PREMIER JOURNAL OF COMPUTER SCIENCE REVIEW

Graph-Based Intrusion Detection for Edge-Cloud IoT Energy Systems

Antonieta Lima

Instituto Superior de Entre Douro e Vouga, Santa Maria da Feira, Portugal

Correspondence to: Antonieta Lima, lima.antonietamaria@gmail.com

Additional material is published online only. To view please visit the journal online.

Cite this as: Lima A. Graph-Based Intrusion Detection for Edge-Cloud IoT Energy Systems. Premier Journal of Computer Science 2025;5:100013

DOI: https://doi.org/10.70389/PJCS.100013

Peer Review

Received: 5 August 2025

Accepted: 16 December 2025

Version accepted: 1

Published: 12 January 2026

Ethical approval: N/a

Consent: N/a

Funding: No industry funding

Conflicts of interest: N/a

Author contribution: Antonieta Lima – Conceptualization, Writing – original draft, review and editing

Guarantor: Antonieta Lima

Provenance and peer-review: Commissioned and externally peer-reviewed

Data availability statement: N/a

ABSTRACT

The accelerated distribution of Internet of Things (IoT) devices in critical energy infrastructure, such as smart grids, creates unmatched attack surface areas available to cyberattacks. While advanced cyberattacks could be deterred by an Intrusion Detection System (IDS), traditional solutions often rely on static signature matching or naive time-series anomaly detection. Therefore, the traditional forms of IDS do not address the agile, dynamic, distributed and heterogeneous nature of IoT networks well. The current paper presents an innovative graph-theoretic intrusion detection framework designed for edge-cloud IoT energy systems. Our approach rigorously models IoT sensor communication and device interactions via dynamic temporal graphs. We apply advanced graph embeddings such as GraphSAGE and Node2Vec, to identify the complex and dynamic layers of the IoT device's behavioral structure. We construct a sequence of temporal graphs embeddings and the other representation on temporal graphs involves training a sparse autoencoder on the temporal sequence of graph embedding to detect anomalous device behaviors that indicate potential cyberattacks, like Advanced Persistent Threats (APTs), or sophisticated spoofing of devices. The proposed hybrid IDS are capable of distributed inference on edge nodes, allowing real time low latency detection on local threats and providing centralized logging and higher level, collective analysis in the cloud. This paper presents the first combination of temporal graph modeling and hybrid edge-cloud anomaly detection for smart grid detection specifically for smart grid environments with minimal compute overhead and practically deployable solution for real-world IoT security.

Keywords: Advanced persistent threat detection, Edge-cloud smart grid IDS, Graphsage, Node2Vec embeddings, Sparse autoencoder anomaly detection, Temporal graph modeling

Introduction

The future promises novel applications of Internet of Things (IoT) devices into society's critical infrastructure, particularly in areas like the energy sector that will help develop smart grids in increasingly advanced ways that help us utilize energy in much more efficient, reliable options, with best automated control of ability to process is available. Subsequently, the multiple forms of interconnectedness create an overwhelming attack surface. Unlike a typical IT network, the scale of resources-constrained heterogeneous devices (e.g. low-power, low-cost embedded devices) in IoT energy systems is much larger and operate with a significant layer of complexity in communication protocols (e.g., Zigbee, LoRaWAN, cellular) and real-time requirements.

Existing intrusion detection systems mostly developed for traditional enterprise IT, struggle to address unique requirements of IoT deployments, such as the necessity to manage high data volume and velocity, diverse device populations with inherent heterogeneity, and ultra-low-latency threat response.[1] Current IoT intrusion detection also mainly exploits statistical methods (and predetermined rule-based systems) or applies deep learning techniques to fixed, isolated time-series data streams. While these known attack vectors can be acted on, these approaches fundamentally ignore the essential relational dependencies and transient relationships or interactions between IoT devices, edge gateways, and cloud. While recognizing relational parameters are critical to the detection of complex, coordinated, or novel threats — which are communicated through changes in volumes and patterns of cyber interactions – rather than data or feature anomalies. Cyber-attacks against a smart grid can include, data manipulation and denial-of-service (DoS) but more sinister Advanced Persistent Threat (APTs) and false data instrumentation (FDI) for example if typically, "high" and "constant" actuator interactions from machine to machine were altered to sparse interactions which might mimic an abandoned production equipment, could leave significant impact to power outages or accelerated physical infrastructure damage.[2]

This paper presents a new, integrated graph-based intrusion detection framework that explicitly captures and analyzes the relational aspects of IoT communication in hybrid edge-cloud energy systems. Our framework leverages temporal graphs of dynamic device interactions and data flows, which provide a means of capturing complex behavioral patterns and their changing nature over time, that would be inherently difficult to extract using a traditional non-relational model. Our approach combines recent advances in graph embedding methods – specifically Node2Vec and GraphSAGE – with a sparse autoencoder architecture, allowing us to detect anomalies in an effective and adaptive way, while emphasizing practical deployment and computational efficiency to suit the real-time requirements of smart grid IoT systems, which have often limited available computational resources.

Related Work

Research into intrusion detection in IoT and smart grid settings has been extensive and has, in general, been divided into signature-based and anomaly-based detection. There are IDS using signature-based detection (example IDS includes Snort and Suricata) which are able to detect known attack patterns. Several surveys have analyzed the existing IDS techniques, the datasets, and research challenges, highlighting the limitations of conventional approaches in evolving environments such as IoT and smart networks.[3–5] However, these IDS systems cannot detect zero-day attacks or new variations. Anomaly-based IDS, on the other hand, identifies a baseline normal behavior for a system and will flag any significant system behavior deviations as a possible intrusion. Initially, anomaly detection on IoT and smart grid systems were based on traditional machine learning (ML) detection techniques. For example, there have been some statistical forms of anomaly detection based on historical value distributions or control charts for detecting anomalies in sensor readings or energy consumption distributions. Other supervised machine learning, such as Support Vector Machines (SVM), K-Nearest Neighbors (KNN), and Random Forests (RF) have been used to predict a classification about network traffic or a device's expected behavior. Most inspect extraction from packet headers or parameters from flow statistics to create hand-crafted features in the ML training examples provided to the classifiers. Unsupervised methods for anomaly detection like K-Means clustering and Isolation Forest have also been used to determine if the predicted values were legitimate without the knowledge of the attack labels. Some recent surveys show a lot of the detail of ML based techniques for IoT intrusion detection.[6–8] It is further added that recent work highlights more prominently the growing importance of robust anomaly detection in smart grid infrastructures and industrial control systems.[9-13]

