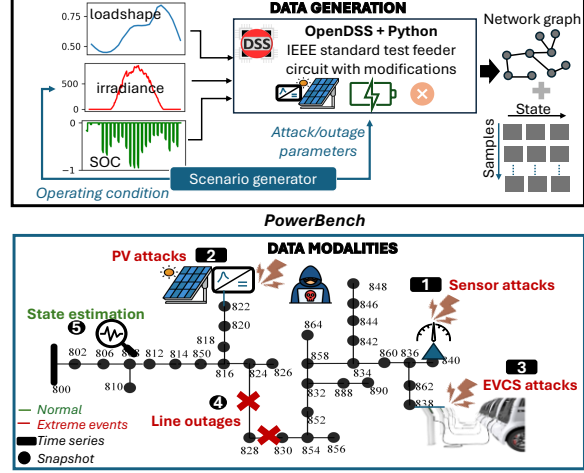


Figure 2: Illustration of data generation setup and the data modalities considered in the power distribution network.

3.2 Event and Scenario Generation

The datasets in PowerBench, summarized in Table 1, are designed for specific tasks as detailed below. The feeder scales considered here are based on standard IEEE distribution network benchmarks commonly used in power systems practice, namely the 34-bus, 123-bus, and 8500-node feeders. The number of scenarios for each task is selected to ensure sufficient diversity of operating conditions and event cases, while also providing adequate coverage of the larger output spaces that arise in larger networks and localization settings. Temporal resolution is also taken into account when determining the number of scenarios. In particular, the EV attack dataset uses a 5-minute temporal resolution because EV charging demand and attack-induced perturbations can evolve over short time intervals; using a coarser resolution could smooth out these effects and make them harder to capture. Since this choice produces longer sequences and a substantially larger data volume, we generate fewer scenarios for this dataset to keep its size practical. Overall, this design supports practical scalability analysis across increasing graph size, task complexity, and dataset diversity.

3.2.1 CYBER ATTACK DATASETS

Cyberattacks are modeled on three key components within the distribution network: (a) voltage sensors, (b) PV control systems, and (c) electric vehicle charging stations (EVCS). These attacks represent adversarial manipulations at the sensing, control, or cyber-physical layer, aiming to mislead grid monitoring systems and induce operational disruptions due

to compromised situational awareness. The construction of the cyber attack datasets is as follows.

(a) *Sensor Attack Dataset*: The sensor attack dataset built on standard IEEE test feeders (34-bus, 123-bus, and 8500-node) includes both normal and anomalous operational scenarios to support the development of FDIA detection methods for voltage sensors under realistic grid conditions. Scenarios are constructed by performing quasi-static time series power flow simulations in OpenDSS using the daily mode with a 30-minute resolution. For each simulation, a daily load shape is randomly selected from the collection of representative annual profiles. At each time step, bus voltage magnitudes and branch power flows are recorded. Normal operation scenarios include: (i) baseline cases with no perturbations, and (ii) cases with abrupt demand surges or curtailments, implemented by scaling segments of the load shape. These simulate physical under- and over-voltage events due to real load anomalies, thereby adding diversity to the dataset by including non-attack but anomalous conditions. Sensor attack scenarios, on the other hand, are created by injecting manipulated voltage values after power flow simulation at selected buses. These manipulations do not alter the actual network state or power flow but only affect reported sensor measurements, emulating real-world FDIA scenarios. Each scenario (normal or attacked) is stored as a dictionary containing scenario identifier and associated attack metadata which includes: a label indicating whether the scenario is anomalous (Yes/No), attack type (under voltage, over voltage, or none), buses targeted for attacks (if any), time series voltages at network buses and power flows through branches.

(b) *PV Attack Dataset*: The dataset is generated using the standard distribution test feeders modified with PV systems integrated at selected buses. Time series simulations are conducted in OpenDSS, leveraging realistic load profiles from representative annual data and irradiance profiles from the National Solar Radiation Database (NSRDB) with 1-hour and 5-minute resolutions, randomly sampled per scenario to provide robustness across changing loading conditions and seasonal variations. Each scenario represents a time series simulation with 30-minute resolution, with system states updated at each step to record bus voltages, branch power flows, and PV active power injections. Normal scenarios capture variations in load and generation without any malicious interference. In contrast, the attack scenarios fall into one of the following categories: (i) Power curtailment attack (PCA) where the PV output is reduced by modifying the active power setpoints; (ii) Power factor attack (PFA) where the PV inverter's power factor is altered, resulting in abnormal reactive power injections and voltage deviations; (iii) Denial-of-service (DoS) attacks where the PV output is suppressed simulating a shutdown. For each attack scenario, the targeted PV units, the attack start time, and the duration of the attack are selected randomly to introduce variability across instances. Each scenario is stored as a structured dictionary containing anomaly metadata, which includes a label indicating whether the scenario is anomalous (Yes/No), attack type (PCA, PFA, DOS, or None), targeted PVs (if any), and time series information such as bus voltage magnitudes, branch power flows, and actual PV power outputs.

(c) *Charging Station Attack Dataset*: The dataset is generated using the standard test feeders modified with charging stations at selected buses. Daily variations in EVCS usage are modeled using hourly-scaled demand shapes with a 1-second resolution. A quasi-static daily time series simulation is performed in OpenDSS using a 5-minute resolution. At each time step, bus voltage magnitudes, branch power flows, and EVCS active power consumptions are

recorded. The dataset includes both normal and anomalous scenarios. The normal scenarios simulate typical load demand and charging activity using varying load and storage shapes. The anomalous scenarios involve four different types of attacks: (i) type 1 with increased charging during peak hours resulting in increased network stress, (ii) type 2 with time shifted charging demand to create forecasting errors, (iii) type 3 with a combination of demand increase and time shift, (iv) type 4 with peak demand reshaped with additional time-shifted surge. For each anomalous instance, the attacked EVCS units, the attack start time, and the attack duration are randomly sampled to ensure scenario diversity. Each scenario (normal or under EVCS attack) is stored as a structured dictionary including an anomaly label (yes/no), attack type, targeted stations, time series of bus voltage magnitudes, branch power flows, and EVCS active power consumption.

3.2.2 LINE FAILURE (OUTAGE) DATASET

The dataset is generated using standard IEEE distribution test feeders with predefined sensor placements at selected buses and branches. Measurements are available at approximately 40% of the buses and branches, including all substation buses and feeder headers. The simulation is conducted in snapshot mode, and normal scenarios are generated without any line failures under varying loading conditions to emulate typical network operations. Outage scenarios, in contrast, are created by randomly disconnecting edges (i.e., lines) within a localized subgraph centered around a randomly selected node (bus). The number of failed lines and the radius of the subgraph are randomized to simulate a range of contingency scales. Additionally, the phases affected by each line failure are randomly selected, introducing further diversity. Each scenario is stored as a structured dictionary containing the outage label (Yes/No), a list of failed lines, the type of failure (i.e., affected phases), voltage magnitudes at sensor-equipped buses, and power flows along sensor-monitored branches.

3.2.3 OUTAGE LOCALIZATION DATASETS

Here, the outage detection task is extended to address fault localization under partial observability. The IEEE 34-bus and 123-bus distribution test feeders are first represented as undirected graphs, with buses modeled as nodes and lines as edges. The graph is then processed to ensure a single connected component containing the substation bus, after which it is oriented into a directed radial tree using a breadth-first traversal. A balanced k -way partitioning algorithm is subsequently applied by recursively splitting the largest component. At each step, the edge whose removal minimizes the imbalance between the upstream and downstream subtrees is selected. This procedure produces approximately equal-sized radial subnetworks, with each edge uniquely assigned to a single partition, ensuring complete feeder coverage without overlap. These subnetworks form the basis for dataset generation and localized analysis. Each subnetwork (region) receives a binary label indicating whether any line failure occurs in that zone, and multiple regions can be affected simultaneously. Thus, we formulate outage localization as a multi label graph classification task.

