Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 6, December 2026, pp. 5228~5239
ISSN: 2302-9285, DOI: 10.11591/eei.v15i6.13931

Multi-stage latent graph learning for internet of things intrusion detection

Rajakumar Ponnumani[1], Anusuya Venkatasamy[2], Selvaraj Palanisamy[3], Padmanabhan P[4], Thenmozhi Elumalai[5], Rajakumar Perumal[6]
[1]Department of Computer Science and Engineering, Parul Institute of Technology, Parul University, Vadodara, India
[2]Department of Computer Science and Engineering, Ramco Institute of Technology, Rajapalayam, India
[3]Department of Computer Science and Engineering–AI, Mother Theresa Institute of Engineering and Technology, Palamaner, India
[4]Department of Computer Science and Engineering, Greater Noida Institute of Technology (Engineering Institute), Greater Noida, India
[5]Department of Information Technology, Panimalar Engineering College, Chennai, India
[6]Department of Computer Science and Applications, Sharda School of Computing Science and Engineering, Sharda University, Greater Noida, India

Article Info

Article history:

Received Jul 10, 2026
Revised Sep 10, 2026
Accepted Oct 4, 2026

Keywords:

Graph attention network
Internet of things
Intrusion detection system
Isolation forest
Latent representation learning
Multi-class intrusion detection
Variational autoencoder

ABSTRACT

Rapid expansion of connected internet of things (IoT) infrastructures has led to increasingly heterogeneous network traffic, making timely identification of malicious activities more difficult. Conventional intrusion detection approaches frequently experience degraded performance because redundant traffic attributes obscure abnormal communication patterns and limit contextual interpretation of suspicious flows. To address these limitations, this study introduces a multi-stage latent graph learning (MS-LGL) framework that performs intrusion analysis through successive refinement of network traffic. The framework first compresses high-dimensional flow features into a compact latent space using a variational autoencoder (VAE). An isolation forest (IF) subsequently isolates suspicious traffic instances directly from the latent representations without requiring labeled attack data. Only these suspicious instances are converted into graph structures, allowing a graph attention network (GAT) to infer attack categories by exploiting structural dependencies among related communication flows. The proposed approach was validated using the CIC IoT-DIAD 2024 benchmark and assessed through standard intrusion detection metrics while being compared with representative state-of-the-art methods. The proposed framework attained 99.08% classification accuracy, a macro-average F1-score of 98.04%, and a macro-average receiver operating characteristic–area under the curve (ROC-AUC) of 0.986. These results indicate that progressive latent-space filtering combined with graph-based reasoning provides accurate and reliable multi-class intrusion detection for IoT networks.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Rajakumar Perumal
Department of Computer Science and Applications, Sharda School of Computing Science and Engineering
Sharda University
Greater Noida, Uttar Pradesh, India
Email: rajkumar.jcet@gmail.com

1. INTRODUCTION

Modern internet of things (IoT) ecosystems interconnect billions of heterogeneous devices that continuously exchange data across smart homes, industrial automation, healthcare services, and critical infrastructure. This unprecedented level of connectivity has substantially increased network complexity and

broadened the exposure to cyber threats [1], [2]. As these devices increasingly exchange large volumes of heterogeneous traffic through cloud platforms, network communication has become more complex. This expansion has also increased the cyberattack surface. IoT networks are highly heterogeneous and operate with limited computational resources. The coexistence of multiple communication protocols further increases network complexity, allowing cyber threats to propagate rapidly. Attack techniques continue to evolve and include distributed denial-of-service (DoS) attacks, spoofing, reconnaissance, botnet activities, and malware. Consequently, maintaining secure network operation has become increasingly difficult. Consequently, intrusion detection systems (IDSs) play a central role in continuously monitoring network behavior and identifying malicious communication before it compromises IoT services [3], [4]. However, achieving high detection accuracy remains challenging. IoT traffic is characterized by high dimensionality, redundant feature distributions, and substantial variations between benign and malicious communication patterns [5]–[7].

Data-driven learning models have enabled IDSs to identify complex traffic patterns that are difficult to capture using rule-based or signature-driven approaches. By learning behavioral characteristics directly from network traffic, these models have substantially improved detection capability in dynamic IoT environments. Sun *et al.* [8] employed graph neural networks to model attack relationships and demonstrated improved detection capability through graph-based feature propagation. Tran and Park [9] combined graph convolutional learning with feature selection to enhance intrusion classification across benchmark datasets. Nguyen and Park [10] incorporated continual learning into graph neural networks to improve adaptation to evolving attack scenarios. Although graph-based models capture structural dependencies effectively, their performance depends heavily on graph quality and often assumes that meaningful graph representations already exist.