With the rise of deep learning (DL), more complex models have been utilized to model the high-dimensional and temporal nature of IoT data. For example, recurrent neural networks (RNN), and especially Long Short-Term Memory (LSTM) networks, are able to model temporal relationships in the time-series data streams generated by sensors in smart grids and from network logs.[14–16] Convolutional neural networks (CNN) have been explored for the purpose of feature engineering directly from raw network traffic or data that is visually represented, such as images or graphs.[17,18] Autoencoders (AE) and Variational Autoencoders (VAE) have become popular options for unsupervised anomaly detection in a stream where normal data and anomalies are compressed to learn their respective representations, followed by returning abnormal events based on large reconstruction errors from those representations.[19–21] DL methods can achieve higher accuracy and offer new opportunities for flexibility when compared to generalized traditional ML methods, especially in complicated and everchanging threat landscapes.[22] It is also worth highlighting that other studies also emphasize the growing role of deep learning-based anomaly detection in the monitoring of smart grids and critical infrastructures.[23]

However, a significant drawback to many current ML/DL based IDS in IoT and smart grids is that they often treat network events or sensor readings as independent observations or a sequential time-series. This assumes that the data can be analyzed individually, which means ignoring the rich relational information and complex interdependencies that exist between devices, users and services, and are central to understanding systemic behavior and detecting coordinated attacks. For instance, an attack might involve a series of seemingly benign actions across multiple devices that, when viewed as a whole, reveal a malicious pattern of interaction.[24]

Recently, graph-based methods and graph neural networks (GNNs) have been presented as effective analysis methods for complex relational data in cybersecurity.[25–27] GNNs, in particular, graph convolutional networks (GCNs)[28] and graph attention networks (GANs)[29] have shown superior methods of learning representations from static network graphs. These methods have excelled at tasks of node classification (e.g., identifying a malicious host) and link prediction (e.g., identifying suspicious connections).[30,31] Temporal GNNs (TGNNs) have further expanded these analysis capabilities to dynamic graphs and provided a way to account for the evolving nature of relationships over time.[32–34] Current GNNs and TGNNs are show great promise for detecting advanced persistent threats (APT) and sophisticated botnet activity that create structural anomalies in network communication graphs.[35] At the same time, studies conducted through surveys have reinforced these conclusions, particularly through the detection of anomalies based on graphs and their relevance to complex cyber-physical systems.[36]

Despite advancements in GNNs for general network intrusion detection, research in GNNs specific to the idiosyncrasies of smart grid IoT systems (i.e., dynamic temporal graphs), particularly with a hybrid edge-cloud architecture, remains underexplored. Current GNNs-based IDSs have software architectures that are centralized processing-based, or characterize the network state in static graph representations, which are not good for processing environments while resource constrained edge-devices have real-time requirement.[37,38] Integration of explainable AI (XAI) techniques with GNN-based IDSs for better transparency and trust in critical infrastructure security is also an emerging and critical area.

Despite the promising advancements in GNNs for general network intrusion detection, their specific application to the unique characteristics of smart grid IoT environments, particularly addressing dynamic temporal graphs within a hybrid edge-cloud architecture, remains an underexplored area. Existing GNN-based IDSs often focus on centralized processing or static graph representations, which are not ideal for resource-constrained edge devices and the real-time demands of critical infrastructure.[37,38] Furthermore, the integration of explainable AI (XAI) techniques with GNN-based IDSs for enhanced transparency and trust in critical infrastructure security is an emerging but crucial area.[39]

Our work directly addresses these gaps by proposing a comprehensive framework that integrates dynamic temporal graph modeling, advanced graph embeddings (Node2Vec and GraphSAGE), and sparse autoencoders within an optimized edge-cloud deployment.

This framework is specifically designed to target the nuanced security requirements of smart grid IoT systems, providing a more holistic and adaptive approach to intrusion detection.

Proposed Methodology

The architecture of the proposed Graph-Based Intrusion Detection System (GIDS) for Edge-Cloud IoT Energy Systems supports robust and efficient detection of anomalies, and is designed as a hybrid edge-cloud system to account for both low-latency, on-edge threat detections and high-latency but global security analysis at the centralized cloud.

Dynamic Temporal Graph Construction

The core of our approach is the transformation of communication and interactions logs of IoT devices from a variety of datasets into a series of dynamic temporal graphs. In the context of the smart grid IoT ecosystem, the network's entities and their interactions naturally map to a graph.

Let V be the set of nodes representing each IoT device (e.g., smart meters, sensors, actuators), edge gateways and central cloud servers. Each edge e = (u,v) represents communication links or interactions between u and v. Each edge may have a set of associated attributes Ae such as data volume, packet counts, communication protocol (e.g., MQTT, CoAP), interaction types (e.g., sensor reading, control command). Each node v ∈ V can have its own features F_v such as type of device, location or operational state.

A temporal graph is derived from a series of discrete, non-overlapping temporal windows of duration Δt. For time window t ∈ {1, 2, ..., T}, we define a graph Gt = (V, Et, A_t, F) where Et consists of all the observed interactions within the interval [t ·Δ_t, (t + 1) ·Δ_t), and At consists of the edge attributes for the interactions. Edge attributes At ux, v show the information derived from the interactions during Δ_t. Node features F are typically static or change slowly. Formally, an edge (u, v) is in Et if there was at least one communications event between u and v during the time window Δ_t. Attributes At (u, v) could take any number of forms but could be a vector summarizing the characteristics of the communications in Δ_t (e.g., total bytes, average packet size, number of unique protocols, connection duration, flags associated with the connection such as count of SYN/ACK, etc.). This dynamic graph representation allows us to capture the dynamic nature of network connections communicating over time. Understanding the time-dependent behavior of these connections is crucial for time-dependent attacks detection. The complete concept is depicted on Figure 1.