3.2.4 STATE ESTIMATION DATASET

The dataset is generated using standard IEEE distribution test networks (34-bus, 123-bus, and 8500-node), with system simulations conducted in snapshot mode using OpenDSS. Similar to the line failure detection dataset, sensors are placed at selected buses and branches, covering

approximately 40% of the network. To introduce variability, different loading conditions are sampled and used for solving the power flow. Voltage measurements at sensor-equipped buses and power flows along instrumented branches are recorded, while the voltage magnitudes at unsensed buses serve as targets for the state estimation task.

4 Learning Baselines and Evaluation

PowerBench formalizes a set of reproducible learning tasks that capture key challenges in modern distribution networks, i.e., partial observability, topological disruptions, and cyber-physical threats. We group them into two families: (i) *static tasks* (single-snapshot inference for line failure detection, outage localization, state estimation), and (ii) *temporal tasks* (time-series inference for cyberattack detection, localization). Each task is backed by labeled data, fixed splits, and standard metrics to ensure fair, “plug-and-play” benchmarking of ML models. See Table 1 for feeder sizes, feature dimensions, and snapshot counts.

4.1 Task Formulations in Graph Domain

4.1.1 STATIC TASKS: LINE FAILURE DETECTION, OUTAGE LOCALIZATION AND STATE ESTIMATION

The *line failure detection* problem is formulated as a graph classification task on power distribution networks. Each weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ represents a snapshot of the network state, where $|\mathcal{V}| = N$ is the total number of nodes, and $\mathcal{W} \in \mathbb{R}^{N_o \times 3}$ contains voltage magnitude measurements at the $N_o \subset N$ observable nodes. Node features are only available for these observable nodes, while the remaining nodes have missing features due to unmonitored sensors.

Each graph is associated with a binary label indicating the presence or absence of a line failure. A label of 1 denotes that the graph exhibits at least one line failure (outage), while a label of 0 indicates normal operation without any outage. This setup defines a binary graph classification problem focused on determining whether a failure has occurred in the network. The primary objective of this task is to evaluate the effectiveness of graph-based models in detecting and characterizing line failures using only partially observed voltage data within complex power distribution systems.

In the *outage localization* task, each graph is annotated with a multi-label target vector that indicates whether an outage occurs in each predefined subnetwork (subregion) of the grid. Specifically, a label value of 1 means the corresponding subnetwork is experiencing an outage, whereas 0 denotes normal operation with no outage in that region. Consequently, the problem is formulated as a multi-label graph classification task, where the goal is to accurately identify the outage locations across multiple subnetworks within the network.

The *state estimation* task addresses node feature imputation across multiple graphs. Given $\mathcal{G} = \{\mathcal{G}_1, \mathcal{G}_2, \dots, \mathcal{G}_N\}$, a collection of graphs, where each graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ consists of nodes $\mathcal{V}_i$, edges $\mathcal{E}_i$, and a node feature matrix $\mathcal{X}_i \in \mathbb{R}^{|\mathcal{V}_i| \times d}$, the goal is to infer the missing node features for a fixed subset of nodes in the test graphs. During training, all node features are fully observed. In contrast, the test graphs contain only a fixed subset of nodes with known features (referred to as *sensor nodes*), while the remaining nodes (referred to as *target* or *unknown nodes*) have missing features. These known and unknown node positions remain

consistent across all graphs since it depends on sensor coverage. The objective is to train a model f_θ that takes as input the partially observed node features $\mathcal{X}_i'$, the graph structure $\mathcal{E}_i$, and an indicator mask $M_i \in \{0, 1\}^{|\mathcal{V}_i|}$ denoting which nodes are observed, and then predicts the missing features accordingly.

4.1.2 TEMPORAL TASKS: CYBERATTACK DETECTION AND LOCALIZATION

In time series task, we address the critical challenge of cyberattack detection and localization in PV systems, EV charging infrastructures, and bus-level sensors. The cyberattack detection and localization problem is formulated as a spatio-temporal graph classification task. The power system is modeled as a sequence of graphs over time, where each graph snapshot at time t is represented as $\mathcal{G}_t = (\mathcal{V}, \mathcal{E}, \mathcal{W}_t, e_t)$. Here, $\mathcal{V}$ denotes the set of nodes (e.g., buses, PV buses, EV charging station buses, sensor buses), $\mathcal{E}$ the set of edges (electrical connections), $\mathcal{W}_t$ the node features, and e_t the edge features at time t . Graphs are sampled over 24 hours at fixed intervals, forming a time series that captures system dynamics. Two complementary tasks are defined:

(a) *Attack Detection (Binary Classification)*: Given a sequence of temporal graph snapshots, the goal is to determine whether any cyberattack has occurred within the sequence. Each sequence is labeled as 1 if there is at least one attack on any component (PV, EV, or sensor) during the time window, and 0 otherwise. This task captures the presence of malicious activity in the power grid over time.

(b) *Attack Localization (Multi-Label Classification)*: Beyond detection, we also aim to identify the specific components under attack. For each time-series input, we generate a multi-hot label vector indicating which PV systems, EV charging stations, and bus sensors were compromised at any point in time. If a particular component is attacked during any time step of the sequence, its corresponding label is marked as 1, and 0 otherwise. Our setting allows for multi-point attacks, where multiple components can be simultaneously targeted, reflecting realistic and practical cyberattack scenarios.

4.2 Dataset Splits and Training Configuration

All models are trained using a consistent configuration to ensure fair comparison. Each dataset is split into training, validation, and test sets using a 60%/20%/20% ratio. Every model is trained and evaluated over 10 independent runs to ensure statistical reliability. We use the Adam optimizer for all models, with two layers, 64 hidden channels, a dropout rate of 0.5, and a learning rate of 0.005. For attention-based models such as the Transformer and GPS, we use 2 attention heads to keep computational complexity feasible. The final reported results are averaged over the 10 runs to provide a robust assessment of model performance. All splits, hyperparameters, and random seeds are published in our code release.

4.3 Datasets and Benchmark Availability

All datasets are publicly available at <https://zenodo.org/records/15400564> (line outage and state estimation), <https://zenodo.org/records/15401632> (PV and sensor cyberattacks), <https://zenodo.org/records/15401290> (EVCS cyberattacks), and code including data generation and evaluation tools is hosted on GitHub at <https://github.com/RoshniAnna/PowerBench>. We invite the community to contribute new feeders, attack sce-

narios and learning tasks, and to use PowerBench for research in robustness, generalization and foundation models for physical systems.

4.4 Baseline Models

In static graph classification, we evaluate ten representative graph learning methods covering classic convolutional approaches, attention mechanisms, and transformer-inspired architectures. The graph convolutional network (GCN) (Kipf and Welling, 2017) uses a linear message passing scheme with renormalized adjacency, while GraphSAGE (Hamilton et al., 2017) learns neighborhood aggregators for inductive learning. The graph attention network (GAT) (Veličković et al., 2018) employs self-attention to weight neighbor contributions dynamically, and the graph isomorphism network (GIN) (Xu et al., 2019) achieves high expressive power through learnable sum aggregators. Topology adaptive GCN (TAGCN) (Du et al., 2017) generalizes convolution by combining multi-hop filters, and Graphormer (Ying et al., 2021) adapts the transformer backbone to graphs using structural encodings and centrality biases. UniMP (Shi et al., 2021) unifies node and edge message passing within a single attention framework, and the GPS model (Rampášek et al., 2022) integrates local message passing with global transformer-style attention. We also include two topological baselines: Single persistent MLP (SP-MLP), which applies a multi-layer perceptron (MLP) to single persistence vectors using persistent homology, and multi-persistent MLP (MP-MLP), which applies MLP to multi-persistence vectors derived from either static graphs or temporal snapshots (Olojede et al., 2024). Detailed descriptions of both the single- and multi-persistence constructions are provided in Section C of the appendix.

