IEEE Open Journal of the Computer Society

Received 27 April 2025; revised 1 July 2025; accepted 6 July 2025. Date of publication 10 July 2025; date of current version 18 July 2025. The review of this article was arranged by Associate Editor Peng Li.

Digital Object Identifier 10.1109/OJCS.2025.3587486

Securing Industrial IoT Environments: A Fuzzy Graph Attention Network for Robust Intrusion Detection

SAFA BEN ATITALLAH [1,2] (Member, IEEE), MAHA DRISS [1,2], WADII BOULILA [1,2], AND ANIS KOUBAA [3]

[1]Robotics and Internet of Things Laboratory, Prince Sultan University, Riyadh 12435, Saudi Arabia
[2]RIADI Laboratory, National School of Computer Science, University of Manouba, Manouba 2010, Tunisia
[3]Alfaisal University, Riyadh 11533, Saudi Arabia

CORRESPONDING AUTHOR: SAFA BEN ATITALLAH (e-mail: satitallah@psu.edu.sa).

ABSTRACT The Industrial Internet of Things (IIoT) faces significant cybersecurity threats due to its ever-changing network structures, diverse data sources, and inherent uncertainties, making robust intrusion detection crucial. Conventional machine learning methods and typical Graph Neural Networks (GNNs) often struggle to capture the complexity and uncertainty in IIoT network traffic, which hampers their effectiveness in detecting intrusions. To address these limitations, we propose the Fuzzy Graph Attention Network (FGATN), a novel intrusion detection framework that fuses fuzzy logic, graph attention mechanisms, and GNNs to deliver high accuracy and robustness in IIoT environments. FGATN introduces three core innovations: (1) fuzzy membership functions to explicitly model uncertainty and imprecision in traffic features; (2) fuzzy similarity-based graph construction with adaptive edge pruning to build meaningful graph topologies that reflect real-world communication patterns; and (3) an attention-guided fuzzy graph convolution mechanism that dynamically prioritizes reliable and task-relevant neighbors during message passing. We evaluate FGATN on three public intrusion datasets, Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022, achieving accuracies of 99.07%, 99.20%, and 99.05%, respectively. It consistently outperforms state-of-the-art GNN (GCN, GraphSAGE, FGCN) and deep learning models (DNN, GRU, RobustCBL). Ablation studies confirm the essential roles of both fuzzy logic and attention mechanisms in boosting detection accuracy. Furthermore, FGATN demonstrates strong scalability, maintaining high performance across a range of varying graph sizes. These results highlight FGATN as a robust and scalable solution for next-generation IIoT intrusion detection systems.

INDEX TERMS Attention mechanisms, fuzzy logic, GNNs, industrial IoT, intrusion detection.

I. INTRODUCTION

The emergence of the Industrial Internet of Things (IIoT) has revolutionized industrial systems and enabled advanced real-time monitoring and data collection across a wide range of devices and environments [1], [2], [3]. However, the dynamic and complex nature of IIoT ecosystems makes them vulnerable to cyber-attacks and threats [4]. Recently, Machine Learning (ML) methods have been widely investigated for intrusion detection in both IoT and IIoT settings [5], [6], [7], [8]. Although conventional algorithms excel at processing large datasets and rapidly detecting current attacks, they often fail to handle the inherent complexity of IIoT systems.

Graph Neural Networks (GNNs) have emerged as a promising option to improve intrusion detection capabilities [9]. Unlike conventional ML models that treat samples independently, GNNs exploit the relational nature of data by learning from the structure of graphs, where nodes represent entities such as devices or data packets, and edges capture interactions or similarities [10], [11]. This graph-centric view enables GNNs to detect contextual patterns that are otherwise obscured in flat data representations. Despite their promise, the application of standard GNNs in IIoT settings remains challenging due to the non-stationary and noisy nature of industrial environments [12]. IIoT networks exhibit

continuously evolving topologies and heterogeneous data flows, which introduce significant ambiguity and limit the robustness of typical GNN architectures. Consequently, there is a pressing need for more adaptive and uncertainty-aware GNN frameworks that can effectively manage these complexities.

To address this gap, fuzzy logic offers a powerful mathematical foundation for handling uncertainty and imprecision in IIoT data [13]. Unlike binary logic, which imposes hard boundaries, fuzzy logic assigns membership values between 0 and 1, reflecting the degree of confidence in a relationship or observation (e.g., how likely a flow is to be malicious). When this fuzzy reasoning is integrated with GNNs, it enhances the model's ability to represent soft, nuanced relationships that mirror real-world ambiguity. Moreover, attention mechanisms complement this by learning to emphasize the most informative neighbors during graph convolution [14]. Attention weights guide the model to focus on connections that are most relevant for accurate classification. When combined with fuzzy similarity measures, the result is a dual-guided learning process that (1) down-weights unreliable or noisy edges based on semantic uncertainty, and (2) up-weights contextually important interactions for task-specific learning.

This article introduces the Fuzzy Graph Attention Network (FGATN), the first DL framework to integrate fuzzy-set theory with graph attention mechanisms for robust intrusion detection in IIoT environments. Fuzzy membership functions are used to explicitly model the uncertainty, noise, and imprecision that characterize real-world IIoT traffic, allowing the system to express degrees of semantic similarity between flows. Simultaneously, graph attention mechanisms adaptively prioritize the most informative neighbors during message passing, focusing the model's capacity on critical, context-relevant relationships. By embedding fuzzy similarity directly into attention-weighted graph convolutions, FGATN learns both the reliability and the relevance of each connection, enabling nuanced reasoning over noisy graph structures. Our objective is to achieve high classification accuracy and F1-score across three diverse IIoT benchmarks, Edge-IIoT, ToN-IoT, and CIC-IoMT-2024, demonstrating enhanced robustness to uncertainty and evolving traffic patterns in complex industrial systems. Our experimental evaluation of FGATN shows that it outperforms existing intrusion detection approaches. Through extensive experimentation with three distinct datasets, the proposed FGATN consistently achieved excellent results in terms of accuracy, precision, recall, F1-score, and Matthews Correlation Coefficient (MCC).

In summary, the proposed approach includes the following main contributions:

1) Incorporating fuzzy logic-based techniques into GNNs to effectively manage imprecise data in IIoT settings.
2) Using attention methods to selectively consider critical nodes and edges in the network, thereby improving the accuracy and efficiency of intrusion detection systems.
3) Validating the proposed approach using three datasets: Edge-IIoTSet, WSN-DS, and CIC-Malmem2022.
4) Performing a comprehensive performance analysis using a diverse set of evaluation metrics to accurately assess the capabilities of the proposed model.
5) Conducting a performance comparison between our approach and other state-of-the-art methods, as well as conventional GNN methods, using the same datasets.

The rest of the article is structured as follows. Section II reviews the proposed intrusion detection systems. Section III introduces our approach, which combines the power of GNN, fuzzy logic, and attention mechanisms. Section IV provides an overview of the datasets used. Section V presents the results of our experimental analysis. In Section VI, we discuss the strengths of the proposed approach and identify its limitations. Finally, our study is concluded in Section VII.

II. RELATED WORK

The rise in cyber threats to IIoT networks demands advanced intrusion detection systems [15], [16]. Although DL and GNNs have been explored, their limitations in addressing IIoT complexities persist [17], [18]. DL approaches struggle with IIoT's topological and uncertain data. Abou El Houda et al. [19] proposed a filtered DL model to handle heterogeneous IIoT data, but its flat feature-based approach overlooks network interactions, risking misdetection of sophisticated attacks. Zang et al. [20] employed federated learning with transfer learning and rank aggregation for anomaly detection. Despite distributed efficiency, the model's neglect of dynamic topologies limits its adaptability to evolving IIoT networks. Kandhro et al. [21] developed DNN, RNN, and CNN models to minimize false positives. However, their inability to capture device interconnections restricts performance in diverse and interconnected IIoT environments.