Another research direction has focused on hybrid learning architectures that combine deep feature extraction with conventional classifiers. Hybrid intrusion detection architectures have also attracted considerable attention. Haque and Soliman [11] combined transformer-based feature encoding with an anomaly detection pipeline to strengthen the identification of abnormal network behaviour. Similarly, Sajid *et al.* [12] developed a hybrid framework that integrates convolutional learning, temporal sequence modelling, and gradient boosting. The framework improves intrusion detection across heterogeneous benchmark datasets. Similarly, Villegas-Ch *et al.* [13] explored dynamic graph construction for IoT security, whereas Ghadermazi *et al.* [14] introduced graph transformer autoencoders to improve feature representation and classification accuracy. These approaches demonstrate the benefits of deep representation learning but generally require extensive training and often exhibit limited robustness when network traffic characteristics evolve.

Recent studies have also investigated distributed learning, feature engineering, and optimization techniques to improve IDS scalability and efficiency. Gao *et al.* [15] proposed a distributed graph neural network architecture for edge-based intrusion detection, reducing computational overhead through decentralized processing. Li *et al.* [16] investigated different approaches for representing IoT network traffic and showed that the choice of data representation has a direct impact on the effectiveness of intrusion detection models. Likewise, Karthikeyan *et al.* [17] employed bio-inspired optimization for feature selection and classification, achieving competitive detection accuracy on conventional intrusion datasets. Despite these advances, most existing studies perform feature learning, anomaly detection, and attack classification as separate stages. This separation limits their ability to exploit progressively refined traffic representations throughout the intrusion detection process [18], [19].

An additional limitation of current IDS frameworks is their reliance on direct classification using either original traffic features or a single learned representation. Such approaches often fail to distinguish subtle anomalies embedded within high-dimensional traffic and rarely consider contextual relationships among suspicious network flows during final classification. Consequently, unknown attacks, coordinated malicious activities, and complex communication patterns remain difficult to identify reliably in dynamic IoT-cloud environments. These observations indicate the need for an intrusion detection framework that progressively refines network traffic representations before performing contextual attack classification.

Motivated by these limitations, the proposed multi-stage latent graph learning (MS-LGL) framework introduces a progressive learning strategy for IoT intrusion detection. Instead of directly classifying high-dimensional network traffic, the framework first transforms network flows into compact latent embeddings using a variational autoencoder (VAE). This process preserves informative traffic characteristics while suppressing redundant feature variations. The latent embeddings are subsequently analyzed using an isolation forest (IF) to identify anomalous traffic without requiring attack labels. Finally, anomalous traffic instances are represented as graph nodes and classified through a graph attention network (GAT). This enables contextual reasoning based on relationships among suspicious communication flows. This sequential learning strategy progressively enhances traffic discrimination while improving detection robustness across diverse attack categories.

The principal contributions of this study are as follows:

a. A progressive IoT intrusion detection architecture integrating latent feature extraction, anomaly-based traffic filtering, and graph attention-based attack classification.
b. A VAE-based latent-space anomaly screening mechanism for separating suspicious traffic from normal traffic before multi-class classification.
c. A graph attention model that captures structural dependencies among anomalous communication flows for contextual attack classification.
d. Comprehensive experimental validation on the CIC IoT-DIAD 2024 benchmark, including comparisons with representative learning-based intrusion detection methods.

2. METHOD

The proposed MS-LGL framework consists of three consecutive learning stages that progressively refine IoT network traffic before final attack classification, as shown in Figure 1. Network flows are first standardized to produce normalized feature vectors suitable for subsequent learning. These features are then encoded by a VAE into a lower-dimensional latent representation that retains informative traffic behavior while suppressing irrelevant feature variations. The resulting latent embeddings form the basis for the subsequent anomaly detection and graph-based classification stages. The resulting latent representations are subsequently analyzed using an IF to identify anomalous traffic without requiring attack labels. Finally, only the anomalous instances are transformed into a graph representation and classified using a GAT, where both node attributes and relationships among suspicious communication flows contribute to the final prediction. This progressive learning strategy enables the framework to refine intrusion patterns before contextual classification, thereby improving detection robustness in dynamic IoT environments.