These methods span a broad range of paradigms and serve as comprehensive references for benchmarking on PowerBench. Furthermore, we apply the four classical GNN architectures, GCN, GraphSAGE, GAT, and GIN, to the state estimation task, due to their versatility across both classification and regression scenarios.

To evaluate the effectiveness of our approach in the spatio-temporal graph classification task, we implement several baseline models that leverage both sequential learning and graph-based methods. These models serve as benchmarks for both the binary attack detection and multi-label attack localization tasks. Although spatio-temporal GNNs, such as Temporal GCN variants (e.g., GCN+LSTM), are a natural choice for this setting because they combine temporal modeling with graph structure, in our datasets cyberattacks are typically localized to a small subregion and often occur only over limited time intervals, while the three-phase node voltages may exhibit only subtle deviations. In this setting, message passing can dilute the weak attack signal by mixing affected nodes with many unaffected neighbors, and global pooling can further obscure these localized anomalies at the graph level.

Therefore, we adopt four widely used sequential learning models: recurrent neural network (RNN), gated recurrent unit (GRU), long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), and transformer (TF) (Vaswani et al., 2017). These models are applied directly to the concatenated node-feature time series, which preserves localized perturbations more effectively. At each time step, we compute the mean node voltage across all nodes in the power grid graph. This aggregated voltage signal forms a time-series input to the sequential models, allowing them to learn temporal patterns and detect deviations

caused by cyberattacks. We note that more scalable spatio-temporal GNN variants remain an important direction for future work.

4.5 Evaluation Metrics

We adopt task-specific evaluation metrics to assess model performance across the different problem settings. Since PowerBench includes detection, localization, and state estimation tasks under both static and temporal settings, our metric choices are designed to match the characteristics of each problem setting. For both binary classification (e.g., line failure detection and cyberattack detection) and multi-label classification tasks (e.g., outage localization), we primarily report accuracy. For binary classification tasks, such as outage detection and cyberattack detection, we additionally report the ROC-AUC score, which provides a threshold-independent measure of the model’s ability to distinguish between classes, especially useful when dealing with relatively imbalanced data. In particular, these datasets become relatively imbalanced once all attack or outage locations are merged into a single positive class, making ROC-AUC an important complement to accuracy.

In the cyberattack localization task (multi-label), where the label matrix is often sparse (i.e., most labels are zeros), we report exact match accuracy, which measures the proportion of samples for which the predicted set of labels exactly matches the true set of labels. This is especially appropriate because the goal is to correctly identify the full set of attacked components. For state estimation (regression), we report mean squared error (MSE) and mean absolute error (MAE), averaged over all test sets. MSE penalizes large errors, while MAE measures average absolute error. Overall, these metrics provide practical default evaluation measures for the corresponding benchmark tasks.

5 Benchmark Results and Analysis

5.1 Static Task Results

The performance of several baseline models for the line failure detection and subnetwork-level outage localization task is assessed on multiple benchmark DNs. Table 3 reports the mean accuracy and standard deviation for each baseline model for the detection task. Across all datasets, the topological model MP-MLP consistently achieves the highest or comparable performance to the best baselines. On the IEEE 34-bus network, MP-MLP attains the highest accuracy of 97.87%, surpassing both GNN-based and Transformer-based models. This result highlights the ability of multipersistance to effectively capture global topological changes in the network induced by line failures. The IEEE 8500-node network, being the largest and most complex, further demonstrates the scalability and strength of our approach. Although GIN (83.45%) and Graphormer (85.39%) perform reasonably well, MP-MLP achieves 86.30%, second only to GPS (87.46%). This suggests that topological features derived from multipersistance provide strong discriminative power

Table 2: **State Estimation.** Performance of baseline models on PowerBench state estimation datasets.

Model	34-bus		123-bus		8500-node	
	MSE	MAE	MSE	MAE	MSE	MAE
GCN	0.4548	0.4777	0.2777	0.2918	0.2891	0.2819
SAGE	0.2247	0.3685	0.5426	0.5753	0.2504	0.3539
GAT	0.5718	0.5634	1.3743	1.0281	0.4412	0.4475
GIN	0.1549	0.2736	0.5133	0.5139	0.2101	0.2641

Table 3: **Outage detection.** Accuracy results of baseline models on our outage detection datasets. ROC-AUC results are given in Table 4.

Datasets	GNNs					Graph Transformers			Topological	
	GCN	GSAGE	GAT	GIN	TAGCN	Graphormer	UniMP	GPS	SP-MLP	MP-MLP
IEEE 34-bus	96.25±0.24	96.31±0.24	96.19±0.32	96.36±0.25	96.29±0.28	96.33±0.22	96.28±0.27	96.90±0.69	96.04±0.67	97.87±0.48
IEEE 123-bus	95.57±0.35	95.60±0.36	95.66±0.44	96.02±0.37	95.98±0.42	<u>96.35±0.30</u>	96.00±0.34	95.88±1.20	93.68±0.80	96.83±0.70
IEEE 8500-node	72.97±0.51	73.31±0.59	76.33±1.09	83.45±2.23	73.64±0.59	85.39±1.62	75.93±0.75	87.46±0.50	81.14±1.69	<u>86.30±0.78</u>

 Table 4: ROC-AUC results for **line failure detection** across IEEE feeders.

Datasets	GNNs					Graph Transformers			Topological	
	GCN	GSAGE	GAT	GIN	TAGCN	Graphormer	UniMP	GPS	SP-MLP	MP-MLP
IEEE 34-bus	96.68±0.33	96.67±0.35	96.51±0.37	96.78±0.42	96.95±0.35	98.31±0.70	96.53±0.41	98.36±0.47	98.11±0.51	98.56±0.21
IEEE 123-bus	93.92±0.47	93.82±0.55	93.94±0.58	95.37±0.85	94.58±0.49	98.49±0.07	93.90±0.57	98.39±0.38	98.01±0.48	98.66±0.10
IEEE 8500-node	71.13±0.75	70.64±0.60	74.26±2.08	87.90±1.96	71.95±0.71	91.74±2.54	73.08±0.94	91.63±1.50	88.86±1.53	94.93±0.69

even in large-scale networks, where classical GNNs struggle to detect such structural changes. The ROC-AUC score for outage detection is also provided in Table 4.

For the outage localization task, Table 5 presents subnetwork-level accuracy comparisons on the IEEE 34-bus and IEEE 123-bus networks. Overall, localization is a more challenging task, yet topological methods remain highly competitive. On the IEEE 34-bus network, MP-MLP achieves an accuracy of 92.85%, closely matching the best-performing Graphormer (92.93%) and outperforming most GNN baselines, indicating that multipersistance features effectively encode localized structural changes. On the IEEE 123-bus network, MP-MLP reaches 93.53%, outperforming all GNN-based models and second only to GPS (94.08%). These results demonstrate that multipersistance-based representations not only excel in detecting outages but also provide strong localization capability.

Why is MP-MLP Effective? MP-MLP outperforms standard GNNs primarily because it captures global structural changes induced by outages without relying on iterative message passing. In contrast, GNNs tend to dilute localized anomaly signals through neighborhood aggregation and global pooling, especially under partial observability. By leveraging multipersistance features, MP-MLP preserves discriminative topological signatures and avoids over-smoothing, leading to improved performance.