On the other hand, GNNs offer promise by modeling network topologies [9], [22]. Caville et al. [23] used self-supervised GNNs to generate embeddings from traffic graphs, but their reliance on source-destination endpoints oversimplifies IIoT's complex relationships, reducing detection precision. Lo et al. [24] introduced E-GraphSAGE, which incorporates edge features for IoT attack detection. However, its lack of attention mechanisms fails to prioritize critical network regions, limiting accuracy. Chang et al. [25] proposed E-GraphSAGE and E-ResGAT with residual learning, but their static graph structures struggle with IIoT's uncertain, dynamic data. Altaf et al. [26] emphasized multi-edge features in GNNs, yet their omission of fuzzy logic hampers handling of imprecise IIoT data, critical for robust detection. These studies reveal significant gaps:

1) *Oversimplified Graph Representations:* Most GNNs rely on basic source-destination graphs, losing nuanced IIoT network interactions and reducing detection accuracy [23], [25].
2) *Limited Attention Mechanisms:* Few models employ attention techniques, missing opportunities to focus on critical nodes and edges, which degrades precision in complex networks [24].

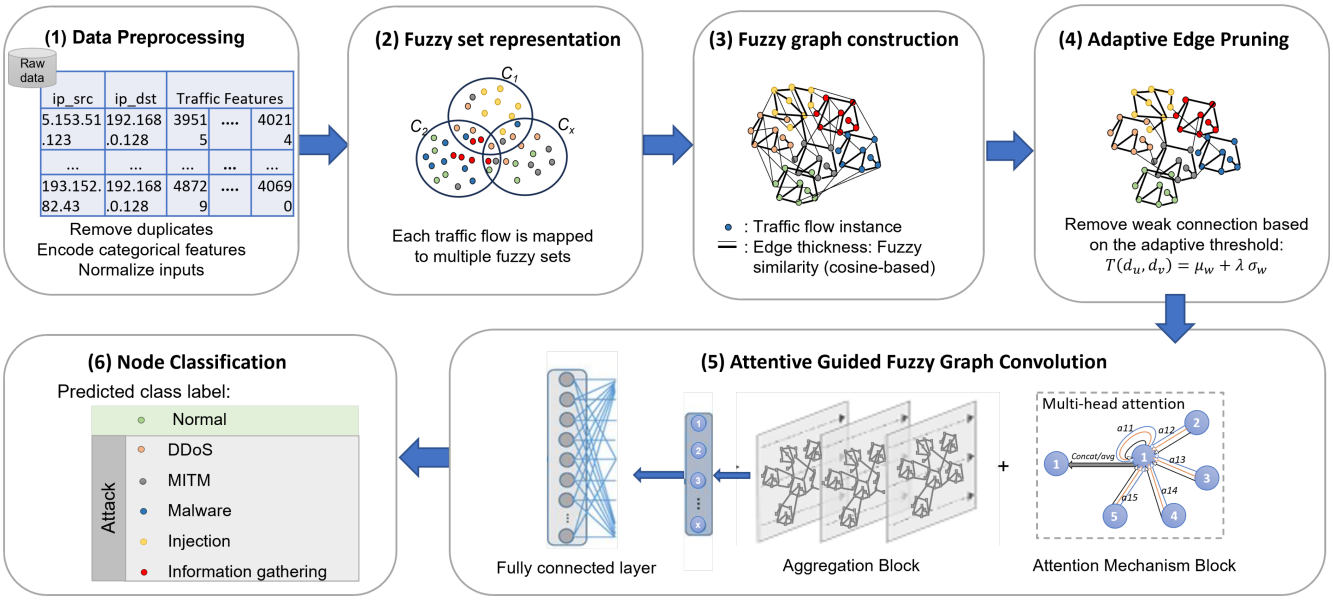


FIGURE 1. End-to-end workflow of the proposed FGATN for Industrial-IoT intrusion detection.

3) *Absence of Fuzzy Logic:* Existing approaches neglect fuzzy logic, essential for managing IIoT's inherent uncertainty, limiting their robustness [26].

In contrast, our proposed FGATN overcomes these limitations through a synergistic integration of fuzzy logic, attention mechanisms, and GNNs. Specifically, fuzzy logic is employed to construct uncertainty-aware graphs that capture the inherent imprecision in IIoT traffic; attention mechanisms dynamically prioritize the most informative and context-relevant nodes and edges; and GNNs are leveraged to model complex topological structures and relational patterns. This integrated approach empowers FGATN to achieve robust, accurate, and interpretable intrusion detection.

III. PROPOSED APPROACH

The primary objective of the proposed approach is to enhance intrusion detection performance in IIoT environments by addressing the inherent challenges of data uncertainty, noise, and complex interdependencies in network flows. To achieve this, we introduce a new framework, Fuzzy Graph Attention Network (FGATN), that combines the strengths of GNNs, fuzzy logic, and attention mechanisms. This hybrid design enables the model to capture both the structural relationships between traffic flows and the semantic uncertainty present in real-world IIoT data. Specifically, fuzzy logic is employed to represent imprecise and noisy input features using interpretable membership functions, allowing the model to reason about partial belonging and uncertainty. The graph-based learning is leveraged to model communication dependencies by representing each flow as a node and its semantic similarities as edges. Attention mechanisms are integrated to dynamically prioritize information from the most relevant neighbors to focus learning on discriminative patterns and filter out irrelevant or misleading connections.

By fusing these components, FGATN performs message passing that is both uncertainty-aware and task-aware. This dual awareness enables robust representation learning in noisy environments, improves generalization to unseen attack patterns, and supports interpretable decision-making. The following subsections provide a detailed description of each component in the FGATN pipeline, including preprocessing, fuzzy transformation, graph construction, adaptive edge pruning, attentive fuzzy graph convolution, and final classification. A high-level visual summary of the full pipeline is provided in Fig. 1.

A. FGATN: INTEGRATING FUZZY LOGIC AND ATTENTION FOR NODE CLASSIFICATION

1) DATA PREPROCESSING

In this initial step, raw traffic flow data is preprocessed to be prepared for graph construction. The preprocessing pipeline involves several key operations, including the removal of duplicate records to eliminate redundant flows, handling of missing values, elimination of irrelevant or constant-value feature columns that do not contribute meaningful information to the learning process, and normalization of numerical features using z-score scaling to ensure that all input features contribute proportionally during training. These preprocessing steps are essential to reduce noise and enhance the quality of the data.

2) FUZZY SET REPRESENTATION

To capture the inherent variability in IIoT network traffic, each traffic flow is mapped to one or more fuzzy sets. This transformation enables the soft classification of node attributes, allowing a traffic flow to belong to multiple categories with varying degrees of confidence. For each numerical feature, we define

fuzzy sets using triangular membership functions. Each function is parameterized by three values a, b, and c, representing the lower bound, peak (full membership), and upper bound, respectively. The membership function is defined in (1):

$$\mu_A(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ \frac{c-x}{c-b}, & b < x < c \\ 0, & x \geq c \end{cases} \tag{1}$$

This formulation assigns to each traffic flow, based on its feature value x, a degree of membership $\mu_A(x) \in [0, 1]$ in a fuzzy set A. Compared to alternatives such as Gaussian or trapezoidal membership functions, triangular functions strike a favorable balance between simplicity, computational speed, and adequate expressiveness. By applying this fuzzy representation, each node in the graph is enriched with interpretable uncertainty-aware features, allowing downstream modules to reason about partial belonging and imprecision.

3) FUZZY GRAPH CONSTRUCTION

To capture the relational structure among traffic flows under uncertainty, we construct a fully connected fuzzy similarity graph $G = (V, E)$, where each node $v_i \in V$ represents a traffic flow, and each edge $(v_i, v_j) \in E$ is assigned a weight based on the fuzzy cosine similarity between the flows fuzzy membership vectors. The edge weight quantifies the semantic similarity between two flows by measuring the degree of overlap in their fuzzy set representations across all features. This allows the model to go beyond raw feature values and reason in the linguistic space, which is more robust to measurement noise and variation. The fuzzy cosine similarity between nodes is defined in (2):

$$\text{cosine}(f_i, f_j) = \frac{\sum_{k=1}^{d} \mu_{f_i}(x_k)\mu_{f_j}(x_k)}{\sqrt{\sum_{k=1}^{d} \mu_{f_i}^2(x_k)}\sqrt{\sum_{k=1}^{d} \mu_{f_j}^2(x_k)}} \tag{2}$$

where d is the number of node features and x_k is the k-th feature of the nodes.