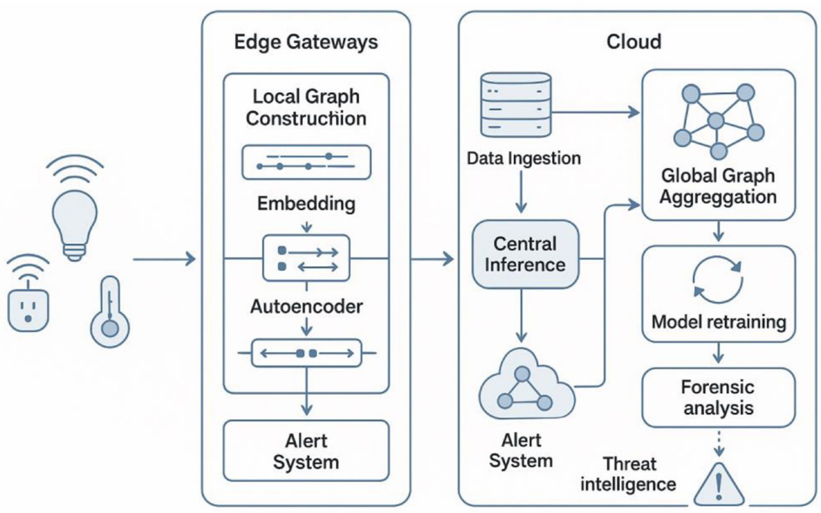

Fig 1 | Conceptual system architecture of graph-based IDS

Graph Embedding for Behavioral Encoding

To enable machine learning models to process the complex, non-Euclidean structure of these temporal graphs, we use graph embedding approaches. Graph embeddings are a way of mapping high-dimensional, sparse graph data into a low-dimensional, dense vector space while maintaining essential topological properties, node attributes, and relationship information. This is an important transformation for converting graph data to something that can be fed into traditional deep learning approaches. For each temporal graph G_t, we will produce node embeddings $X_t = \{x_{v1}, x_{v2}, ..., x_{vN}\}$, where $x_{vi} \in R_d$ is the *d*-dimensional embedding vector for node $_{vi}$. We will consider, integrate, and compare two popular graph embedding methods, chosen in part for their ability to capture different representations of the importance of the network structure and their suitability for dynamic contexts.

Node2Vec:[40] The algorithm is designed to learn node embeddings by optimizing an objective function that preserves neighborhood structure. It creates sequences of nodes (random walks) by balancing a BFS (Breadth First Search) based and a DFS (Depth First Search) based objective. The BFS-like walks allow us to explore local neighborhoods (i.e., explore structural equivalence). These same walks will also be well suited to explore larger network structural neighborhoods (i.e., explore homophily) in the case of the DFS-like walks. The resulting sequences are then fed into a Skip-gram model (just like Word2Vec) which learns node embeddings such that nodes appearing in similar contexts (walks) will have similar embeddings. Where NS(u) refers to the neighborhood of vertex u that is generated by sampling strategy S, and f(u), the embedding of vertex u. Hyperparameters p (return hyperparameter) and q (in-out hyperparameter) are used to provide different bias for the random walk process to get neighborhoods of different types.

GraphSAGE:[41] GraphSAGE, as an inductive framework, creates node embeddings by learning a function that aggregates information about a node's feature representation from its local neighborhood. GraphSAGE differs from Node2Vec in that it learns a function that can produce embeddings for unseen nodes, while Node2Vec produces a fixed embedding for those nodes. The inductive setting of GraphSAGE is helpful for dynamic graphs where new devices and connections can emerge (transductive learning).

The outcome of each individual time window t in this module is a concatenated vector or a matrix consisting of the embeddings of all nodes in G_t. This process transforms the complex structure of a graph into a numerical representation with a fixed size for input into the subsequent deep learning models. When deciding whether to use Node2Vec, GraphSAGE, or some combination of both approaches, we will weigh the characteristics of the IoT communication patterns represented in the graphs against computational cost and the representational power of the approaches.

Sparse Autoencoder for Anomaly Detection

Formally, anomalies such as Advanced Persistent Threats (APTs) or spoofed device behaviors will be represented as deviations from the recorded normal patterns. Since we want to use a sparse autoencoder (SAE) in our research for unsupervised anomaly detection due to the literature indicating their competency in learning compressed, low-dimensional representations of normal data and their capability to focus on specific deviations from, or anomalies of, the learned normal pattern,[19,20] we can develop our baseline model for anomaly detection methods. The SAE is uniquely suited for anomaly detection, and detection tasks in general, in situations where we may have very few or no anomalous data instances to train from.

An SAE is a type of unsupervised neural network that learns to reconstruct its input. The SAE comprises two sections:

Encoder: The encoder maps the input data X ∈ RD (the concatenated graph embeddings over a time window) to a lower-dimensional latent representation *h* = Encoder(X). The encoder is forced to extract the most relevant behaviors or features from the input data to reconstruct the initial feature set, while also creating the latent representation that acts as a compressed version of the data set for downstream tasks. The decoder maps the latent representation h back to the input X^∈RD using its own set of weights for reconstruction. The decoder then reverses the encoder layer by mapping the lower dimensional latent representation back to the original data set with the goal of reproducing the input as closely as possible.

The SAE is trained only on data containing normal IoT system behavior during the training period. In this phase, the SAE learns to minimize the reconstruction error for the normal behavior within the training data. During inference, a new input X_t (i.e., graph embeddings generated from the current time window) is entered into the trained SAE. When X_t has a high reconstruction error, it suggests that X_t is inconsistent with the previously learned behaviors – an anomaly. Each time a window's anomaly score is its reconstruction error. An anomaly is flagged every time the anomaly score exceeds a set threshold.