The Average MSE and MAE of baseline models for bus voltage state estimation is presented in Table 2. GIN performs comparatively well to other methods across datasets. On the IEEE 34-bus network, it achieves the lowest MSE (0.1549) and MAE (0.2736). For the IEEE 123-bus and 8500-node networks, GIN also yields good results, indicating its robustness and effectiveness in predicting node voltage in both small and large-scale networks.

 Table 5: **Outage localization.** Accuracy results of baseline models on our outage localization datasets.

Dataset	GCN	GAT	GIN	GraphSAGE	UniMP	Graphormer	GPS	SP-MLP	MP-MLP
IEEE 34-bus	90.53±0.47	90.69±1.65	92.82±0.33	91.85±0.29	92.40±0.32	92.93±0.32	92.76±0.33	90.42±0.30	<u>92.85±0.24</u>
IEEE 123-bus	90.62±0.31	90.76±0.48	92.62±0.26	91.62±0.23	91.99±0.33	92.66±0.30	94.08±0.37	91.93±0.45	<u>93.53±0.47</u>

Table 6: **Cyberattack detection.** Accuracy results of baseline models on our cyberattack detection datasets. ROC-AUC results are given in Table 7.

Baseline	PV Attack			EV Attack			Sensor Attack		
	34-bus	123-bus	8500-node	34-bus	123-bus	8500-node	34-bus	123-bus	8500-node
RNN	87.55±2.74	78.05±2.37	77.74±0.69	84.60±1.63	78.75±3.65	75.00±4.89	98.70±1.01	97.68±0.70	54.80±3.33
LSTM	86.35±3.35	76.32±3.09	78.32±1.40	85.10±2.20	79.15±4.14	75.00±3.20	97.75±1.23	97.55±0.79	53.70±2.54
GRU	86.40±2.46	77.32±2.50	78.47±1.27	86.20±1.67	79.10±3.42	75.61±2.95	97.85±1.62	96.50±0.78	52.95±4.32
TF	85.55±4.92	75.28±2.17	78.53±0.90	85.65±2.39	79.00±4.41	75.83±4.80	98.45±1.01	97.12±0.83	57.65±2.73

Table 7: ROC-AUC results for **cyberattack detection** across attack types and IEEE feeders.

Baseline	PV Attack			EV Attack			Sensor Attack		
	34-bus	123-bus	8500-node	34-bus	123-bus	8500-node	34-bus	123-bus	8500-node
RNN	94.40±2.67	78.67±2.61	56.62±1.10	89.80±1.18	78.91±3.36	58.13±3.49	98.81±0.30	98.87±0.11	57.62±3.35
LSTM	91.86±1.86	75.98±1.99	62.42±2.81	90.03±1.35	79.67±2.87	57.13±4.89	98.52±0.66	98.85±0.19	57.66±3.66
GRU	92.53±2.80	77.97±2.38	60.44±2.22	91.35±1.30	80.03±2.02	62.39±4.99	98.64±0.34	98.59±0.24	56.47±2.86
TF	92.60±3.37	74.23±2.96	62.00±3.17	91.31±1.24	80.61±2.60	59.33±4.39	98.86±0.23	98.49±0.38	57.98±4.35

5.2 Time Series Task Results

The performance of various baseline models on two key task settings, i.e., cyberattack detection and attack localization for attacks on grid-connected PV systems, EV charging stations, and voltage sensors are evaluated. Table 6 summarizes the accuracy of different models across three IEEE networks (34-bus, 123-bus, and 8500-node) for PV, EV, and sensor attacks. The results indicate that baseline models generally perform well on smaller networks, such as the IEEE 34-bus, with a noticeable drop in accuracy as the network size increases. This trend highlights the scalability challenges and increased complexity in larger networks. Despite this, all models achieve relatively competitive performance across datasets, with the Transformer (TF) model delivering the most consistent results, particularly for sensor attacks in the largest network. Because real SCADA, AMI, and DER telemetry may contain measurement noise, we include an additional noise-sensitivity study in Appendix A.3. The corresponding noisy-data version will also be released as part of the PowerBench benchmark.

In Table 8, we present results for the localization task. Compared to detection, localization proves significantly more difficult, particularly in larger networks. Nevertheless, the Transformer model again achieves the highest performance across most tasks and datasets. For example, it reaches 75.65%, 80.05%, and 83.35% mean accuracy for localizing PV, EV, and sensor attacks, respectively, in the IEEE 34-bus network. Even in the IEEE 123-bus network, TF shows superiority over RNN, LSTM, and GRU in most settings. These findings

Table 8: **Cyberattack localization.** Accuracy results of baseline models on our cyberattack localization datasets.

Baseline	PV Attack			EV Attack			Sensor Attack	
	34-bus	123-bus	8500-node	34-bus	123-bus	8500-node	34-bus	123-bus
RNN	63.30±4.02	31.05±2.08	24.65±2.89	77.55±2.73	36.20±4.18	21.50±3.85	64.65±4.28	53.40±2.60
LSTM	62.65±3.76	30.77±1.44	25.75±2.36	76.85±1.83	32.40±2.99	21.67±4.03	68.30±4.22	51.12±2.12
GRU	67.55±3.29	33.88±2.80	25.05±1.86	80.15±2.96	42.70±4.95	22.17±4.75	67.10±3.72	53.72±1.85
TF	75.65±2.67	40.55±2.72	24.60±2.79	80.05±2.79	36.40±4.95	24.17±4.43	83.35±2.57	58.20±4.19

suggest that attention-based models can better leverage temporal dynamics and subtle topological cues for more accurate attack localization. Here, sensor attack localization is omitted for the IEEE 8500-node network due to the high dimensionality of the problem. In these networks, sensors are placed ubiquitously at buses, resulting in an extremely large input space for any learning model. While a theoretical approach could involve partitioning the network into smaller subnetworks with localized threat detection and diagnostics, practical implementations still rely on system-wide measurements and typically use state estimation to detect anomalies. Adapting a machine learning-based localization method under these conditions would necessitate a significant reformulation of the problem, making it inconsistent with the other benchmark tasks. Therefore, for both practical and methodological consistency the 8500-node sensor attack scenario is excluded from model validation. However, the data is included in the benchmark release to facilitate further research.

6 Limitations and Future Work

While PowerBench offers a rich and diverse set of simulated scenarios, it currently does not incorporate real-world sensor noise or other operational anomalies. The benchmarks are based on quasi-static simulations of distribution networks and do not capture transient dynamics, frequency stability. Furthermore, the DER/EVCS placements are fixed and could be diversified to enhance scenario variability. These limitations present opportunities for future work, including the integration of sensor uncertainty and modeling of dynamic system behavior.

7 Conclusion

This paper introduced PowerBench, a comprehensive benchmark suite for machine learning in distribution networks that encompasses outage detection, cyberattack identification, and system state estimation. Built on standard IEEE feeders, PowerBench provides both static snapshots and time-indexed datasets under realistic operating scenarios and partial observability. Each dataset includes consistent graph representations, task annotations, and evaluation protocols to support supervised and semi-supervised learning across diverse network scales and conditions. To demonstrate its utility, we evaluated a broad spectrum of baseline models from classical graph neural networks to recent graph transformers. By bridging the gap between power systems simulation and ML-ready datasets, PowerBench establishes an extensible platform for advancing research in graph-based learning for critical infrastructure monitoring and security.