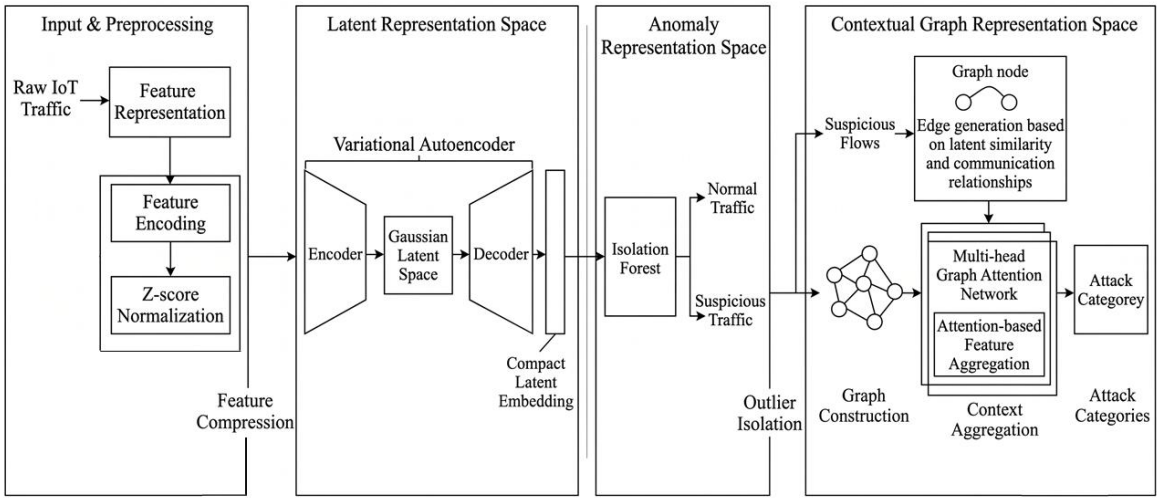


Figure 1. Architecture of the proposed MS-LGL framework for IoT intrusion detection

2.1. Multi-stage latent representation learning

The first learning stage focuses on compressing network traffic into a lower-dimensional representation while retaining the information necessary to distinguish normal and malicious communication. Let the normalized traffic dataset be represented as:

$$X = \{x_1, x_2, \dots, x_n\}, x_i \in R^d$$

where d denotes the number of traffic features.

Each feature is standardized using Z-score normalization to eliminate scale differences among heterogeneous network attributes,

$$z_i = \frac{x_i - \mu}{\sigma} \quad (1)$$

where μ represents the feature mean and σ its corresponding standard deviation.

The normalized feature vectors are then encoded by the VAE [20], [21] into a lower-dimensional latent representation. Unlike conventional autoencoders, the VAE learns a probability distribution rather than a deterministic feature mapping, enabling compact representations that remain robust to traffic variations. To enable gradient-based optimization, latent samples are generated using the reparameterization strategy,

$$z = \mu + \sigma \odot \varepsilon, \varepsilon \sim \mathcal{N}(0, I) \quad (2)$$

where μ and σ are the learned latent parameters and ε is sampled from a standard Gaussian distribution. The VAE is optimized by jointly minimizing reconstruction error and latent-space regularization [22],

$$L = L_{\text{rec}} + \beta D_{KL}\big(q(z|x) \,|\, p(z)\big) \quad (3)$$

where L_{rec} denotes the reconstruction loss and D_{KL} constrains the latent distribution to follow a Gaussian prior.

The learned latent vectors provide a compact representation that suppresses redundant traffic variations while retaining discriminative behavioral information for subsequent anomaly detection.

2.2. Latent-space anomaly detection

The compressed latent representations are analyzed using an IF [23], [24] to separate suspicious traffic from normal communication. Instead of learning explicit decision boundaries, the IF isolates anomalous observations through recursive random partitioning [25], making it particularly suitable for imbalanced intrusion datasets.

For each latent representation, the anomaly score is computed as (4):

$$s(x) = 2^{-\frac{E(h(x))}{c(n)}} \quad (4)$$

where *E(h(x))* denotes the average isolation path length associated with sample *x*, and *c(n)* is the normalization factor corresponding to the expected path length for a dataset of size *n*.

Traffic samples assigned high anomaly scores are considered suspicious and forwarded to the next stage, whereas benign samples are discarded from further processing. This intermediate filtering stage significantly reduces the search space for graph learning while improving the discrimination of previously unseen attack behaviors.

2.3. Graph-based contextual classification

Although anomaly detection identifies suspicious traffic, many cyberattacks exhibit coordinated communication patterns that cannot be recognized from individual flows alone. Therefore, the anomalous traffic is represented as a graph:

$$G = (V, E)$$

where each node corresponds to an anomalous network flow and edges represent relationships established according to communication similarity, temporal proximity, and latent feature correlation.