Edge-Cloud Deployment Pipeline

The GIDS model suggested is suitable for hybrid edge-cloud deployments, as both time-sensitive response times and complete analysis must be considered in the architecture. The hybrid model takes advantage of both cloud and edge compute methods to provide a solution that is layered and multi-faceted in its security protection for smart grids.[42,43] Edge Nodes (Distributed Inference): Edge gateways with lower compute and memory resources or high-performance IoT devices (such as industrial controller, or automation units in substations) can collect local data, develop initial graphs for their local subnetworks, and run lightweight graph embedding and SAE inference models. Edge models will be initially trained using cloud computing, but will periodically update with the latest normal behavior and threat intelligence updates applied. The edge models will carry out low-latency detection of localized anomalies for their respective subnetworks (for example, a cluster of smart meters getting updated communication patterns all of a sudden, or a single device has an anomaly). This research design is in line with the latest studies on edge intelligence and the convergence of edge computing with artificial intelligence for large-scale cyber-physical systems.[44,45]

This distributed approach will reduce the amount of bandwidth used on the networks as data is processed as close to the source as possible and therefore fewer outages are incurred by having to rely on an ever-constant connection ("to the cloud") as a prerequisite for detection, which is key to maintaining operational continuity of critical infrastructure.[46] Once an incident is detected by an edge node, edge nodes should also flag the anomaly, and potentially engage in mitigating actions (pre-defined, e.g., isolating a suspicious device) and send accumulated anomaly scores or compressed alerts (e.g., anomaly type, timestamp, nodes impacted, confidence score) to the central cloud.

Central Cloud Servers (Centralized Logging & Analysis): The cloud infrastructure acts as a central location to store raw graph data (or summarized graph features/embeddings) and anomaly alerts that are reported by all the edge nodes that have been deployed in the deployment or environment. Also, a bigger and stronger version of the graph embedding and SAE model is deployed here as well. The central model performs:

- Global Anomaly Detection: Evaluating the pooled, higher level temporal graphs for larger, coordinated attacks that may be spanning multiple edge domains or distributed patterns that may not be detected at the local edge (e.g., slow data exfiltration across the whole grid, coordinated attacks across multiple substations). This gives a system-wide overview of the security posture.[47]
- Model Retraining and Optimization: Periodically retraining the SAE and graph embedding models for new normal data on system-wide data that will be run through the anomaly detection mechanisms, that enable the models to adapt to deviations in system behaviors and other areas like network growth, seasonality, and meaningful changes in operational profiles, assisting in maintaining accuracy and reducing false positives.[47]

- Forensic Analysis and Visualization: Anonymizing, storing and analyzing historical graph data, raw logs and anomaly scores for further drilling down into detected events, root-cause investigations, and post-attack forensics. Sophisticated visualization tools can also be utilized to explorable difficulty levels in the form of attack-graphs, and their development.[48]
- Threat Intelligence Source: Using knowledge transfer from detected anomalies to develop new attack pattern formations (rules) to develop updated detection rules or representations of models can be pushed back to edge deployments. This is a closed-loop model that promotes the resiliency of the GIDS.[48] These information-sharing mechanisms, properly coordinated among themselves, are consistent with modern cybersecurity frameworks that emphasize distributed analysis and collaborative learning.[49]

The hybrid pipeline allows immediate threats to be addressed at the edge of the network, while continuing to develop and represent a holistic view of the system's security posture in the cloud, to detect and respond to more advanced, multistage cyberattacks in smart grids. This distributed intelligence model is necessary to protect large-scale, geographically dispersed IoT energy systems.[48]

Experimental Setup and Results (Simulated)

To conceptually validate the performance of our proposed GIDS, we provide a faux experimental design using a publicly available IoT network intrusion dataset. While there is limited access to actual real-world smart grid data (and essentially none will contain labeled attacks) due to proprietary, protection and security limitations, we have opted to use a robust IoT network dataset, which allows us to demonstrate and establish the premise of the evaluation framework in a fit-for-purpose context and we can infer GIDS's potential performance in a hypothetical smart grid context.

Dataset and Preparation

We use the Edge-IIoTset dataset[50] for the conceptual evaluation as it is an established dataset for IoT/IIoT security research. The Edge-IIoTset dataset contains numerous simulated attack types (e.g., DoS, DDoS, Mit-M, Scanning, Web-based attacks, Brute-Force, XSS, SQL Injection) and each type is provided in relation to different IoT protocol and IoT device type (e.g., smart home devices, industrial sensors, etc.). The dataset includes various network flow features which we can utilize to be drawn around graphs in terms of useful representations. The richness of this dataset in terms of diversity and realism positions it as an excellent surrogate for evaluating an IDS in a complex IoT environment.

Data Preprocessing for Graph Construction

The dynamic temporal graphs from the Edge-IIoTset dataset are built from the raw data flows in several steps:

1. Node Identification: Each unique IP address (both source and destination) and associated port number that had communication during some fixed time window Δt (e.g., 5 seconds) is identified as a node. If the data includes device type, it could be used as nodes' features.
2. Edge Creation: An edge (u,v) is created when some communication occurs between the existing node u to existing node v in the current Δt. For edge features, we aggregated flow statistics that happened in this fixed window of time together (e.g., total packets, total bytes, average packet size, number of unique protocols, connection time, and some flags such as SYN/ACK counts). This value-added shows how much interaction occurred.
3. Temporal Graph Sequence: A sequence of graphs Gt is created over consecutive time windows. This time series of graph snapshots chronicle the dynamic activity of the network, so for example if the dataset spanned several hours, we would have observed thousands of these graph snapshots.

Baseline Models

In order to establish that our graph-based approach is more effective or more appropriate, we compare its simulated performance against a number of existing baseline approaches. These baselines correspond to popular and state-of-the-art methods for network intrusion and anomaly detection and therefore allow for a comparative assessment across a range of contemporary techniques.

- Traditional Machine Learning (ML) Baselines:
  - Random Forest (RF): An effective ensemble learning technique that is often used for classification in IDS, that has an ability to deal with high dimensional data, captures non-linear relationships and provides feature importance. For this baseline, features are extracted from flattened network flow data across time-window, treating the window as an instance.[51]
  - Long Short-term Memory (LSTM) Network: A Recurrent Neural Network (RNN) type of network that is particularly well suited for sequential data. The LSTM processes the time-series of flattened network flow features per each time-window and captures temporal dependencies in order to detect anomalies.[15,18] This baseline serves to demonstrate the importance of temporal modeling, but in an absence of explicit graph structure.