Broader Impact Statement

PowerBench broadens the impact of resilience research in modern distribution networks by providing openly accessible, physics informed benchmark datasets and standardized evaluation protocols for outage detection and localization, cyberattack detection and localization, and state estimation under partial observability. By enabling reproducible comparisons across graph neural networks, graph transformers, and topology aware models at multiple feeder scales, it lowers the barrier for researchers to develop and stress test reliable monitoring

methods, helping accelerate progress toward faster situational awareness and improved grid reliability. The resource is intended for defensive and research use, avoiding utility sensitive operational data while supporting education and cross disciplinary collaboration between power systems and machine learning communities.

Acknowledgments and Disclosure of Funding

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

References

- Samrat Acharya, Hafiz Anwar Ullah Khan, Ramesh Karri, and Yury Dvorkin. Madeviot: Cyberattacks on ev charging can disrupt power grid operation. In *2024 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5. IEEE, 2024.
- Saima Akhtar, Muhammad Adeel, Muhammad Iqbal, Abdallah Namoun, Ali Tufail, and Ki-Hyung Kim. Deep learning methods utilization in electric power systems. *Energy Reports*, 10:2138–2151, 2023.
- Emran Altamimi, Abdulaziz Al-Ali, Qutaibah M Malluhi, and Abdulla K Al-Ali. Smart grid public datasets: Characteristics and associated applications. *IET Smart Grid*, 7(5): 503–530, 2024.
- Ali Moradi Amani and Mahdi Jalili. Power grids as complex networks: Resilience and reliability analysis. *Ieee Access*, 9:119010–119031, 2021.
- R. F. Arritt and R. C. Dugan. The ieee 8500-node test feeder. In *IEEE PES T&D 2010*, pages 1–6, 2010. doi: 10.1109/TDC.2010.5484381.
- Magnus Bakke Botnan and Michael Lesnick. An introduction to multiparameter persistence. *arXiv preprint arXiv:2203.14289*, 2022.
- Richard J Campbell and Sean Lowry. Weather-related power outages and electric system resiliency. Congressional Research Service, Library of Congress Washington, DC, 2012.
- Yuzhou Chen, Roshni Anna Jacob, Yulia R Gel, Jie Zhang, and H Vincent Poor. Learning power grid outages with higher-order topological neural networks. *IEEE Transactions on Power Systems*, 2023.
- Shweta Dahale, Hazhar Sufi Karimi, Kexing Lai, and Balasubramaniam Natarajan. Sparsity based approaches for distribution grid state estimation-a comparative study. *IEEE Access*, 8:198317–198327, 2020.
- Kaveh Dehghanpour, Zhaoyu Wang, Jianhui Wang, Yuxuan Yuan, and Fankun Bu. A survey on state estimation techniques and challenges in smart distribution systems. *IEEE Transactions on Smart Grid*, 10(2):2312–2322, 2018.

- Andac Demir, Baris Coskunuzer, Yulia Gel, Ignacio Segovia-Dominguez, Yuzhou Chen, and Bulent Kiziltan. Todd: Topological compound fingerprinting in computer-aided drug discovery. *Advances in Neural Information Processing Systems*, 35:27978–27993, 2022.
- Tamal Krishna Dey and Yusu Wang. *Computational Topology for Data Analysis*. Cambridge University Press, 2022.
- Jian Du, Shanghang Zhang, Guanhong Wu, José MF Moura, and Soumya Kar. Topology adaptive graph convolutional networks. *arXiv preprint arXiv:1710.10370*, 2017.
- Roger C Dugan and Thomas E McDermott. An open source platform for collaborating on smart grid research. In *2011 IEEE power and energy society general meeting*, pages 1–7. IEEE, 2011.
- Mohamad El Iaali, Reza Razi, Antoine Bruyere, Bruno Francois, Joao Soares, and Zita Vale. Machine learning methods for state estimation in sparse distribution networks. In *2025 IEEE Conference on Artificial Intelligence (CAI)*, pages 376–381. IEEE, 2025.
- Khairul Eahsun Fahim, Kassim Kalinaki, LC De Silva, and Hayati Yassin. The role of machine learning in improving power distribution systems resilience. *Future Modern Distribution Networks Resilience*, pages 329–352, 2024.
- Salah Ghamizi, Aleksandar Bojchevski, Aoxiang Ma, and Jun Cao. Safepowergraph: Safety-aware evaluation of graph neural networks for transmission power grids. *arXiv preprint arXiv:2407.12421*, 2024. URL <https://github.com/yamizi/SafePowerGraph>.
- W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1025–1035, 2017.
- High Performance Computing at UTD. Juno cluster, hpc@utd - high performance computing at the university of texas at dallas. <https://www.utdallas.edu/oit/research/hpc>. Accessed May 2025.
- Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- Bohao Huang, Jichen Yang, Artem Streltsov, Kyle Bradbury, Leslie M Collins, and Jordan M Malof. Gridtracer: Automatic mapping of power grids using deep learning and overhead imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 15:4956–4970, 2021.
- Roshni Anna Jacob, Steve Paul, Souma Chowdhury, Yulia R Gel, and Jie Zhang. Real-time outage management in active distribution networks using reinforcement learning over graphs. *Nature Communications*, 15(1):4766, 2024.
- Hamidreza Jahangir, Subhash Lakshminarayana, and H Vincent Poor. Charge manipulation attacks against smart electric vehicle charging stations and deep learning-based detection mechanisms. *IEEE Transactions on Smart Grid*, 2024.

- William H Kersting. Radial distribution test feeders. *IEEE Transactions on Power Systems*, 6(3):975–985, 2001.
- Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
- Dheepak Krishnamurthy and Paulo Meira. Opendssdirect.py: a cross-platform python package that implements a native/direct library interface to the alternative opendss engine from dss-extensions.org, 2024. URL <https://dss-extensions.org/OpenDSSDirect.py/>. Released on 2024-02-12.
- Gaoqi Liang, Junhua Zhao, Fengji Luo, Steven R Weller, and Zhao Yang Dong. A review of false data injection attacks against modern power systems. *IEEE Transactions on Smart Grid*, 8(4):1630–1638, 2016.
- M Majidi, A Arabali, and M Etezadi-Amoli. Fault location in distribution networks by compressive sensing. *IEEE Transactions on Power Delivery*, 30(4):1761–1769, 2014.
- National Renewable Energy Laboratory (NREL). National solar radiation database (nsrdb), 2020. Accessed 2024. Data for Location ID 1614490, year 2018. Available at <https://nsrdb.nrel.gov>.
- Christian Nauck, Michael Lindner, Konstantin Schürholt, and Frank Hellmann. Toward dynamic stability assessment of power grid topologies using graph neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(10), 2023.
- Damilola R Olojede, Md Joshem Uddin, Roshni Anna Jacob, Baris Coskunuzer, and Jie Zhang. Cyber attack detection in distribution networks with topological data analytics aided learning. In *2024 IEEE Kansas Power and Energy Conference (KPEC)*, pages 1–6. IEEE, 2024.
- Anggoro Primadianto and Chan-Nan Lu. A review on distribution system state estimation. *IEEE Transactions on Power Systems*, 32(5):3875–3883, 2016.
- Junjian Qi, Adam Hahn, Xiaonan Lu, Jianhui Wang, and Chen-Ching Liu. Cybersecurity for distributed energy resources and smart inverters. *IET Cyber-Physical Systems: Theory & Applications*, 1(1):28–39, 2016.
- Ladislav Rampásek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. *Advances in Neural Information Processing Systems*, 35:14501–14515, 2022.
- Yunsheng Shi, Zhengjie Huang, Shikun Feng, Hui Zhong, Wenjing Wang, and Yu Sun. Masked label prediction: Unified message passing model for semi-supervised classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pages 1548–1554, 2021.
- Anna Varbella, Kenza Amara, Blazhe Gjorgiev, Mennatallah El-Assady, and Giovanni Sansavini. Powergraph: A power grid benchmark dataset for graph neural networks. *Advances in Neural Information Processing Systems*, 37:110784–110804, 2024.

- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations (ICLR)*, 2018. URL <https://arxiv.org/abs/1710.10903>.
- Oliver Vipond, Joshua A Bull, Philip S Macklin, Ulrike Tillmann, Christopher W Pugh, Helen M Byrne, and Heather A Harrington. Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors. *Proceedings of the National Academy of Sciences*, 118(41):e2102166118, 2021.
- Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.
- Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*, 2019. URL <https://arxiv.org/abs/1810.00826>.
- Jin Ye, Annarita Giani, Ahmed Elasser, Sudip K Mazumder, Chris Farnell, Homer Alan Mantooth, Taesic Kim, Jianzhe Liu, Bo Chen, Gab-Su Seo, et al. A review of cyber-physical security for photovoltaic systems. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 10(4):4879–4901, 2021.
- Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.
- Abdollah Younesi, Hossein Shayeghi, Zongjie Wang, Pierluigi Siano, Ali Mehrizi-Sani, and Amin Safari. Trends in modern power systems resilience: State-of-the-art review. *Renewable and Sustainable Energy Reviews*, 162:112397, 2022.
- Chaohao Yuan, Kangfei Zhao, Ercan Engin Kuruoglu, Liang Wang, Tingyang Xu, Wenbing Huang, Deli Zhao, Hong Cheng, and Yu Rong. A survey of graph transformers: Architectures, theories and applications. *arXiv preprint arXiv:2502.16533*, 2025.
- Yuxuan Yuan, Kaveh Dehghanpour, Fankun Bu, and Zhaoyu Wang. Outage detection in partially observable distribution systems using smart meters and generative adversarial networks. *IEEE Transactions on Smart Grid*, 11(6):5418–5430, 2020.

Appendix

Appendix A. Dataset Generation and Experimental Setup

This appendix documents the end-to-end setup used to generate PowerBench and to train and evaluate baseline models. We first describe the simulation and component modeling assumptions used in OpenDSS, including PV and EVCS placements and load profiles. We then detail the construction of time-series cyberattack scenarios. Finally, we summarize the learning protocol used for benchmarking, including handling of partial observability, preprocessing, and evaluation metrics.

A.1 Simulation Environment and Component Modeling

PowerBench scenarios are generated by running power-flow simulations on standard IEEE distribution feeders with additional DER and EV charging components. All simulations are carried out in OpenDSS. Each scenario corresponds to either a single steady-state snapshot (static tasks) or a 24-hour trajectory (temporal tasks), from which node- and edge-level measurements are recorded and mapped to a graph representation.

A.1.1 PHOTOVOLTAIC (PV) SYSTEMS

To emulate DER penetration, we add utility-scale PV systems to each feeder using the OpenDSS `PVSystem` element (PV array and inverter). We deploy PV units at 3, 5, and 7 buses for the IEEE 34-bus, 123-bus, and 8500-node feeders, respectively. Irradiance is driven by 5-minute resolution Global Horizontal Irradiance (GHI) data from the National Solar Radiation Database (NSRDB) (National Renewable Energy Laboratory (NREL), 2020). Table 9 list PV locations and ratings for each feeder.

Table 9: Photovoltaic (PV) systems deployed in PowerBench IEEE feeders.

Feeder	ID	Bus	Phases	Rated Power (kW)	Voltage level (kV)
IEEE 34-bus	1	890	3	146	4.16
	2	806	3	144	24.9
	3	816	3	200	24.9
IEEE 123-bus	1	56	3	200	4.16
	2	79	3	210	4.16
	3	82	3	220	4.16
	4	91	3	225	4.16
	5	108	3	250	4.16
IEEE 8500-node	1	13104136	3	1000	12.47
	2	12895449	3	1000	12.47
	3	13010560	3	1050	12.47
	4	12876797	3	1050	12.47
	5	12876814	3	1100	12.47
	6	13081380	3	1200	12.47
	7	12766718	3	1500	12.47

A.1.2 LOAD PROFILES

To capture temporal demand variability, we use annual representative load profiles with hourly resolution provided within OpenDSS. For each scenario, the selected load shape is applied to all loads via time-dependent multipliers on the base demand, yielding realistic operating conditions across different times of the year.

A.1.3 ELECTRIC VEHICLE CHARGING STATIONS (EVCS)

EV charging stations are modeled using OpenDSS **Storage** elements configured in charging-only mode. Time-varying charging profiles derived from real-world EVCS usage data in Texas are used as active power inputs. We deploy 3, 5, and 7 EVCS units for the IEEE 34-bus, 123-bus, and 8500-node feeders, respectively. Table 10 list EVCS locations and ratings.

Table 10: Electric vehicle charging stations (EVCS) deployed in PowerBench IEEE feeders.

Feeder	ID	Bus	Phases	Voltage level (kV)	Rated Power (kW)
IEEE 34-bus	1	814	3	24.9	368.5
	2	890	3	4.16	261
	3	852	3	24.9	544.5
IEEE 123-bus	1	25	3	4.16	368.5
	2	40	3	4.16	261
	3	54	3	4.16	544.5
	4	62	3	4.16	368.5
	5	76	3	4.16	544.5
IEEE 8500-node	1	l3104136	3	12.47	368.5
	2	l2895449	3	12.47	261
	3	l3010560	3	12.47	544.5
	4	l2876797	3	12.47	368.5
	5	l2876814	3	12.47	261
	6	l3081380	3	12.47	544.5
	7	l2766718	3	12.47	261

A.2 Temporal Cyberattack Scenario Construction

Overview. Temporal cyberattack datasets consist of 24-hour simulations recorded at discrete intervals. The sampling resolution depends on the scenario type (e.g., 30-minute intervals for PV and sensor attacks, and 5-minute intervals for EVCS attacks). Each trajectory is simulated under diverse load and generation profiles. The resulting time series is mapped to a sequence of graphs while preserving the correspondence between physical buses/branches and graph nodes/edges.

Sensor attack construction. Sensor attacks are generated by injecting adversarial voltage measurements at selected buses during the time-series simulation:

1. Run a baseline 24-hour simulation with a selected load shape (and DER/EVCS profiles when applicable).

2. Sample one or more target buses, together with an attack start time and attack duration.
3. During the attack window, perturb voltage magnitudes at target buses by a multiplicative scaling factor sampled from:
 - $[0.00, 0.85]$ for undervoltage attacks,
 - $[1.05, 1.30]$ for overvoltage attacks.

Outside the attack window, measurements remain unchanged and are consistent with the underlying power-flow solution.

A.3 Robustness to Measurement Noise

Since PowerBench is generated from simulation, it is important to evaluate whether the benchmark remains informative under realistic measurement uncertainty. To this end, we constructed noisy variants of the EVCS attack datasets by injecting additive white Gaussian noise (AWGN) into the sensor measurements of the IEEE 34-bus and 123-bus networks.

For each sensor measurement $x[b, t]$, where b indexes the sensor and t denotes the 5-minute time step, the noisy observation is defined as

$$\tilde{x}[b, t] = x[b, t] + \varepsilon[b, t], \quad \varepsilon[b, t] \sim \mathcal{N}(0, \sigma_b^2).$$

The noise scale is chosen relative to the empirical variation of each sensor:

$$\sigma_b = \alpha \cdot \sigma_{\text{global}, b}, \quad \alpha = 10^{-\text{SNR}_{\text{dB}}/20},$$

where $\sigma_{\text{global}, b}$ denotes the standard deviation of feature b computed over all scenarios and time steps in the clean dataset. This feature-wise scaling is important because the measurements have different physical magnitudes, such as per-unit bus voltages, branch power flows, and EVCS active power. In the results reported here, we use $\text{SNR}_{\text{dB}} = 10$, which corresponds to $\alpha \approx 0.316$. Noise is applied independently to bus voltage magnitudes, branch power flows, and EVCS active power measurements, while attack labels and targeted EVCS identifiers remain unchanged.