4) ADAPTIVE EDGE PRUNING STRATEGY

To enhance efficiency and sparsity, we apply a dynamic pruning strategy to remove weak connections. The pruning threshold is computed using (3):

$$\tau(d_u, d_v) = \mu_w + \lambda\sigma_w \tag{3}$$

where μ_w is the mean edge weight in the graph, σ_w is the standard deviation of edge weights, λ is a scaling factor that controls the sparsity level. Edges with weights below this threshold are considered statistically insignificant and are pruned from the graph. The resulting pruned edge set is defined in (4):

$$E' = \{(u, v) \in E \mid w_{uv} > \tau(d_u, d_v)\} \tag{4}$$

This adaptive, data-driven thresholding strategy enables the model to dynamically retain only the most meaningful connections, according to the structure of each graph instance. This step helps to:

- Reduces noise by eliminating weakly correlated flows that introduce ambiguity into the learning process.
- Sharpens the signal-to-noise ratio, allowing the model to focus on the most informative relationships.
- Improves scalability by reducing the number of edges, leading to lower memory and computational costs.
- Mitigates overfitting by preventing the model from learning from irrelevant or spurious patterns, especially in highly connected graphs.

Compared to fixed and top-k pruning approaches, this method maintains better flexibility and adaptability across different traffic conditions. It ensures that the essential topological and semantic structure of the graph is preserved, while redundant or misleading connections are discarded.

5) ATTENTIVE GUIDED FUZZY GRAPH CONVOLUTION

The core of the FGATN model lies in a novel fuzzy-aware graph convolution mechanism that integrates both fuzzy similarity and attention weighting to propagate features through the graph. Each node's features are updated by aggregating information from its neighbors, guided by the product of two key coefficients: the fuzzy similarity $S_{ij}^{(l)}$, which reflects the semantic reliability of the edge, and the attention coefficient $A_{ij}^{(l)}$, which reflects its contextual importance. The node update equation for the layer $l + 1$ is defined in (5):

$$(H^{(l+1)})_i = \sum_{j \in \mathcal{N}_i} \frac{1}{Z_{ij}} S_{ij}^{(l)} A_i^{(l)} (H^{(l)})_j \tag{5}$$

where N_i denotes the neighborhood of node i, Z_{ij} is a normalization constant, and $(H^{(l)})_j$ is the feature vector of node j at layer l. The attention scores are computed with (6):

$$A_{ij}^{(l)} = \frac{exp(LR(W^T[F_i||F_j]))}{\sum_{k=1}^{n} exp(LR(W^T[F_i||F_j]))} \tag{6}$$

where F_i is the node's i features, W presents the learned weights, LR denotes the activation function and $||$ is the concatenation operation.

In each layer, the product $S_{ij}^{(l)} A_{ij}^{(l)}$ forms a two-stage filter. First, the fuzzy similarity $S_{ij}^{(l)}$ down-weights edges with high uncertainty. Then, the attention score $A_{ij}^{(l)}$ reallocates weight to the most informative of the remaining neighbors. A higher $A_{ij}^{(l)}$ maximizes the contribution of the neighbor j's feature vector $(H^{(l)})_j$, thereby shaping the node i's embedding to prioritize semantically reliable and task-discriminative inputs. This formulation allows the model to propagate signals that are contextually meaningful and structurally sound.

6) NODE CLASSIFICATION

In the final stage of the FGATN framework, the learned node embeddings, obtained after multiple layers of attentive fuzzy

graph convolution, are passed through a fully connected output layer followed by a Softmax activation function. This layer transforms the final embedding of each node into a probability distribution over the predefined set of attack categories. Each node, corresponding to a traffic flow, is then assigned a class label based on the highest predicted probability.

The model is trained in a supervised manner using cross-entropy loss, which encourages the network to produce highly confident and accurate predictions for each traffic flow. This classification step leverages the rich and uncertainty-aware representations learned by the preceding graph layers, enabling precise and robust identification of diverse attack patterns in IIoT environments.

Algorithm 1 outlines the primary steps involved in developing the proposed FGATN. The algorithm begins with data preprocessing, where raw traffic flows are cleaned, encoded, and normalized to ensure consistency and suitability for graph-based modeling. Next, in the fuzzy set representation step, each feature is mapped to interpretable fuzzy membership values using triangular functions, chosen for their simplicity, computational efficiency, and ability to model uncertainty in IIoT environments. The fuzzy graph construction step builds a fully connected graph based on fuzzy cosine similarity, which captures nuanced linguistic overlaps between traffic flows. To address scalability, an adaptive edge pruning strategy is used to eliminate weak and noisy connections using a threshold based on the mean and standard deviation of edge weights. The core component of the framework, attentive guided fuzzy graph convolution, performs feature propagation by combining fuzzy similarity (to down-weight unreliable links) and attention mechanisms (to emphasize task-relevant neighbors), enabling more discriminative embeddings than standard GCN or GAT models alone. Finally, the node classification step employs a fully connected layer and Softmax activation to assign each flow to a specific intrusion type. Together, these steps form a unified architecture capable of handling the uncertainty, heterogeneity, and dynamic nature of IIoT traffic.

Algorithm 1: Execution Workflow of The FGATN.

Require: Raw traffic flow data $\mathcal{D}$
Ensure: Intrusion detection output (class labels)

1: **Data Preprocessing:** Remove duplicates, handle missing values, encode categorical variables, and normalize numerical features.
2: **Fuzzy Set Representation:** Map each flow to fuzzy sets using triangular membership functions to obtain fuzzy membership vectors $\boldsymbol{\mu}_i$.
3: **Fuzzy Graph Construction:** Construct a fully connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ using fuzzy cosine similarity as edge weights.
4: **Adaptive Edge Pruning:** Remove edges with weights below the threshold $\tau = \mu_w + \lambda\sigma_w$ to retain only meaningful connections.
5: **Initialization:** Set up the network architecture and hyperparameters.
6: **for** $l = 1$ to L **do**
7: **Fuzzy Similarity Computation:** Compute node similarity matrix $S^{(l)}$ from fuzzy memberships.
8: **Attention Computation:** Compute attention coefficients $A^{(l)}$ for each node using a learnable attention mechanism.
9: **Node Feature Update:** Update node features using fuzzy-attentive graph convolution using (5)
10: **Pooling:** Apply a pooling operation to reduce the dimensionality of $H^{(l+1)}$.
11: **end for**
12: **Classification:** Flatten $H^{(L)}$ and pass it through a fully connected layer with Softmax to predict class labels.

IV. DATASETS

To evaluate the proposed FGATN, we consider three datasets, including different categories of attack flows in addition to a benign class: the Edge-IIoTSet, the WSN-DS, and CIC-MalMem-2022. A concise summary of these datasets is presented below.

1) EDGE-IIOTSET DATASET:

The Edge-IIoTSet dataset is a recent cybersecurity dataset specifically collected for IoT and IIoT applications [27]. It was compiled by incorporating various devices, sensors, protocols, and configurations from cloud and edge environments. This dataset is distributed over five primary attack categories: Distributed Denial of Service (DDoS), Injection attacks, Man-in-the-Middle (MITM) attacks, Malware, Information Gathering, and Normal traffic patterns characterized by 62 features. The data collection process involved creating an authentic IIoT environment that simulated more than ten different types of IoT devices. These devices include different sensors to detect temperature, humidity, water level, soil moisture, pH, and heart rate.