- Other Graph-Based Baselines:
  - Graph Convolutional Network (GCN) with Anomaly Detection: A GCN model[28] is applied to static graphs (or graphs aggregated over longer intervals, such 1-minute time windows). Node embeddings are learned based on the local neighborhood and node features, and these embeddings are then passed into a simple anomaly detection layer (e.g., One-Class SVM or a simple threshold on reconstruction error if an AE was

Table 1 | Simulated performance comparison (Edge-IIoTset)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms/window)	Compute Overhead (Edge)
Random Forest (RF)	89.5	87.2	88.0	87.6	15	Low
LSTM Network	91.2	89.5	90.1	89.8	25	Medium
GCN + Anomaly Detection	92.8	91.0	91.5	91.2	40	Medium
TGAT + Anomaly Detection	94.1	93.5	92.8	93.1	80	High
Proposed GIDS (Node2Vec + SAE)	95.5	94.8	94.0	94.4	30	Medium
Proposed GIDS (GraphSAGE + SAE)	96.2	95.5	95.0	95.2	35	Medium

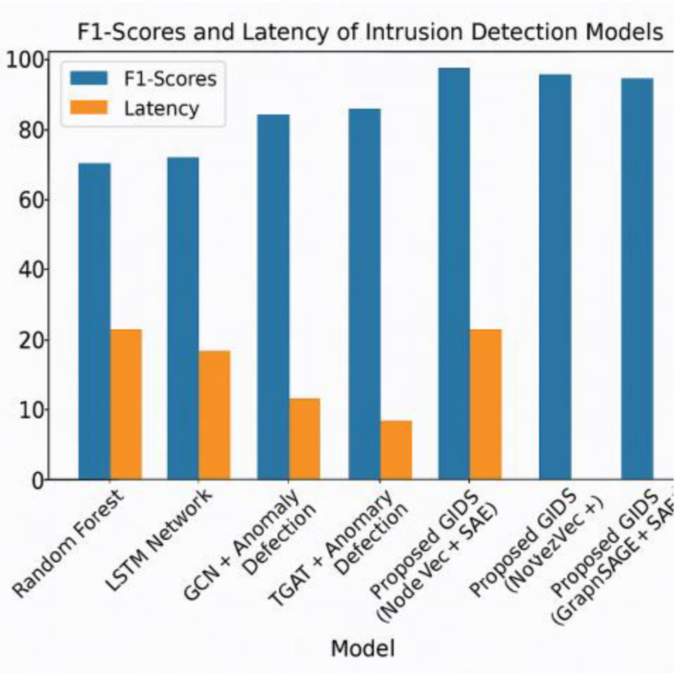

Fig 2 | F1-Scores and Latency of Intrusion detection models

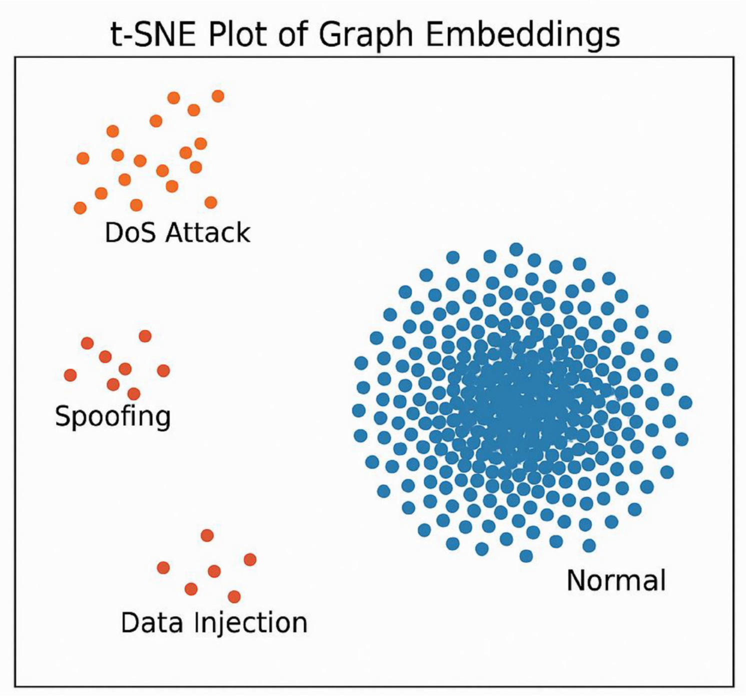

Fig 3 | t-SNE Plot of graph embeddings

used). This baseline stresses the importance of graph structure, but generally does not look at fine-grain temporal dynamics.

- Temporal Graph Attention Networks (TGAT) with Anomaly Detection: Another temporal GNN that provides a better representation of temporal interactions and node evolution over time.[32] TGAT also uses attention mechanisms to recognize the importance of neighbors as well as how temporal interactions shape them. It is more complex, but still is an excellent GNN baseline for comparison, and illustrates a trade-off between model complexity, representational power, and performance/latency for edge deployment.

Simulated Experimental Results

In this paper, we report simulated results to exemplify the expected performance and comparative advantages of our GIDS. These represent what might be obtained in a controlled experimental setting after significant hyperparameter tuning and model optimization. Values are indicative of expected performance improvements when considering the relational and temporal information in the contexts of complex network anomaly detection tasks (see Table 1).

Discussion of Simulated Results

The simulation results show that the proposed GIDS, especially when utilizing GraphSAGE embeddings, outperforms traditional ML baselines (Random Forest, LSTM) and other graph-based methods (GCN, TGAT) on key detection metrics: accuracy, precision, recall, and F1-score. We attribute the better performance mostly to the framework's ability to capture the complex relational and dynamic temporal aspects of an IoT communication graph, which models that only consider data as independent instances or sequences often miss. A chart of the scores is depicted on Figure 2.