A GAT [26], performs contextual classification by aggregating information from neighboring nodes through adaptive attention weights. The node representation is updated as (5):

$$h_i' = \sigma\big(\textstyle\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W h_j\big) \quad (5)$$

where W denotes the learnable transformation matrix, α_{ij} represents the attention coefficient between neighboring nodes, and σ(·) is the nonlinear activation function.

Unlike conventional graph convolution methods that assign equal importance to neighboring nodes, the attention mechanism automatically emphasizes more informative communication relationships. Consequently, the classifier captures both local traffic characteristics and global structural dependencies, improving the identification of coordinated and stealthy attacks.

2.4. Unified detection pipeline

The proposed framework integrates representation learning, anomaly detection, and graph reasoning into a single sequential intrusion detection pipeline. Initially, raw IoT traffic is transformed into compact latent representations through the VAE. These latent embeddings are subsequently examined by the IF, which filters suspicious communication patterns without requiring attack labels. Rather than directly

classifying all traffic instances, only anomalous samples are converted into graph nodes, allowing the GAT to focus on contextual relationships among suspicious communications. This progressive refinement strategy reduces redundant computations while enabling increasingly discriminative feature representations across successive learning stages. Consequently, the framework combines statistical feature learning, unsupervised anomaly filtering, and graph-based contextual reasoning within a unified architecture suitable for dynamic IoT-cloud environments.

3. RESULTS AND DISCUSSION

Performance evaluation was carried out using the CIC IoT-DIAD 2024 benchmark, which contains benign network traffic together with multiple categories of IoT attacks. The MS-LGL framework was developed using the PyTorch and Scikit-learn libraries and executed in a Google Colab environment equipped with an NVIDIA Tesla T4 GPU. Detection capability was assessed through accuracy, precision, recall, F1-score, and receiver operating characteristic–area under the curve (ROC-AUC). The experimental settings adopted throughout the study are summarized in Table 1.

Table 1. Experimental configuration of the MS-LGL framework

Experimental setting	Specification	Experimental setting	Specification
Data split	70%/10%/20%	GAT dropout	0.3
VAE latent dimension	21	GAT learning rate	0.0005
VAE selection	Ablation: 10, 15, 21, 30, 40	GAT training	200 epochs
IF	200 estimators; contamination=0.1; τ=0.55	Optimizer	Adam
GAT architecture	2 layers; 64 hidden units	Hardware	NVIDIA Tesla T4, 16 GB VRAM
Attention heads	8 (first layer), 1 (output layer)	CPU/RAM	Intel Xeon, 2.2–2.6 GHz/13–25 GB RAM
GAT activation	LeakyReLU, α=0.2	Hardware	NVIDIA Tesla T4, 16 GB VRAM

3.1. Experimental configuration

All experiments followed the configuration presented in Table 1. During inference, network traffic was first projected into a latent representation by the VAE, after which the IF identified suspicious traffic instances. The retained anomalous samples were subsequently processed by the GAT to determine their corresponding attack categories. For multi-class evaluation, ROC curves, and AUC values were computed using the one-vs-rest strategy with macro-averaging.

Class imbalance was considered during model development and evaluation. Stratified sampling was used to preserve class proportions across the data partitions, and class weighting was applied during GAT training to increase the contribution of under-represented attack classes. The IF contamination parameter was calibrated during parameter selection, while class-wise and macro-averaged metrics were reported to reduce the influence of majority-class dominance during evaluation.

3.2. Comparative performance evaluation

To examine the relative performance of the proposed approach, MS-LGL was benchmarked against recent graph-based and hybrid intrusion detection models reported in the literature. The comparative results are presented in Table 2. On the CIC IoT-DIAD 2024 benchmark, the proposed framework attained an accuracy of 99.08% together with a macro-average F1-score of 98.04%, exceeding the F1-score reported by the selected reference methods while maintaining consistently high classification accuracy.

Table 2. Performance comparison with representative intrusion detection methods

Reference	Method	Dataset	Accuracy (%)	F1-score (%)
Sun *et al*. [8]	GCN/GAT	CICIDS2017+Synthetic	98.87	96.62
Tran and Park [9]	FN-GNN	CICIDS2017, UNSW-NB15	98.65	95.65
Nguyen and Park [10]	EL-GNN	CICIDS2017, UNSW-NB15	98.44	95.72
Sajid *et al*. [12]	CNN-LSTM-XGBoost	CICIDS2017, UNSW, NSL-KDD	97.35	95.10
Gao *et al*. [15]	Distributed GNN	NF-CSE-CIC-IDS2018-v2	97.83	96.79
Proposed MS-LGL	**VAE-IF-GAT**	**CIC IoT-DIAD 2024**	**99.08**	**98.04**

The comparison also indicates that progressively refining network traffic before final classification benefits intrusion recognition. Instead of directly processing the original feature space, MS-LGL first derives

latent traffic representations, identifies suspicious instances through anomaly screening, and subsequently exploits structural relationships among anomalous flows using graph attention. This sequential processing strategy contributes to improved discrimination across multiple IoT attack categories while maintaining balanced overall detection performance.