2) WSN-DS

This dataset is designed to facilitate the advancement of intrusion detection systems in wireless sensor networks (WSNs) [28], which is an essential part of the IoT infrastructure. The WSN-DS data was collected using the Low Energy Aware Cluster Hierarchy (LEACH) protocol, a widely adopted hierarchical routing method in WSNs [29]. By collecting data and features using NS-2, the dataset provides insights into the characteristics and behaviors of IoT devices. WSN-DS includes 19 features and four different categories of DoS attacks: flooding, scheduling, blackhole, and grayhole, along with normal instances.

3) CIC-MALMEM-2022

This dataset originated from the Canadian Institute of Cybersecurity and was made available to the public in 2022 [30].

It is designed to address memory-based obfuscated attacks to provide information on the threats that can impact IoT devices with limited memory capacity. Using data from real-world cyberattacks, the CIC Malmem 2022 dataset reflects the types of challenge and threat encountered in practical cybersecurity scenarios. It consists of three categories of attack family, including Trojan, Spyware, and Ransomware, in addition to the benign class, and incorporates 56 distinct features.

V. EXPERIMENTS

In this part, we evaluate the performance of our proposed approach by conducting a series of experiments using three datasets. To start our evaluation, we provide a detailed description of the experimental protocols and define the evaluation metrics used. Following that, we conduct a comprehensive analysis of the attack classification results. Furthermore, we provide a comprehensive evaluation of our model, including a comparative analysis against several state-of-the-art GNN-based methods, as well as an ablation study that examines the individual contributions of the fuzzy logic and attention components within the FGATN framework.

A. EXPERIMENTAL SETUP

Google Colaboratory Pro Plus was used as the computational backbone for this study with Python 3.7.12. For GNN implementation, PyTorch Geometric (PyG), an adaptable open-source library that expands PyTorch's functionality, was employed. Furthermore, we used the t-distributed stochastic neighbor embedding (t-SNE) package to visualize the learned embedding. Best practices guided our choice of training hyper-parameters. We have opted for the cross-entropy loss function and employed the adaptive power of the Adam optimizer. The learning rates were set at 0.05 for the FGATN model, a thoughtful calibration that contributed to the robustness and precision of our experiments.

B. PERFORMANCE EVALUATION METRICS

The accuracy, precision, recall, F1-score, and MCC metrics are used to assess how well the proposed FGATN performs. They are expressed in (7)–(11). The values TP, TN, FP, and FN denote the number of True Positives, True Negatives, False Positives, and False Negatives. In addition, we have also used the confusion matrix and the Receiver Operating Characteristic (ROC) plots to provide a visual representation and enhance the interpretation of the results.

Accuracy: It assesses the general efficacy of the model across all classes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision: It evaluates how well the model classified an instance as positive or negative.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

Recall: It evaluates how well the model detects the positive instances.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

F1-score: It generates a value-added rating for performance confirmation by combining the accuracy and recall metrics.

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{10}$$

MCC: It is determined by comparing the real and predicted classifications' correlation coefficients.

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{11}$$

C. EXPERIMENTAL RESULTS

For experiments, we selected a well-chosen subset of the datasets considered. This decision is based on the impressive size of the dataset, which poses computational challenges if used in its entirety. To prepare raw data for model learning, several preprocessing operations are followed. First, the duplicated rows are removed from the dataset. To concentrate on relevant traffic characteristics, unnecessary flow features such as timestamps and IP addresses are eliminated. Categorical features are then converted into numeric values to comply with the requirements of the ML algorithms. Following that, a scaling step is applied.

To enable robust and accurate intrusion detection in IoT environments, we present FGATN, a novel framework that integrates fuzzy logic with attention mechanisms to model complex and uncertainty-aware relationships between traffic flows. The core of FGATN is a dual-weighted message passing mechanism that combines fuzzy similarity scores and learned attention coefficients. During graph convolution, fuzzy similarity scores act as a reliability gate, attenuating the influence of noisy or ambiguous edges. In parallel, attention coefficients are computed to emphasize the most informative and task-relevant neighbors. The model architecture consists of two attentive fuzzy graph convolution layers. The first layer employs $K = 8$ attention heads, experimentally chosen to capture diverse semantic relationships without overfitting or excessive computation. The second layer aggregates the refined embeddings using a single attention head, followed by a Softmax classifier to produce the final predictions. In the following, the experimental results for each dataset are reported.

1) EDGE-IIOTSET

Within the context of the Edge-IIoTSet dataset, we conducted multi-class attack classification across six attack categories. The performance metrics results of the proposed FGATN classifier are illustrated in Table 1. The table shows that the suggested FGATN performed well in all statistical parameters using the edge-IIoTSet. In addition, it achieved high accuracy with 99.07% and an MCC of 99.66%. Fig. 2 shows

TABLE 1. FGATN Performance Results Using Edge-IIoTSet

Attack Category	DDoS	Information Gathering	Injection	MITM	Malware	Normal
Pre. (%)	99	100	95	100	100	100
Rec. (%)	100	99	100	99	96	100
F1. (%)	99	99	97	99	98	100

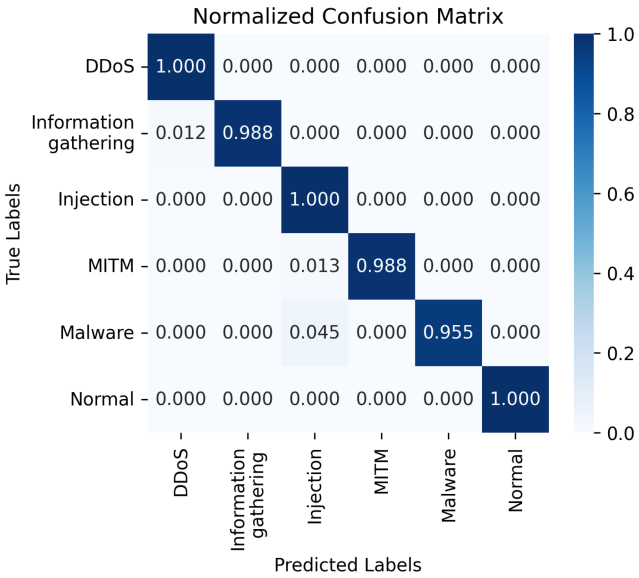


FIGURE 2. Normalized confusion matrix illustrating the classification results of the FGATN model applied to the edge-IIoTSet dataset.

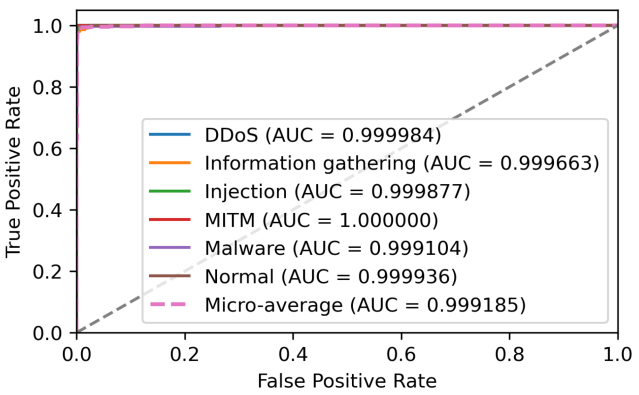


FIGURE 3. ROC curve of the proposed FGATN model evaluated on the edge-IIoTSet dataset.

the normalized confusion matrix, which demonstrates the performance of the model in various attack categories. This effectiveness is further shown by the ROC plot in Fig. 3, which achieves a micro-average AUC of 99.91%. These visualizations indicate the model's ability to handle multi-class classification challenges with high precision and reliability.

2) WSN-DS DATASET

In this experiment, the classification task was performed on five distinct attack categories, including: Flooding, Blackhole, Scheduling, Greyhole, and Normal traffic. The same experimental procedures as in the previous dataset were applied to evaluate the model's effectiveness in WSN environments. The comprehensive results of this analysis are presented in Table 2. The proposed FGATN performs well for the five classes. It achieves an accuracy of 99.2 %. Figs. 4 and 5 show the normalized confusion matrix and the ROC plot, confirming the high performance of the proposed FGATN across the five classes.