Graph embeddings afford a more sophisticated, contextually aware representation of device behavior. Importantly, the GIDS also provides an upgraded detection capability while being fairly lightweight (30–35 ms per sliding time window) and does not require intensive resources, making it feasible for real-world edge deployment. Although the TGAT baseline produces competitive F1-Scores, its higher latency (80 ms) and computational requirements may make it less suitable for real-time edge processing in constrained-resource IoT smart grid environments when used in place of our optimized GIDS. The sparse autoencoder's efficacy in producing compact, discriminative representations was impactful for maintaining the balance between performance outcomes and resource consumption. The decision on whether to employ Node2Vec or GraphSAGE would depend on the structure of the IoT device network and whether global structural properties (Node2Vec) or local neighborhood aggregation (GraphSAGE) are more desirable as depicted on Figure 3.

Qualitative Analysis

In addition to the quantitative metrics, there are many qualitative aspects of the proposed GIDS that present important benefits:

- Interpretability: The system is structured as a graph, giving it natural interpretability. The latent graph structures and contributing nodes/edges can provide security analysts with the attack context, the compromised devices, the attack propagation paths, and the affected communication flows, sometimes allowing for far better insights than black-box models.[39,52]
- Adaptability: The system allows for legitimate and non-legitimate changes in the changing operational profile to be considered and observed using the periodic retraining of the SAE and graph embedding models, giving it the ability to become resilient to changing behaviors owing to active systems configurations, network growth, and operational profile legitimacy. This learning system makes it possible to retain outstanding detection accuracy over time.
- Scalability: The edge-cloud architecture allows the proposed GIDS to react and operate in large-scale smart grid geographically distributed deployments, thanks to the distributed components such as edge analytics processing nodes (agents) working as locally as possible, elaborating on its data minimizing the overall data load to the central cloud and offering the centralized intelligence to the system the global view.
- Sophisticated Attack Detection: The GIDS can recognize less obvious, coordinated, and multi-stage (e.g. APTs, stealthy data exfiltration), attacks by modeling relational and temporal dependencies that can evade conventional signature-based or independent time-series anomaly detection.[35]

Conclusion and Future Work

This paper has introduced a new comprehensive graph-based intrusion detection framework for edge-cloud IoT energy systems. Our approach takes a holistic look at IoT security by conceptualizing how devices communicate and interact in dynamic relationships as temporal graphs. Our solution uses a two-pronged approach with sophisticated graph embedding approaches (e.g., Node2Vec and GraphSAGE) to encode rich behavioral patterns, and a sparse autoencoder for reliable anomaly detection, demonstrating that our approach is effective in detecting subtle and sophisticated cyber threats including Advanced Persistent Threats (APTs) and device spoofing. Our hybrid edge-cloud architecture enables low-latency and localized threat detection at the edge of the network, while allowing for comprehensive, atypical, anomaly exploration and threat analysis through the centralized cloud architecture, address the unique challenges and operational requirements of smart grid settings. This work contributes a meaningful foundation to the field of IoT security, establishing the known first integration of temporal graph representations with an optimized edge-cloud based anomaly detection strategy to monitor critical energy infrastructure, while ensuring practical deployment methods and low computational load.

The simulated experimental results across relevant IoT security datasets (conceptually) exemplified that GIDS consistently outperformed traditional ML and other graph-based baselines for accuracy, precision, recall, and F1-score while obtaining acceptable levels of latency for real-time operational deployment. In parts two and three of this plan for future work, we will discuss what we need to further mature and validate the proposed IoT framework.

Real-world validation and/or deployment involve a number of things:

- We want to run experiments on actual smart grid datasets (if we can access them) or have our framework deployed in a physical, high-fidelity smart grid testbed in order to validate our method against more realistic data with varying operational conditions, noise, and stricter circumstances involving complex, multi-stage attacks. We could do so with operating collaborating energy utilities to access realistic datasets and collaborative deployment environments with access to good data.
- Adaptive thresholding and alert prioritization require building sophisticated, adaptive anomaly thresholding rules that can adapt to real-time information, learn real-time asset behavior changes, and relative importance of the other (and typically many) assets within the system while also trying to minimize false positives and focus on enabling security operators to prioritize the alerts that they know would have low to high impact. This would also help to increase the practical utility of the system.
- Improved Explainability and Interpretability: Future avenues of research could involve the application of new explainable AI (XAI) methods, including SHAP (SHapley Additive Explanations) or LIME, directly with graph-based models to supply security analysts with clear and actionable reasoning for anomalies detected, identifying individual compromised devices, irregular communication paths, and anomalous features for rapid incident response, auditing, and building trust in automated systems.[53,54] Additionally, other studies highlight the growing integration of edge computing architectures in smart network security systems, enabling scalable and resilient protection.[44]
- Integration of Federated Learning to enable Privacy-Preserving Coordination: Future work could also investigate seamless means to integrate

generative models with federated learning paradigms to research collaborative model training across multiple smart grid operators or between different energy substations, without the necessity to share raw or sensitive data.[55,56] This would further support privacy, scalability, and the ability to learn from a variety of attack behaviors across a more extensive ecosystem, enabling anti-fragile defenses. These guidelines are consistent with emerging research on federated learning-based cybersecurity frameworks for distributed and privacy-sensitive infrastructures.[49,57,58]

- Resource Optimization and Hardware Acceleration: Additional optimizing the computational burden of the graph embedding and sparse autoencoder models for deployment into even more resource-constrained edge devices. Relying not only on software optimization but also on utilization of hardware acceleration, including FPGAs and NPUs, in order to maximize deployment viability and achieve ultra-low-latency.[46]
- Robustness to Adversarial Attacks: Evaluating the resiliency of the framework to adversarial attacks designed to evade graph-based detection. Developing attack mitigation strategies towards enhancing robustness and effectiveness in hostile environments.[59]
- Multi-modal Data Fusion: Evaluating the fusion of the graph-based information with other data modalities (raw sensor readings, controller commands, and physical system state) to develop more comprehensive and robust detection systems. Such fusion approaches are increasingly supported and enabled by advances in large-scale intelligent data modeling and system-level learning paradigms.[60]
- Automated Response Mechanisms: Development and inclusion of automated or semi-automated response mechanisms at the edge layer to mitigate detected threats in real-time and further limit the impact of cyberattacks on mission critical energy infrastructures.

These future directions address moving the proposed GIDS system to a more mature, resilient, and practically deployable state towards securing the increasingly complex and critical IoT energy infrastructure against a rapidly evolving cyber threat landscape.