3.3. Effect of multi-stage learning

To understand the contribution of each component, the MS-LGL framework was examined by progressively activating its three learning stages. The evaluation began with the VAE, followed by the addition of the IF, and finally the complete framework incorporating the GAT. The corresponding performance at each stage is reported in Table 3.

Table 3. Incremental performance of the proposed MS-LGL framework

Configuration	Accuracy (%)	F1-score	Learning task
VAE	97.81	0.9512	Reconstruction-based anomaly detection
VAE+IF	98.85	0.9940	Binary anomaly detection
MS-LGL (VAE+IF+GAT)	99.08	0.9804	Multi-class intrusion detection

The results reveal that each learning stage fulfils a distinct role within the overall framework. Using only the VAE, the model learned compact traffic representations suitable for anomaly characterization. Incorporating the IF substantially improved binary anomaly discrimination, increasing the detection accuracy to 98.85% with an F1-score of 0.9940. Integrating the GAT completed the learning pipeline and produced an overall multi-class detection accuracy of 99.08%, indicating that contextual graph reasoning complements latent representation learning and anomaly filtering.

The training characteristics of the VAE are illustrated in Figure 2. Both the training and validation losses decrease smoothly before reaching a stable plateau, suggesting consistent convergence during optimization and effective learning of the latent feature space.

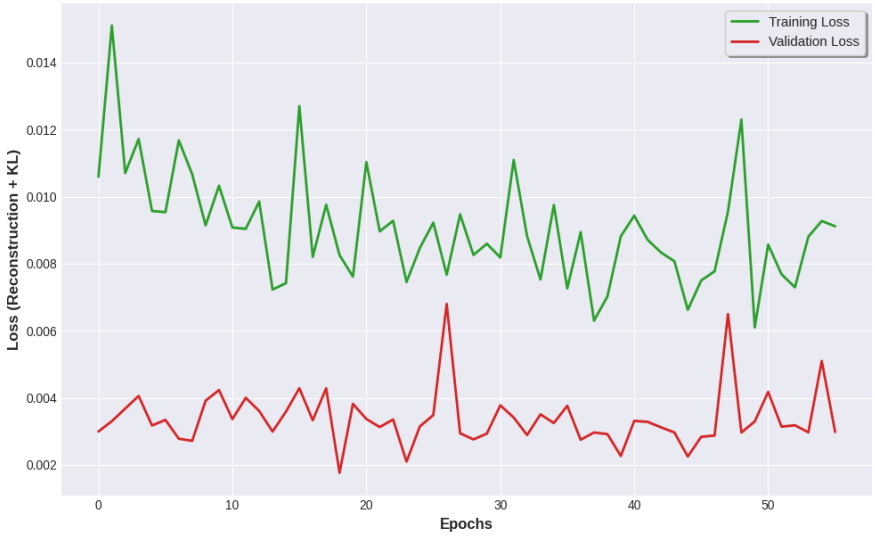


Figure 2. Training and validation loss of the VAE

The anomaly detection results produced by the IF are presented in Figure 3. Figure 3(a) shows the overall anomaly detection confusion matrix, while Figures 3(b)-(d) present the corresponding confusion matrices for Recon, Mirai, and DoS attacks, respectively. The overall matrix in Figure 3(a) confirms that most anomalous traffic instances were correctly separated from benign communication. Similarly, Figures 3(b)–(d) show high correct-classification counts for the Recon, Mirai, and DoS categories, with comparatively limited false-positive and false-negative predictions. These observations indicate that the latent embeddings generated by the VAE provide a reliable feature space for identifying suspicious traffic before the graph-based classification stage. Taken together, the incremental evaluation, convergence analysis,

and confusion matrices show that the three-stage learning strategy consistently strengthens the overall intrusion detection process, with each component contributing a complementary function to the final MS-LGL framework.