TABLE 2. FGATN Performance Results Using WSN-DS

Attack Category	Blackhole	Flooding	Grayhole	Normal	Scheduling
Pre. (%)	100	100	96	100	100
Rec. (%)	100	99	100	97	100
F1. (%)	100	99	98	98	100

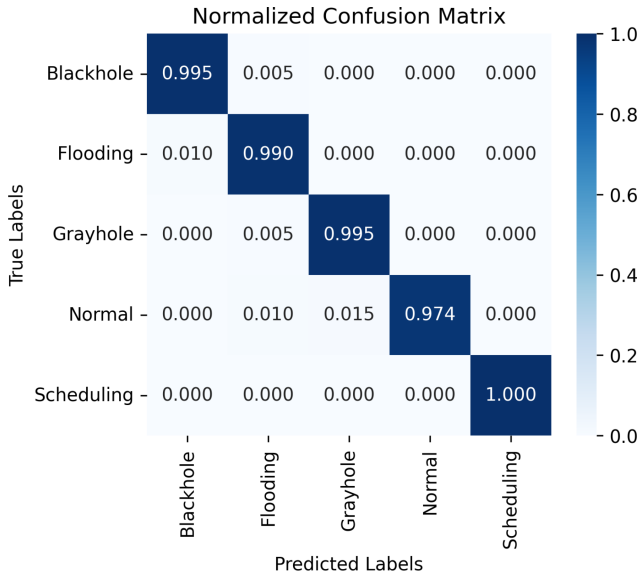


FIGURE 4. Normalized confusion matrix illustrating the classification results of the FGATN model applied to the WSN-DS dataset.

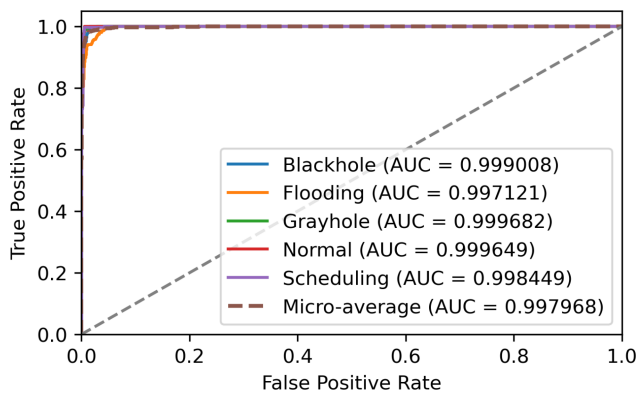


FIGURE 5. ROC curve of the proposed FGATN model evaluated on the WSN-DS dataset.

TABLE 3. FGATN Performance Results Using CIC-Malmem-2022

Attack Category	Benign	Ransomware	Spyware	Trojan
Pre. (%)	100	97	99	100
Rec. (%)	100	100	98	100
F1. (%)	100	99	98	100

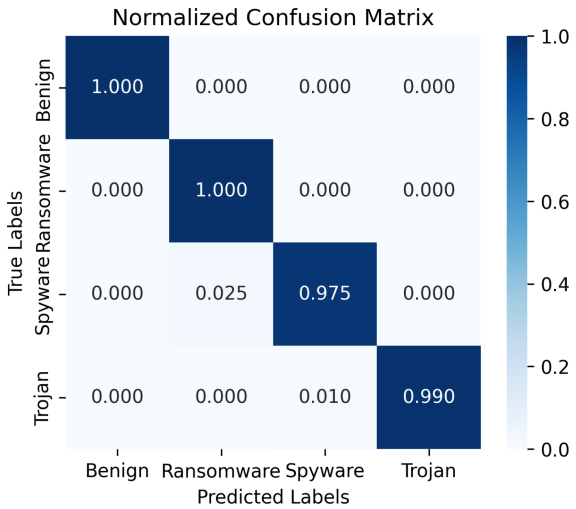


FIGURE 6. Normalized confusion matrix of the FGATN model using CIC-Malmem-2022 dataset.

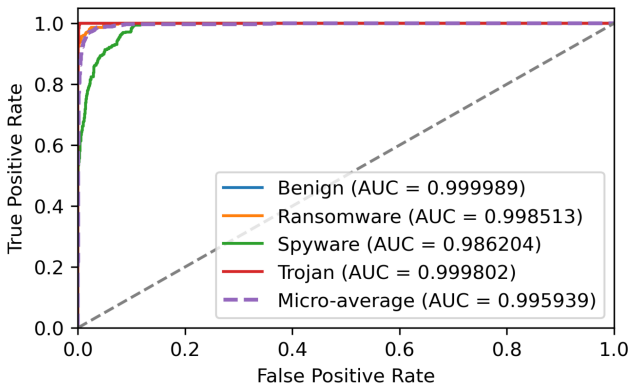


FIGURE 7. ROC curve of the proposed FGATN model evaluated on the CIC-Malmem-2022 dataset.

3) CIC-MALMEM-2022

The third experiment was carried out using the CIC-Malmem-2022 dataset, following the same procedure as the previous experiments. In this scenario, the FGATN must discriminate between three different forms of attacks, including Spyware, Ransomware, and Trojans, in addition to Benign traffic. As shown in Table 3, the proposed FGATN model demonstrated strong performance across all evaluation metrics, achieving excellent accuracy of 99.05% using this dataset. The confusion matrix and ROC plot of this experiment are presented in Figs. 6 and 7.

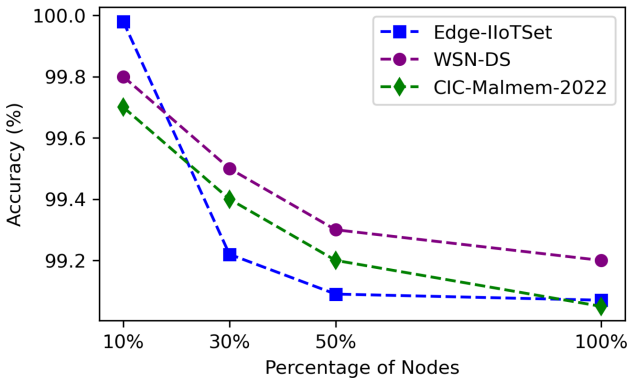


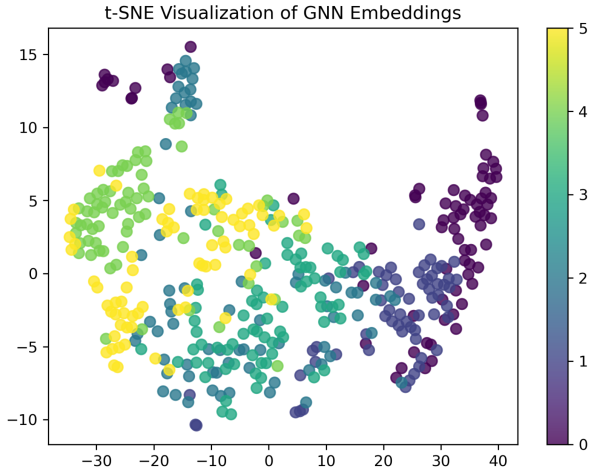
FIGURE 8. Scalability analysis of FGATN across three datasets with varying percentages of nodes.

Fig. 9 provides a comparative visualization of the learned node embeddings from the FGATN model across the three used datasets: Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022. Each row corresponds to a dataset, showing t-SNE projections of node features before and after model training. The left column illustrates the initial fuzzy-based graph representations, where data points are more entangled and less separable. In contrast, the right column shows the transformed embeddings after training with FGATN, revealing distinct and well-formed clusters for each class. This visual evidence confirms that FGATN effectively learns meaningful feature representations, improves inter-class separability, and enhances its capability to accurately detect diverse intrusion patterns in IIoT environments.