Table 11 compares model performance on the original and noisy datasets. As expected, all baselines show performance degradation under noise, confirming that measurement

Table 11: Comparison of EVCS attack detection performance on the original and noisy datasets at $\text{SNR} = 10$ dB.

Dataset	Model	Acc (Orig)	Acc (Noisy)	AUC (Orig)	AUC (Noisy)
34-bus	RNN	84.60±1.63	80.55±1.57	89.80±1.18	86.58±1.14
	LSTM	85.10±2.20	82.90±0.76	90.03±1.35	88.50±0.80
	GRU	86.20±1.67	81.15±1.64	91.35±1.30	86.71±1.85
	Transformer	85.65±2.39	82.70±1.34	91.31±1.24	89.02±1.67
123-bus	RNN	78.75±3.65	77.50±3.54	78.91±3.36	65.49±3.60
	LSTM	79.15±4.14	78.62±2.40	79.67±2.87	70.48±3.03
	GRU	79.10±3.42	78.38±1.30	80.03±2.02	67.55±3.84
	Transformer	79.00±4.41	78.96±2.77	80.61±2.60	73.45±6.76

uncertainty makes EVCS attack detection more challenging. On the 34-bus dataset, both accuracy and ROC-AUC decrease moderately across all models, with LSTM and Transformer remaining the most robust baselines. On the 123-bus dataset, the drop in accuracy is relatively small, while the drop in ROC-AUC is more pronounced, indicating that noise mainly reduces the separability between normal and attacked scenarios. Overall, these results show that PowerBench can support robustness evaluation beyond clean synthetic data and provide a controlled setting for studying model behavior under realistic measurement corruption.

The noisy-data version used in this experiment will also be released with PowerBench, together with the corresponding data splits and evaluation scripts, so that users can directly benchmark models under both clean and noisy measurement settings.

Appendix B. Learning and Evaluation Protocol

This section summarizes the benchmark training pipeline shared across tasks, including input construction under partial observability, preprocessing, and the evaluation measures reported in the main paper and appendix.

B.1 Static Tasks: Outage Detection and Localization

Line failure detection is formulated as **binary graph classification**. Each node is represented by a 3-dimensional feature vector of per-phase voltages. For GNN and graph transformer baselines, we apply **global mean pooling** over node embeddings to obtain a graph-level representation, followed by a linear classifier. Outage localization uses the same architecture template but predicts a multi-label region vector.

B.2 Additional Results: ROC-AUC for Detection Tasks

Table 4 reports ROC-AUC for line failure detection, complementing the accuracy results in the main paper.

B.3 Temporal Tasks: Cyberattack Detection and Localization

Cyberattack detection and localization are treated as **time-series classification** baselines using RNN, LSTM, GRU, and Transformer models. At each time step we compute a graph-level summary from voltages and concatenate these summaries across time to form a sequence input to the temporal model. For localization, predictions are evaluated using **exact match** accuracy.

B.4 Partial Observability and Preprocessing

A key challenge across tasks is **partial observability**. For nodes without voltage measurements, we use a placeholder feature vector of $[-1, -1, -1]$ to indicate missing values. Since observed voltages are often tightly concentrated in operating ranges, **feature normalization is critical** for stable training; we apply Min-Max normalization to node features before training (for temporal tasks, normalization is applied consistently across all time steps). For

topological baselines, graph-level embeddings are computed from single- and multi-parameter persistence summaries (Appendix C) and classified using an MLP.

B.5 State Estimation

State estimation is formulated as **regression** to predict 3-phase voltage vectors at unobserved nodes from sparse measurements. During training, a binary mask indicates observed vs. unobserved nodes and the model learns from partially visible inputs. At test time, voltages at unobserved nodes are set to zero and performance is reported using **MSE** and **MAE**, averaged over all test graphs.

Appendix C. Topological Methods on Power Grid Networks

This appendix expands on the topological methods used in our analysis of power grid networks. While the main paper provides a brief overview, here we detail how single- and multi-parameter persistent homology extracts robust structural signatures from graph representations of distribution systems. We outline the construction of sublevel and bifiltrations, the interpretation of persistence diagrams, and their vectorization for ML tasks. These topological summaries offer principled, scalable descriptors of evolving grid conditions, enabling reliable detection of outages, attacks, and system instabilities.

C.1 Single Persistence

Persistent Homology (PH) provides a principled framework from algebraic topology to identify and track topological structures hidden within data. In the context of graph-structured data such as power grids, PH enables the discovery of evolving patterns like connected components, loops, and voids as we examine the system under different resolutions or filtration levels. This section outlines the single-parameter PH pipeline applied to graphs, particularly power networks. For broader applications in point clouds and images, refer to Dey and Wang (2022).

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ denote a weighted graph with node set $\mathcal{V}$ and edge set $\mathcal{E}$, where $|\mathcal{V}| = N$. The edge weight matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$ is symmetric and defined as $\omega_{rs} > 0$ for each edge $e_{rs} \in \mathcal{E}$ and $\omega_{rs} = 0$ otherwise. For unweighted graphs, we assume $\omega_{rs} = 1$ for $e_{rs} \in \mathcal{E}$ and 0 otherwise. The PH pipeline consists of three key stages: filtration, persistence diagram construction, and vectorization.

Step 1: Filtration Construction

The first step involves building a filtration, a nested sequence of subgraphs and their associated simplicial complexes. This filtration encodes how the topology of the graph evolves with respect to a scalar function. We begin by defining a filtering function $f : \mathcal{V} \rightarrow \mathbb{R}$, which can represent node attributes such as voltage, power injection, or structural properties like degree or centrality.

We select an increasing set of thresholds $\{\alpha_1, \alpha_2, \dots, \alpha_N\}$ such that $\alpha_1 = \min_{v \in \mathcal{V}} f(v)$ and $\alpha_N = \max_{v \in \mathcal{V}} f(v)$. For each α_i , we define the vertex subset $\mathcal{V}_i = \{v \in \mathcal{V} \mid f(v) \leq \alpha_i\}$ and the induced subgraph $\mathcal{G}^i = (\mathcal{V}_i, \mathcal{E}_i)$, where $\mathcal{E}_i = \{e_{rs} \in \mathcal{E} \mid v_r, v_s \in \mathcal{V}_i\}$. This generates a sequence $\mathcal{G}^1 \subseteq \dots \subseteq \mathcal{G}^N = \mathcal{G}$ known as the *sublevel filtration*.

Next, we lift these subgraphs to simplicial complexes. For each $\mathcal{G}^i$, we construct a simplicial complex $\widehat{\mathcal{G}}^i$ by using, for example, the *clique complex*, where each $(k+1)$ -clique in $\mathcal{G}^i$ is completed with a k -simplex. The result is a filtration of simplicial complexes:

$$\widehat{\mathcal{G}}^1 \subseteq \widehat{\mathcal{G}}^2 \subseteq \dots \subseteq \widehat{\mathcal{G}}^N.$$

In power grids, this construction allows one to model and analyze how the topological structure evolves as operational parameters change. For example, filtering by voltage can help identify disconnected components during faults or outages, as buses with near-zero voltages will appear earlier in the filtration.

Step 2: Persistence Diagrams

The second step involves tracking the birth and death of topological features across the filtration using homology theory. Each k -dimensional feature (e.g., connected component for $k=0$, loop for $k=1$) appears at some step b_σ and disappears at d_σ , where $1 \leq b_\sigma < d_\sigma \leq N$. These are referred to as the birth and death times, and the lifespan is $d_\sigma - b_\sigma$.