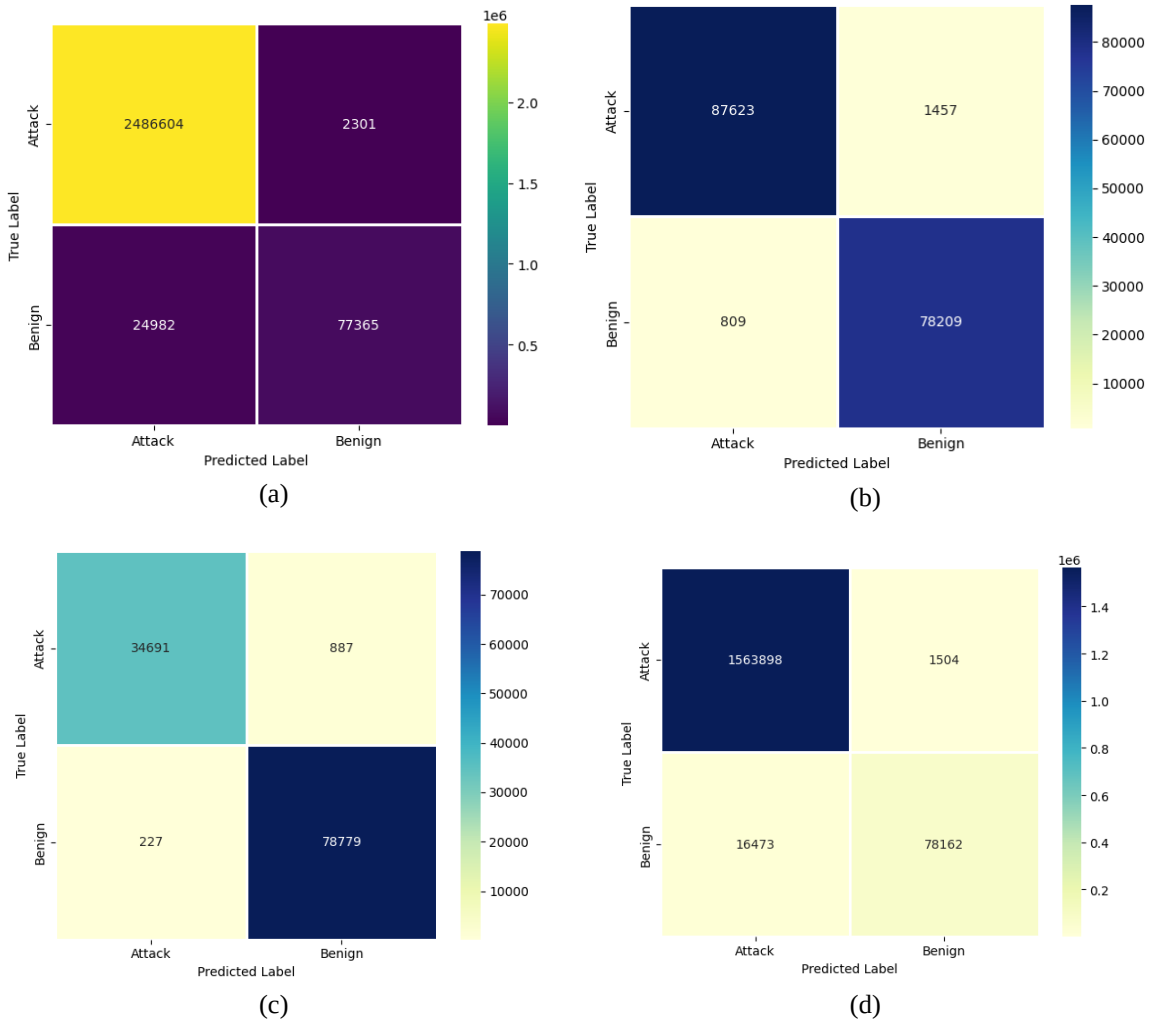


Figure 3. Confusion matrices of the IF based on VAE latent representations; (a) overall anomaly detection, (b) Recon attack detection, (c) Mirai attack detection, and (d) DoS attack detection

3.4. Class-wise detection performance evaluation

Performance across individual attack categories is reported in Table 4, providing a detailed assessment of the classification capability of the proposed MS-LGL framework. Strong and consistent results were obtained for all evaluated attack classes, with F1-scores between 0.9610 and 0.9940. Among the evaluated attacks, DoS achieved the highest F1-score (0.9940), whereas web-based attacks produced the lowest value (0.9610). The lower F1-score for web-based attacks is primarily associated with their lower precision (0.9351), despite a high recall of 0.9884, indicating that some suspicious flows from other attack categories were assigned to the web-based class. This behavior may arise from greater variability and feature overlap in web-based traffic, where heterogeneous communication patterns can exhibit characteristics similar to other anomalous flows. Nevertheless, the high recall indicates that the framework successfully identifies most web-based attack instances once they are forwarded to the graph-based classification stage. Nevertheless, the framework maintained a macro-average F1-score of 0.9804 together with an overall accuracy of 0.9908, indicating balanced recognition performance across both dominant and relatively infrequent intrusion categories.