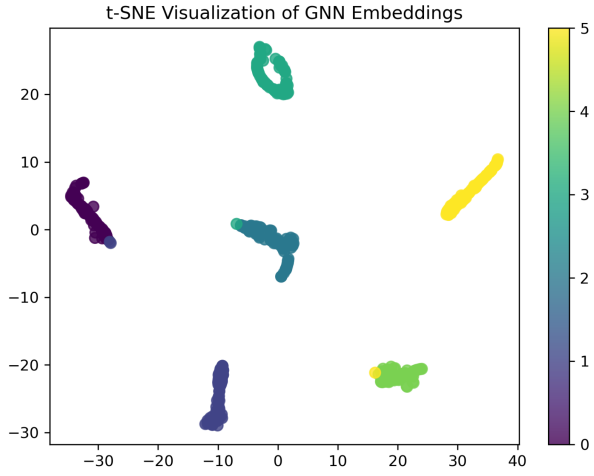
These findings demonstrate the efficiency of FGATN, which is significantly attributed to the use of fuzzy logic and attention mechanisms. Integrating fuzzy-weighted edges for graph construction plays a key role in learning representative patterns for different classes. By considering the degree of membership between connected nodes in different fuzzy groups, the model can extract meaningful patterns that differentiate one class from another in the feature space. In addition, the use of attention mechanisms improves the performance of FGATN. This method enables the model to continuously focus on the most informative graph nodes while effectively balancing the importance of each connection. The results indicate a slight decrease in accuracy as the size of the dataset increases. However, this decline is very small, with a value of less than 1%. The consistency in accuracy, despite the increase in the complexity of the dataset, highlights the scalability and resilience of the model. These findings confirm that the proposed model is highly scalable and adaptable, making it well-suited for real-world IIoT intrusion detection applications.

D. SCALABILITY ANALYSIS

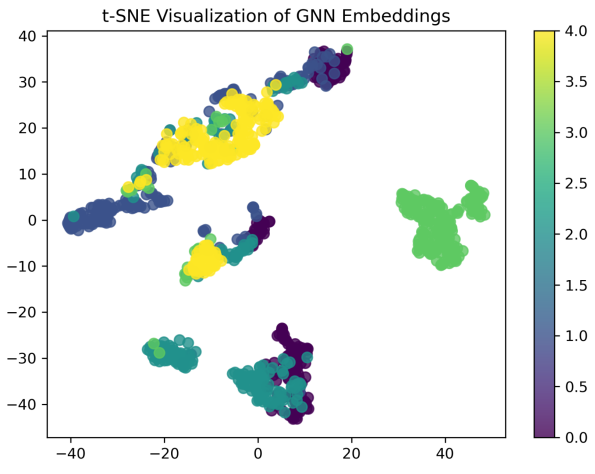
To evaluate the scalability of the proposed model, we performed a series of experiments, each carried out with different percentages of total nodes (10%, 30%, 50%, and 100%). Fig. 8



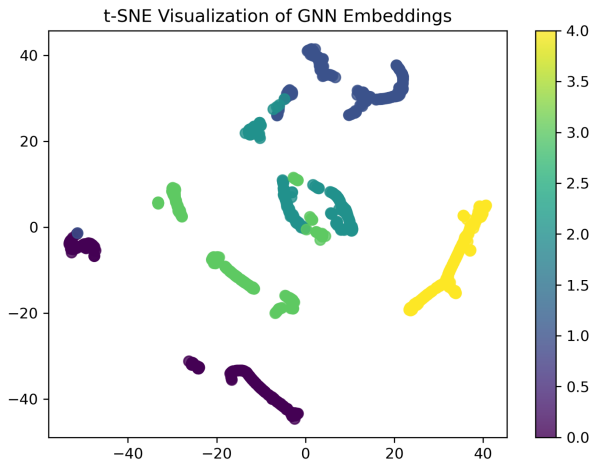
(a) Edge-IIoTSet (Before Training)



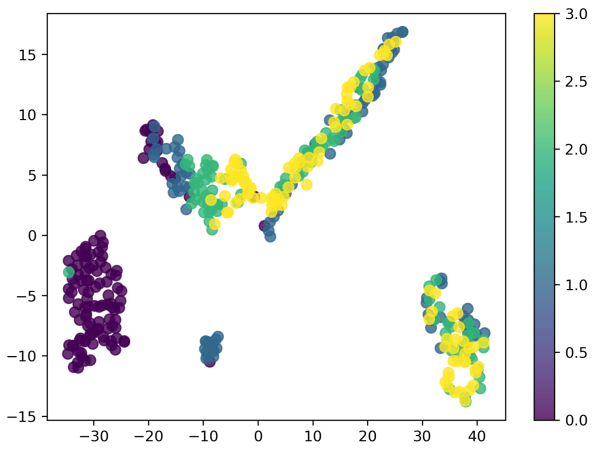
(b) Edge-IIoTSet (After Training)



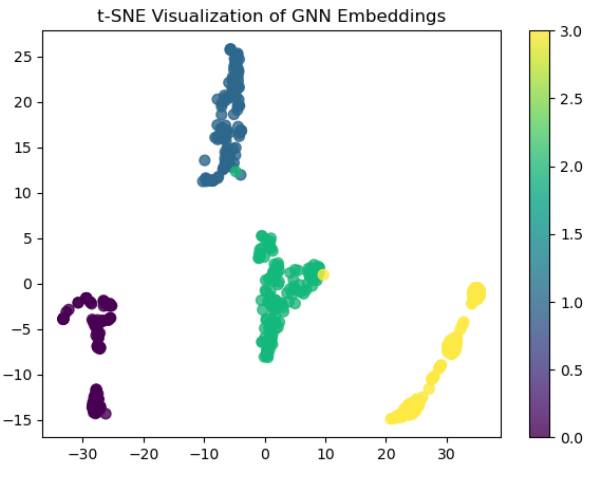
(c) WSN-DS (Before Training)



(d) WSN-DS (After Training)



(e) CIC-Malmem (Before Training)



(f) CIC-Malmem (After Training)

FIGURE 9. t-SNE visualizations comparing node embeddings from the FGATN model on the Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022 datasets. Embeddings before training are shown on the left, with post-training representations on the right.

TABLE 4. Comparison of Edge Pruning Strategies on Classification Performance

Dataset	Pruning Method	Acc. (%)	Pre. (%)	Rec. (%)	F1. (%)
Edge-IIoTSet	Adaptive (Ours)	**99.07**	**99.94**	**99.89**	**99.90**
	Top-k ($k = 5$)	97.35	97.12	97.20	97.15
	Fixed Threshold (0.5)	96.82	96.70	96.45	96.58
WSN-DS	Adaptive (Ours)	**99.20**	**99.20**	**99.23**	**99.19**
	Top-k ($k = 5$)	96.40	96.10	96.35	96.22
	Fixed Threshold (0.5)	95.85	95.60	95.90	95.75
CIC-Malmem-2022	Adaptive (Ours)	**99.05**	**98.98**	**99.10**	**99.04**
	Top-k ($k = 5$)	96.75	96.60	96.72	96.66
	Fixed Threshold (0.5)	95.90	95.20	95.85	95.52

illustrates the impact of the size of the dataset on the accuracy of the model in the three datasets used: Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022.

E. COMPARATIVE ANALYSIS OF EDGE PRUNING STRATEGIES

In addition to evaluating the model scalability, we compared our proposed *adaptive edge pruning strategy* with two commonly used alternatives: 1) fixed-threshold pruning and 2) top-k edge retention per node. Table 4 summarizes the classification performance across different datasets under these three strategies. Our adaptive method, which dynamically adjusts the pruning threshold based on the mean and standard deviation of edge weights, consistently outperformed the fixed-threshold and top-k ($k = 5$) variants. On the Edge-IIoTSet dataset, it achieved higher accuracy and F1-score while maintaining a more balanced sparsity level. On the other hand, the fixed threshold method ($\tau = 0.5$) removed too many edges, which led to a significant drop in the performance results. The results in Table 4 confirm that adaptive pruning not only improves efficiency but also preserves graph structure in a data-driven manner, enhancing classification performance.