Such features are captured in the k -th persistence diagram:

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

where H_k denotes the k -th homology group representing k -dimensional holes.

In power grid applications, persistent features highlight anomalous regions. For example, during an outage, nodes with zero voltage appear early in the filtration, forming isolated components (0-holes) that later merge as the threshold increases. The evolution of these components differs significantly between normal and fault conditions, which is reflected in their persistence diagrams.

Step 3: Vectorization of Persistence Diagrams

Persistence diagrams are sets of points in $\mathbb{R}^2$, making them less suitable for direct use in machine learning tasks. Thus, the final step involves vectorizing these diagrams. *Single persistence (SP) vectorizations* map diagrams to fixed-dimensional vectors or functions, enabling integration with statistical and ML models.

Popular SP vectorization methods include *Persistence Images*, *Persistence Landscapes*, *Persistence Silhouettes*, and *Persistence Curves*. These representations summarize the topological features of the data in a structured and learnable form. In this work, we aim to enhance SP vectorizations by incorporating Betti numbers, where the i -th Betti number β_i counts the number of i -dimensional features. This opens up the possibility for capturing more complex temporal evolutions and multidimensional patterns by treating power grids as dynamic systems. Such multipersistence perspectives yield a richer topological fingerprint for anomaly detection and resilience analysis.

C.2 Multipersistence for Power Grids

Motivation. In the context of power grids, many types of measurements—such as bus voltages, branch flows, and power injections—contain crucial information about the system’s dynamics and vulnerability. However, these signals are often interdependent and cannot be

fully understood in isolation. This motivates the need for multiparameter persistence, where the interaction between multiple measurement modalities can be captured via topological descriptors that evolve over simultaneous filtrations.

Filtration Functions. In power grid networks $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, multipersistence begins by selecting a pair (or tuple) of scalar-valued functions that reflect key physical or structural properties of the grid. For instance, we can define:

- Node-based functions such as bus voltage magnitude $f : \mathcal{V} \rightarrow \mathbb{R}$ and electrical centrality (e.g., degree, betweenness).
- Edge-based functions such as branch flow magnitude $g : \mathcal{E} \rightarrow \mathbb{R}$ or line impedance.

These functions provide complementary views of the grid’s condition—voltage encodes potential imbalance or instability at nodes, while flow values signal loading stress across transmission lines.

Bifiltration Construction. Let us consider two filtering functions $f, g : \mathcal{V} \rightarrow \mathbb{R}$ defined on the nodes (e.g., voltage and centrality), or $f : \mathcal{V} \rightarrow \mathbb{R}$ and $g : \mathcal{E} \rightarrow \mathbb{R}$ (e.g., voltage and flow). We construct a bifiltration of subgraphs by defining a grid of thresholds $\{\alpha_i\}_{i=1}^m$ and $\{\beta_j\}_{j=1}^n$ for f and g , respectively. For each pair (α_i, β_j) , define:

$$\mathcal{V}^{ij} = \{v \in \mathcal{V} \mid f(v) \leq \alpha_i, g(v) \leq \beta_j\},$$

and let $\mathcal{G}^{ij}$ be the subgraph of $\mathcal{G}$ induced by $\mathcal{V}^{ij}$ and the edges between them. This results in a bifiltration $\{\mathcal{G}^{ij}\}_{i,j}$, which captures how the topological structure of the network evolves as both physical parameters change.

Topological Signatures. From this bifiltration, we compute homology groups $\{H_k(\mathcal{G}^{ij})\}$ for each index pair (i, j) , tracking the evolution of k -dimensional topological features such as connected components (H_0), loops (H_1), and voids (H_2). These features often correlate with physically meaningful patterns in power grids:

- Emergence or disappearance of isolated regions (e.g., disconnected buses during outages).
- Creation of high-flow loops that may indicate overload or rerouting.

The result is a bigraded persistence module that captures the lifetime of these features across both filtrations.

Challenges and Representations. Unlike single-parameter persistence, the partial order in bifiltration space (α_i, β_j) prevents a barcode-like decomposition. Nevertheless, various recent tools such as rank invariant surfaces, fibered barcodes, Hilbert functions, and multidimensional persistence images have been developed to summarize this high-dimensional topological information Botnan and Lesnick (2022); Vipond et al. (2021); Demir et al. (2022). These summaries can be vectorized and used as discriminative features for downstream machine learning tasks such as outage detection or stability classification.

Application. In our setting, we construct the bifiltration using node voltages and branch flows. This dual-perspective approach allows us to track how topology evolves as less critical

elements (low voltage, low flow) are gradually added to the network. Changes in Betti numbers or persistent feature maps under this filtration serve as unique fingerprints of grid conditions, making them powerful tools for identifying anomalies or predicting outages.

Appendix D. Extended Description of PowerBench

This appendix complements the main paper by documenting implementation-facing details of PowerBench that are useful for reuse and reproducibility. We summarize (i) the benchmark scope and supported tasks, (ii) the graph representation and labeling conventions, and (iii) the key axes of variability that control difficulty, including feeder scale, sensing regimes, and adversarial settings. Our goal is to make it straightforward for readers to reproduce the released datasets, train comparable baselines, and extend the benchmark to new feeders or event types.

D.1 Benchmark Scope and Tasks

PowerBench converts standard IEEE distribution feeders into ML-ready graph datasets for resilience-oriented monitoring under partial observability. It supports three task families: (i) **line failure detection** as binary graph classification (normal vs. at least one failed line), (ii) **outage localization** as multi-label graph classification over predefined operational regions, and (iii) **state estimation** as node-level regression for imputing unobserved voltages from sparse measurements. In addition, PowerBench provides **temporal cyberattack detection and localization** as spatio-temporal graph classification for PV, EV charging, and voltage sensor attack scenarios, where labels indicate whether an attack occurs within a time window and which components are compromised.

D.2 Graph Representation and Labels

Each sample is a distribution feeder represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a fixed topology per feeder. Nodes correspond to buses and include role metadata (e.g., substation, load, DER, EVCS, sensor), while edges correspond to lines or inline transformers and store physical attributes (e.g., impedance and phase information). Node features include per-phase voltage measurements at instrumented buses and are treated as missing at uninstrumented buses according to the selected sensing regime. For static tasks, each graph snapshot is paired with a task-specific label: a binary outage indicator for detection, a multi-hot region vector for localization, or voltage targets for state estimation. For temporal tasks, each instance is a sequence $\{\mathcal{G}_t\}_{t=1}^{|t|}$ with labels defined over the full sequence window (attack present vs. absent, and a multi-hot vector over attacked components).

D.3 Axes of Variability and Scalability

PowerBench is designed to test robustness and generalization across multiple sources of heterogeneity:

- **Feeder scale and topology:** IEEE 34-bus, 123-bus, and 8500-node feeders spanning distinct graph sizes and structural properties.

- **Operating conditions:** diverse load and DER profiles, including time-varying demand and generation patterns.
- **Event diversity:** line outage scenarios for static monitoring, and multiple cyberattack types for temporal monitoring (PV, EVCS, and sensor attacks).
- **Observability regimes:** full and sparse sensing settings with fixed sensor placements within each dataset split to enable consistent comparisons.
- **Static vs. temporal modeling:** single-snapshot inference for outage detection/localization and state estimation, and multi-step trajectories for attack detection/localization.

These axes allow PowerBench to scale from small feeders to large, highly heterogeneous networks while keeping the benchmark protocol consistent. The released framework is modular, enabling the addition of new feeders, sensing layouts, and event generators without changing the graph interface or evaluation pipeline.