Table 4. Per-class intrusion detection performance of the proposed MS-LGL framework on the CIC IoT-DIAD 2024 dataset

Attack class	Precision	Recall	Specificity	F1-score	Accuracy
DoS	0.9989	0.9891	0.9814	0.9940	0.9891
DDoS	0.9983	0.9872	0.9829	0.9927	0.9866
Mirai	0.9753	0.9931	0.9888	0.9841	0.9908
Recon	0.9841	0.9911	0.9817	0.9876	0.9867
BruteForce	0.9628	0.9721	0.9995	0.9674	0.9993
Spoofing	0.9590	0.9929	0.9832	0.9757	0.9861
Web-based	0.9351	0.9884	0.9979	0.9610	0.9973
Macro average	**0.9734**	**0.9877**	**0.9879**	**0.9804**	**0.9908**

To complement the class-wise evaluation, Figure 4 presents the macro-average ROC curve obtained from the final GAT classifier. The ROC profile remains close to the upper-left boundary and yields a macro-average AUC of 0.986, demonstrating strong separability across the evaluated attack classes. Together with the results in Table 4, the ROC analysis indicates that the proposed framework maintains stable discrimination capability over multiple intrusion categories despite differences in class distribution.

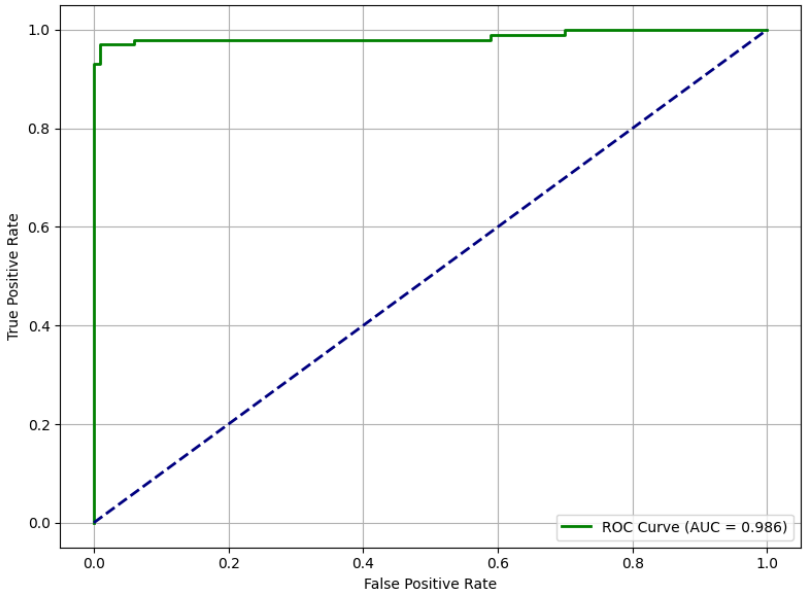


Figure 4. Macro-average ROC curve of the proposed MS-LGL framework for multi-class intrusion detection

3.5. Ablation analysis

An ablation study was conducted to examine the influence of the latent representation dimension on intrusion detection performance. Different latent dimensions were evaluated while maintaining identical training conditions. As presented in Table 5, very small latent spaces resulted in information loss, whereas larger latent spaces introduced redundant representations that marginally reduced generalization performance. A moderate latent dimension achieved the best balance between feature compression and classification accuracy, demonstrating that an appropriate latent representation is essential for effective anomaly discrimination and subsequent graph learning.

Table 5. Effect of latent dimension on detection performance

Latent dimension	Precision	Recall	Accuracy (%)	F1-score	Reconstruction loss (↓)
10	0.9817	0.9714	97.92	0.9765	0.0127
15	0.9879	0.9819	98.39	0.9849	0.0114
21	**0.9989**	**0.9891**	**98.92**	0.9940	**0.0105**
30	0.9916	0.9785	98.21	0.9850	0.0099
40	0.9848	0.9720	98.31	0.9784	0.0097

3.6. Discussion

The experimental findings indicate that progressively refining network traffic before final classification improves intrusion detection performance in IoT environments. The comparative study demonstrated that MS-LGL achieved favorable performance relative to recent learning-based intrusion detection models, while the stage-wise evaluation showed that each component contributed a complementary function within the overall framework. Latent representation learning reduced feature redundancy, the IF effectively identified suspicious traffic for further analysis, and graph attention learning strengthened the recognition of multiple attack categories by exploiting relationships among anomalous communication flows. Consistent convergence behavior, confusion matrix analysis, and the macro-average ROC results collectively support the reliability of the proposed learning strategy.