F. ABLATION STUDY

In our ablation study, we removed the key components of the proposed FGATN to evaluate their individual contributions to performance. We considered two scenarios: 1) excluding the fuzzy logic for graph construction (GATN), and 2) omitting the attention mechanism (FGNN). The results of these ablation studies are presented in Table 5. Without the use of fuzzy logic, there is a decrease in the performance. For example, on the Edge-IIoTSet dataset, the accuracy dropped from 99.07% to 94.14%, precision fell from 99.06% to 93.63%, recall decreased from 99.17% to 94.24%, and the F1-score dropped from 99.11% to 94.06%. This highlighted the role of fuzzy weighted edges within graph nodes in effectively grouping and representing the data.

In addition, the exclusion of the attention mechanisms leads to a marginal decline in performance metrics. For the WSN-DS dataset, the accuracy decreased from 99.20% to 96.11%, precision from 99.20% to 96.55%, recall from 99.23% to 96.64%, and F1-score from 99.19% to 97.12%. This decline was particularly evident in complex attack scenarios, highlighting the usefulness of the attention mechanism in focusing on relevant features and connections for accurate classification.

G. COMPARISON WITH STATE-OF-THE-ART GNN MODELS

Table 6 summarizes the performance of the proposed FGATN model in comparison with several state-of-the-art GNN-based approaches across the three benchmark datasets. As evidenced by the results, FGATN consistently outperforms the baseline models.

1) GCN [31]

This model extracts hidden representations of nodes through a first-order approximation technique, achieving an accuracy of 75% on Edge-IIoTSet, 89% on WSN-DS, and 77.25% on CIC-Malmem-2022.

2) GRAPHSAGE [24]

It is designed to effectively create node embeddings for large graphs by sampling and aggregating information from its neighbors. It achieved 92% accuracy on Edge-IIoTSet, 98% on WSN-DS, and 97.63% on CIC-Malmem-2022. Even with good performance, it is not as excellent as FGATN in terms of accuracy and F1-score.

3) FGCN

This is a customized baseline model derived from the traditional GCN [31], adapted by us to operate on the fuzzy-based graph generated through our methodology. Unlike the standard GCN, which relies on a normalized adjacency matrix representing hard connections between nodes, the FGCN uses a fuzzy similarity graph in which edge weights reflect the degree of similarity between traffic flows based on their fuzzy set membership. It achieved an accuracy of 91% on Edge-IIoTSet, 92% on WSN-DS, and 94.50% on CIC-Malmem-2022. This model performs better than GCN, but is still below the FGATN results.

H. COMPARATIVE EVALUATION WITH EXISTING WORKS

To further validate the effectiveness of our proposed FGATN model, we conducted a comparative evaluation against methods reported in the literature, tested on the same benchmark datasets. Table 7 presents the classification accuracies achieved by FGATN compared to these existing approaches. Using the Edge-IIoTSet dataset (six-class classification), FGATN achieved an accuracy of 99.07%, outperforming the baseline Deep Neural Network (DNN), which achieved 96.01%. The DNN employs a conventional multi-layer feed-forward architecture trained on flat feature vectors, which limits its ability to capture structural or contextual dependencies among network flows.

TABLE 5. Evaluation Results of the Ablation Study Across Three Datasets

Model	Edge-IIoTSet				WSN-DS				CIC-Malmem-2022			
	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)
GATN (w/o fuzzy)	94.14	93.63	94.24	94.06	96.11	96.55	96.64	97.12	97.20	96.83	96.80	97.38
FGNN (w/o attention)	91.68	91.51	91.68	91.57	94.43	94.43	93.44	94.43	93.84	94.68	94.75	94.71
FGATN (ours)	**99.07**	**99.06**	**99.17**	**99.11**	**99.20**	**99.20**	**99.23**	**99.19**	**99.05**	**99.10**	**99.14**	**99.11**

TABLE 6. Performance Comparison Between the Proposed FGATN and State-of-The-Art GNN Models

Model	Edge-IIoTSet					WSN-DS					CIC-Malmem-2022				
	Acc. (%)	Pre. (%)	Rec. (%)	F1 (%)	MCC (%)	Acc. (%)	Pre. (%)	Rec. (%)	F1 (%)	MCC (%)	Acc. (%)	Pre. (%)	Rec. (%)	F1 (%)	MCC (%)
GCN	75.00	78.00	75.00	74.00	71.00	89.00	90.00	88.00	88.00	87.00	77.25	77.00	76.00	77.00	70.42
GraphSAGE	92.00	89.00	92.00	91.00	90.00	98.00	98.00	98.00	98.00	97.50	97.63	97.00	98.00	97.00	96.84
FGCN	91.00	92.00	91.00	91.00	90.00	92.00	92.00	92.00	92.00	90.00	94.50	95.00	94.00	94.00	93.45
FGATN (Ours)	**99.07**	**99.06**	**99.17**	**99.11**	**99.66**	**99.20**	**99.20**	**99.23**	**99.19**	**99.12**	**99.05**	**99.10**	**99.14**	**99.11**	**99.01**

TABLE 7. Comparison of Classification Accuracy Between the Proposed FGATN Model and Existing DL-Based Methods

Dataset	Classification	Method	Acc. (%)
Edge-IIoT-Set	6-class	**FGATN (ours)**	99.07
		DNN [27]	96.01
WSN-DS	5-class	**FGATN (ours)**	99.20
		GRU [32]	97
CIC-Malmem-2022	4-class	**FGATN (ours)**	99.05
		RobustCBL [33]	84.56

Using the WSN-DS dataset (five-class classification), FGATN achieved 99.20%, surpassing the Gated Recurrent Unit (GRU) model's 97.00%. Although GRU is effective for learning temporal dependencies in sequential sensor data, it lacks spatial reasoning and cannot exploit the structural layout of sensor nodes. The graph-based representation of FGATN, enhanced with fuzzy and attention mechanisms, allows it to learn both local and global spatial dependencies, resulting in a more robust classification under structural variability. Using the CIC-Malmem-2022 dataset (four-class classification), FGATN achieved 99. 05%, outperforming the RobustCBL model (CNN-BiLSTM hybrid), which achieved only 84.56%. Although RobustCBL captures localized and sequential memory patterns, it falls short in modeling higher-order interactions or long-range dependencies. FGATN, with its attention-guided fuzzy graph learning, effectively captures subtle, context-aware anomalies within complex memory traces.

VI. DISCUSSION

The proposed FGATN model demonstrates outstanding accuracy rates of 99.07%, 99.20%, and 99.05% on the Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022 datasets, respectively. These results confirm the strong generalization capacity and robustness of the model in various IoT and IIoT attack scenarios. One of the key strengths of FGATN is its ability to jointly benefit from both fuzzy semantics and attention-aware learning. By utilizing a fuzzy graph structure, the model accommodates the inherent vagueness of IIoT traffic data. The adaptive edge pruning mechanism refines this structure by removing weak and redundant connections, significantly improving sparsity and scalability without compromising accuracy. Furthermore, the integration of attention into the graph convolution process enables FGATN to adaptively focus on the most informative neighbors and enhance feature propagation and robustness to irrelevant nodes.

However, some limitations should be considered:

1) *Generalizability to Real-World IIoT Traffic:* Although benchmark datasets provide valuable insight, real-world IIoT systems contain more varied and unpredictable data, which can challenge the generalizability of the proposed model. Our future efforts will focus on the implementation of FGATN in real situations to evaluate its ability to adapt to changing environments with varying data patterns and noise. We will also explore self-supervised learning techniques to enable FGATN to learn from unlabeled data and adapt to evolving attack patterns [34].
2) *Graph Thresholding Strategy:* The current fuzzy-based graph construction employs an adaptive thresholding mechanism based on statistical heuristics. This approach effectively filters weak and redundant connections and improves graph sparsity. However, it remains a heuristic-driven process that does not explicitly optimize for downstream classification performance. To overcome this limitation, we propose exploring a reinforcement learning-based thresholding strategy in future work, where an agent learns to dynamically adjust edge pruning thresholds by observing graph statistics and maximizing task performance.