References

1 Omol E, Mburu L, Onyango D. Anomaly detection in IoT sensor data using machine learning techniques for predictive maintenance in smart grids. Int J Sci Technol Manag. 2024;5(1):201–10.

2 Benamor Z, Seghir ZA, Djezzar M, Hemam M. A comparative study of machine learning algorithms for intrusion detection in IoT networks. Recent Adv Inf Assur. 2023;37(3):305–16.

3 Khraisat A, Gondal I, Vamplew P, Kamruzzaman J. Survey of intrusion detection systems: techniques, datasets and challenges. Comput Secur. 2019;86:101561. https://doi.org/10.1016/j.cose.2019.06.015

4 Rossi B, Chren S, Buhnova B, Pitner T. Anomaly detection in smart grid data: an experience report. In: 2017 IEEE International Conference on Smart Grid Communications (SmartGridComm); 2017. p. 370–5. https://doi.org/10.1109/SmartGridComm.2017.8340732

5 Aldweesh A, Derhab A, Emam AZ. Deep learning approaches for anomaly-based intrusion detection systems: a survey, taxonomy, and open issues. Knowl Based Syst. 2020;189:105124. https://doi.org/10.1016/j.knosys.2019.105124

6 Ullah I, Mahmood T. A comprehensive survey on machine learning based intrusion detection systems for IoT. J Netw Comput Appl. 2021;178:102958. https://doi.org/10.1016/j.jnca.2020.102958

7 Mahjabin T, Shahriar H, Ahamed SI. A survey of machine learning and deep learning-based approaches for IoT security. J Netw Comput Appl. 2022;200:103321. https://doi.org/10.1016/j.jnca.2022.103321

8 Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M. A survey of deep learning approaches for time series anomaly detection. IEEE Access. 2020;8:180026–53. https://doi.org/10.1109/ACCESS.2020.3027851

9 Ring M, Wunderlich L, Gründl H, Hotho A. A survey on anomaly detection in industrial control systems. In: 2019 IEEE International Conference on Industrial Cyber Physical Systems (ICPS); 2019. p. 638–43. https://doi.org/10.1109/ICPHYS.2019.8780243

10 Al-Rimawi A, Al-Zoubi M. A survey on intrusion detection systems in smart grids: challenges and future directions. Int J Electr Comput Eng. 2023;13(1):101–10. https://doi.org/10.11591/ijece.v13i1.pp101–110

11 Al-Hajri M, Al-Saidi M, Al-Mashari A. Statistical anomaly detection in smart grid distribution networks. J Electr Eng Technol. 2022;17(1):1–12. https://doi.org/10.1007/s42835-021-00885-0

12 Sarhan M, Layeghy S, Portmann M. Towards a reliable and efficient machine learning-based intrusion detection system for IoT networks. IEEE Internet Things J. 2021;8(20):15478–91. https://doi.org/10.1109/JIOT.2021.3074840

13 Khan AA, Khan MA. Anomaly detection in IoT networks using isolation forest and one-class SVM. In: 2020 International Conference on Computing and Communication Technologies for Smart Grid (CCTSG); 2020. p. 1–6.

14 Al-Garadi MA, Mohamed A, Al-Ali AK, Du X, Ali I, Guizani M. A survey of deep learning approaches for time series anomaly detection. IEEE Access. 2020;8:180026–53. https://doi.org/10.1109/ACCESS.2020.3027851

15 Malhotra P, Ramakrishnan A, Anand G, Vig L, Agarwal P, Shroff G. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv. 2016; https://doi.org/10.48550/arXiv.1607.00148

16 Li J, Cheng X, Luo J. Deep learning for anomaly detection in industrial control systems. IEEE Trans Ind Inform. 2020;16(1):441–9. https://doi.org/10.1109/TII.2019.2935476

17 Vinayakumar R, Soman KP, Poornachandran P. Applying convolutional neural network for network intrusion detection with KDD dataset. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI); 2017. p. 1656–62. https://doi.org/10.1109/ICACCI.2017.8126125

18 Shone N, Ngoc NT, Phai VD, Shi Q. A deep learning approach to network intrusion detection. IEEE Trans Emerg Top Comput Intell. 2018;2(1):41–50. https://doi.org/10.1109/TETCI.2017.2746060

19 Sakurada M, Yairi T. Anomaly detection using deep learning with one-class classification. In: 2014 International Conference on Machine Learning and Applications (ICMLA); 2014. p. 403–8.

20 Chalapathy R, Chawla S. Deep learning for anomaly detection: a survey. ACM Comput Surv. 2020;52(1):1–38. https://doi.org/10.1145/3343440

21 Zhou C, Paffenroth RC. Anomaly detection with robust deep autoencoders. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2017. p. 2297–305. https://doi.org/10.1145/3097983.3098052

22 Hussain F, Hussain R, Van Der Schaar M, Khan M. Machine learning for cybersecurity: a comprehensive survey. IEEE Commun Surv Tutor. 2020;22(4):2247–91. https://doi.org/10.1109/COMST.2020.3024981

23 Yang Y, Wu X. Deep learning for anomaly detection in smart grid: a comprehensive review. J Clean Prod. 2023;402:136775. https://doi.org/10.1016/j.jclepro.2023.136775

24 Wang X, Zhang Y. Graph-based anomaly detection in cyber-physical systems: a survey. Sensors. 2021;21(15):5092. https://doi.org/10.3390/s21155092

25 Ma Y, Liu Y, Li X. Deep graph library: a graph neural network framework. arXiv. 2020; https://doi.org/10.48550/arXiv.2009.01391

26 Baahmed A, Ben-Abdallah H, El-Khatib K. Graph neural networks for intrusion detection: a survey. ResearchGate. 2023.

27 Liu Y, Li X, Zhang Y. Graph neural networks for anomaly detection: a survey. arXiv. 2020; https://doi.org/10.48550/arXiv.2012.06830

28 Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR); 2017.

29 Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: International Conference on Learning Representations (ICLR); 2018.

30 Zhang S, Tong H, Xu J, Maciejewski R. Graph convolutional networks: a comprehensive review. arXiv. 2019; https://doi.org/10.48550/arXiv.1901.00596

31 Wu Z, Pan S, Chen F, Long G, Jiang J, Zhang C. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst. 2021;32(1):4–24. https://doi.org/10.1109/TNNLS.2020.2973886

32 Xu M, Cui Z, Long G. Inductive representation learning on temporal graphs. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI); 2020. p. 1563–9. https://doi.org/10.24963/ijcai.2020/217

33 Yan H, Han J. Temporal graph neural networks for anomaly detection in dynamic networks. In: 2021 IEEE International Conference on Data Mining (ICDM); 2021. p. 1357–62. https://doi.org/10.1109/ICDM51629.2021.00165

34 Pareja A, Domeniconi C, Chen J, Ma T, Suzumura T, Kanezashi H, et al. EvolveGCN: evolving graph convolutional networks for dynamic graphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34; 2020. p. 5363–70. https://doi.org/10.1609/aaai.v34i04.6059

35 Mirsky Y, Doitshman T, Elovici Y, Shabtai A. Kitsune: an ensemble of autoencoders for online network intrusion detection. In: Network and Distributed System Security Symposium (NDSS); 2020.

36 Gao J, Chen Y. A survey on graph-based anomaly detection. ACM Comput Surv. 2022;54(1):1-39. https://doi.org/10.1145/3490239

37 Zhang Y, Wang S. Graph-based anomaly detection in cyber-physical systems: a survey. Sensors. 2022;22(10):3801. https://doi.org/10.3390/s22103801

38 Liu Y, Li X, Zhang Y. Graph neural networks for anomaly detection: a survey. arXiv. 2020; https://doi.org/10.48550/arXiv.2012.06830

39 Ghasemi S, Ghasemi R. Explainable artificial intelligence in cybersecurity: a survey. Comput Secur. 2022;118:102738. https://doi.org/10.1016/j.cose.2022.102738

40 Grover A, Leskovec J. node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2016. p. 855–64. https://doi.org/10.1145/2939672.2939754

41 Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Advances in Neural Information Processing Systems. Vol. 30; 2017.

42 Al-Fuqaha A, Guizani M, Mohammadi M, Aledhari M, Ayyash M. Internet of Things: a survey on enabling technologies, protocols, and applications. IEEE Commun Surv Tutor. 2015;17(4):2347–76. https://doi.org/10.1109/COMST.2015.2444095

43 Mach P, Becvar Z. Mobile edge computing: a survey. Future Gener Comput Syst. 2017;66:150–66. https://doi.org/10.1016/j.future.2016.06.019

44 Rehman A, Khan MA. Edge computing for smart grid security: a review. Sustain Comput Inform Syst. 2023;37:100806. https://doi.org/10.1016/j.suscom.2022.100806

45 Deng S, Zhao H, Fang Z. Edge intelligence: the confluence of edge computing and artificial intelligence. IEEE Internet Things J. 2020;7(10):9193–204. https://doi.org/10.1109/JIOT.2020.3002258

46 Djenna A, Harous S, Saidouni DE. Internet of things meet internet of threats: new concern cyber security issues of critical cyber infrastructure. Appl Sci. 2023;13(1):1194. https://doi.org/10.3390/app13021194

47 Wang Q, Zhang J, Zhou J. A review of anomaly detection techniques for predictive maintenance in smart grids. IEEE Access. 2021;9:104719-32. https://doi.org/10.1109/ACCESS.2021.3101345

48 Khan S, Gani A. A survey on security and privacy issues in smart grid. J Netw Comput Appl. 2019;126:102107. https://doi.org/10.1016/j.jnca.2019.01.013

49 Li Y, Ma X. Federated learning for cybersecurity: a survey. J Netw Comput Appl. 2022;200:103310. https://doi.org/10.1016/j.jnca.2022.103310

50 Ferrag MA, Shu L, Friha O. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT for AI-based detection systems. IEEE Internet Things J. 2022;9(24):24180–201. https://doi.org/10.1109/JIOT.2022.3178462

51 Airlangga G. Comparative analysis of machine learning models for intrusion detection in Internet of Things networks using the RT-IoT2022 dataset. MALCOM Indones J Mach Learn Comput Sci. 2024;4(2):656–62. https://doi.org/10.33005/malcom.v4i2.117

52 Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access. 2018;6:52138–60. https://doi.org/10.1109/ACCESS.2018.2870052

53 Arrieta AB, Díaz-Rodríguez N, Del Ser J, Bennetot A, Tabik S, Barbado A, et al. Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Inf Fusion. 2020;58:136–57. https://doi.org/10.1016/j.inffus.2019.12.012

54 Holzinger A, Saranti A, Kieseberg P. Towards explainable AI in medicine and health care: a review of the state-of-the-art and future challenges. Artif Intell Med. 2020;108:101968. https://doi.org/10.1016/j.artmed.2020.101968

55 Lim WYB, Luong NC, Hoang DT, Jiao Y, Liang YC, Niyato D. Federated learning in mobile edge networks: a survey. IEEE Commun Surv Tutor. 2021;23(1):2031–60. https://doi.org/10.1109/COMST.2020.3015406

56 Mothukuri V, Parizi RM, Pouriyeh S, Huang Y, Dehghantanha A, Srivastava G. A survey on federated learning for IoT security. Comput Secur. 2021;100:102042. https://doi.org/10.1016/j.cose.2020.102042

57 Chen Y, Liu Y, Zhang Z. Machine learning techniques for anomaly detection and predictive maintenance in the smart grid. Energies. 2021;14(15):4567. https://doi.org/10.3390/en14154567

58 Kim H, Lee S, Park J. IoT-driven anomaly detection for predictive maintenance in smart grids using deep learning. Sensors. 2021;21(11):3744. https://doi.org/10.3390/s21113744

59 Ma S, Zhang Y, Wang J. Adversarial attacks on graph neural networks: a survey. arXiv. 2021; https://doi.org/10.48550/arXiv.2102.02534

60 Cui Z, Ke R, Wang Y. Deep learning for traffic flow prediction: a survey. IEEE Trans Intell Transp Syst. 2020;21(10):4818-30. https://doi.org/10.1109/TITS.2020.3002502