VII. CONCLUSION

This article presents a novel approach to detect and categorize IIoT network traffic flows. Our proposed approach, FGATN, leverages the power of GNNs, fuzzy logic, and attention mechanisms to achieve high performance in detecting different types of intrusions. Through a careful experimental setup and rigorous analysis, we have demonstrated the effectiveness of the proposed FGATN model. It outperformed other state-of-the-art GNNs using the Edge-IIoTSet, WSN-DS, and CIC-Malmem-2022 datasets.

In future work, we aim to include explainability in our proposed approach by using techniques that offer clear interpretations of the decision-making process. This is particularly important in IIoT cybersecurity applications, where understanding the rationale behind classifying specific traffic patterns as attacks is crucial. To enhance interpretability, we plan to integrate explainable graph learning techniques such as attention weight visualization, subgraph extraction (e.g., PGExplainer), and gradient-based attribution methods (e.g., GNNExplainer). These approaches will allow us to identify influential nodes, edges, and feature contributions that drive the predictions of the model.

ACKNOWLEDGMENT

The authors would like to thank Prince Sultan University for supporting this work and covering the article processing charges (APC).

REFERENCES

[1] M. Ozkan-Okay, R. Samet, Ö. Aslan, and D. Gupta, "A comprehensive systematic literature review on intrusion detection systems," *IEEE Access*, vol. 9, pp. 157727–157760, 2021.

[2] S. Latif et al., "Deep learning for the industrial Internet of Things (IIoT): A comprehensive survey of techniques, implementation frameworks, potential applications, and future directions," *Sensors*, vol. 21, no. 22, 2021, Art. no. 7518.

[3] S. Alshathri, E. E.-D. Hemdan, W. El-Shafai, and A. Sayed, "Digital twin-based automated fault diagnosis in industrial IoT applications," *Comput., Mater. Continua*, vol. 75, no. 1, pp. 183–196, 2023.

[4] M. A. Alkhonaini et al., "Sandpiper optimization with hybrid deep learning model for blockchain-assisted intrusion detection in IoT environment," *Alexandria Eng. J.*, vol. 112, pp. 49–62, 2025.

[5] A. Salih, S. T. Zeebaree, S. Ameen, A. Alkhyyat, and H. M. Shukur, "A survey on the role of artificial intelligence, machine learning and deep learning for cybersecurity attack detection," in *Proc. 7th Int. Eng. Conf. "Res. Innov. Amid Glob. Pandemic"*, 2021, pp. 61–66.

[6] S. B. Atitallah, M. Driss, and H. B. Ghezala, "Fedmicro-IDA: A federated learning and microservices-based framework for IoT data analytics," *Internet Things*, vol. 23, 2023, Art. no. 100845.

[7] S. Ullah, W. Boulila, A. Koubaa, and J. Ahmad, "MAGRU-IDS: A multi-head attention-based gated recurrent unit for intrusion detection in IIoT networks," *IEEE Access*, vol. 11, pp. 114590–114601, 2023.

[8] F. S. Alrayes, M. Zakariah, S. U. Amin, Z. I. Khan, and M. Helal, "Intrusion detection in IoT systems using denoising autoencoder," *IEEE Access*, vol. 12, pp. 122401–122425, 2024.

[9] M. Zhong, M. Lin, C. Zhang, and Z. Xu, "A survey on graph neural networks for intrusion detection systems: Methods, trends and challenges," *Comput. Secur.*, vol. 141, 2024, Art. no. 103821.

[10] W. Jiang, "Graph-based deep learning for communication networks: A survey," *Comput. Commun.*, vol. 185, pp. 40–54, 2022.

[11] Z. Guo and H. Wang, "A deep graph neural network-based mechanism for social recommendations," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2776–2783, Apr. 2021.

[12] W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 2, pp. 1600–1610, Feb. 2023.

[13] I. Hussain et al., "Unified fuzzy logic based approach for detection and classification of PV faults using IV trend line," *Energies*, vol. 15, no. 14, 2022, Art. no. 5106.

[14] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[15] S. Gamage and J. Samarabandu, "Deep learning methods in network intrusion detection: A survey and an objective comparison," *J. Netw. Comput. Appl.*, vol. 169, 2020, Art. no. 102767.

[16] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *J. Netw. Comput. Appl.*, vol. 153, 2020, Art. no. 102526.

[17] S. Muneer, U. Farooq, A. Athar, M. Ahsan Raza, T. M. Ghazal, and S. Sakib, "A critical review of artificial intelligence based approaches in intrusion detection: A comprehensive analysis," *J. Eng.*, vol. 2024, no. 1, 2024, Art. no. 3909173.

[18] H. J. Hadi et al., "iKern: Advanced intrusion detection and prevention at the kernel level using eBPF," *Technologies*, vol. 12, no. 8, 2024, Art. no. 122.

[19] Z. Abou El Houda, B. Brik, and S.-M. Senouci, "A novel IoT-based explainable deep learning framework for intrusion detection systems," *IEEE Internet Things Mag.*, vol. 5, no. 2, pp. 20–23, 2022.

[20] J. Zhang, C. Luo, M. Carpenter, and G. Min, "Federated learning for distributed IIoT intrusion detection using transfer approaches," *IEEE Trans. Ind. Inform.*, vol. 19, no. 7, pp. 8159–8169, Jul. 2023.

[21] I. A. Kandhro et al., "Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures," *IEEE Access*, vol. 11, pp. 9136–9148, 2023.

[22] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural networks for intrusion detection: A survey," *IEEE Access*, vol. 11, pp. 49114–49139, 2023.

[23] E. Caville, W. W. Lo, S. Layeghy, and M. Portmann, "Anomal-E: A self-supervised network intrusion detection system based on graph neural networks," *Knowl.-Based Syst.*, vol. 258, 2022, Art. no. 110030.

[24] W. W. Lo, S. Layeghy, M. Sarhan, M. Gallagher, and M. Portmann, "E-GraphSAGE: A graph neural network based intrusion detection system for IoT," in *Proc. IEEE/IFIP Netw. Operations Manage. Symp.*, 2022, pp. 1–9.

[25] L. Chang and P. Branco, "Graph-based solutions with residuals for intrusion detection: The modified e-graphsage and e-resgat algorithms," 2021, *arXiv:2111.13597*.

[26] T. Altaf, X. Wang, W. Ni, G. Yu, R. P. Liu, and R. Braun, "A new concatenated multigraph neural network for IoT intrusion detection," *Internet Things*, vol. 22, 2023, Art. no. 100818.

[27] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[28] I. Almomani et al., "WSN-DS: A dataset for intrusion detection systems in wireless sensor networks," *J. Sensors*, vol. 2016, 2016, Art. no. 4731953.

[29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. Syst. Sci.*, 2000, vol. 2, pp. 10–pp.

[30] Malware memory analysis, cic-malmem-2022 dataset, 2022. [Online]. Available: https://www.unb.ca/cic/datasets/malmem-2022.html

[31] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[32] Z. A. Khan, S. Amjad, F. Ahmed, A. M. Almasoud, M. Imran, and N. Javaid, "A blockchain-based deep-learning-driven architecture for quality routing in wireless sensor networks," *IEEE Access*, vol. 11, pp. 31036–31051, 2023.

[33] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated memory malware detection in resource-constrained IoT devices for smart city applications," *Sensors*, vol. 23, no. 11, 2023, Art. no. 5348.

[34] S. Ben Atitallah, C. Ben Rabah, M. Driss, W. Boulila, and A. Koubaa, "Self-supervised learning for graph-structured data in healthcare applications: A comprehensive review," *Comput. Biol. Med.*, vol. 188, 2025, Art. no. 109874.