From an engineering perspective, the measured inference characteristics indicate the feasibility of the proposed framework for near-real-time IoT traffic analysis. The VAE and IF stages require only 0.45 ms/sample and 0.12 ms/sample, respectively, while the complete pipeline achieves an end-to-end throughput of approximately 11,500 flow records/s. The graph-based stage requires higher computational resources, with peak GPU memory usage of approximately 7.8 GB, reflecting the additional cost of graph construction and contextual attention. Since the GAT is applied only to traffic retained by the anomaly-filtering stage, this computational cost is restricted to suspicious flows rather than the complete traffic stream. For constrained edge devices, the current GPU-based configuration may not be directly deployable; however, the lightweight VAE and IF stages can serve as an initial filtering layer, while GAT inference can be offloaded to a nearby gateway or edge-cloud node. Further model compression and lightweight graph architectures can be investigated for fully on-device deployment. Although the framework was validated using the CIC IoT-DIAD 2024 benchmark, this single-dataset evaluation limits direct assessment of its generalizability across different network environments and attack distributions.

Future work will evaluate MS-LGL on heterogeneous benchmarks, such as UNSW-NB15 and CICIDS2017, to assess cross-dataset robustness and transferability under varying traffic characteristics and attack distributions. For rare attack types, the anomaly-filtering stage can prioritize suspicious flows before graph classification, while class-weighted GAT training can increase the influence of under-represented attack categories. For evolving or previously unseen attacks, the unsupervised latent-space representation and anomaly detection stages provide a basis for identifying deviations from established traffic patterns, while incremental updating of the graph classifier could be used to incorporate newly observed attack patterns without redesigning the entire pipeline. Future investigations will also evaluate the adversarial robustness of MS-LGL against perturbed or strategically manipulated network traffic to assess its resilience under attack-aware conditions.

4. CONCLUSION

This paper presented the MS-LGL framework for multi-class intrusion detection in IoT networks. The framework integrates VAE-based latent representation learning, unsupervised IF anomaly filtering, and graph-based contextual classification through a GAT, progressively refining network traffic before final attack identification. Experimental evaluation on the CIC IoT-DIAD 2024 dataset demonstrated that MS-LGL achieved 99.08% accuracy, a macro-average F1-score of 98.04%, and a macro-average ROC-AUC of 0.986. The comparative, incremental, and class-wise evaluations confirmed the complementary contributions of the three learning stages and demonstrated effective detection across diverse IoT attack categories. The proposed framework therefore provides a progressive and context-aware approach that reduces redundant feature variations and focuses graph-based classification on suspicious traffic. Future work will extend the framework through cross-dataset validation, adaptive online learning, and computational optimization to improve robustness and support real-time deployment in resource-constrained IoT environments.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rajakumar Ponnumani	✓			✓		✓		✓		✓	✓			
Anusuya Venkatasamy		✓	✓		✓		✓		✓			✓		
Selvaraj Palanisamy	✓			✓		✓		✓		✓	✓			
Padmanabhan P	✓		✓		✓			✓		✓	✓			
Thenmozhi Elumalai		✓		✓	✓	✓		✓		✓		✓	✓	
Rajakumar Perumal	✓		✓		✓		✓		✓		✓		✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [initials: R.P], upon reasonable request.

REFERENCES

[1] S. A. Abdulkareem, C. H. Foh, M. Shojafar, F. Carrez, and K. Moessner, "Network Intrusion Detection: An IoT and Non IoT-Related Survey," *IEEE Access*, vol. 12, pp. 147167–147191, 2024, doi: 10.1109/ACCESS.2024.3473289.

[2] Y. Li, Z. Li, and M. Li, "A comprehensive survey on intrusion detection algorithms," *Computers and Electrical Engineering*, vol. 121, Jan. 2025, doi: 10.1016/j.compeleceng.2024.109863.

[3] M. Algarni *et al.*, "An intrusion detection system based on lightweight BERT with meta-classifier for internet of things environments," *PeerJ Computer Science*, vol. 12, p. e3590, Feb. 2026, doi: 10.7717/peerj-cs.3590.

[4] M. Wang, N. Yang, Y. Guo, and N. Weng, "Learn-IDS: Bridging Gaps between Datasets and Learning-Based Network Intrusion Detection," *Electronics (Switzerland)*, vol. 13, no. 6, Mar. 2024, doi: 10.3390/electronics13061072.

[5] T. Zhukabayeva, Z. Ahmad, A. Adamova, N. Karabayev, and A. Abdildayeva, "An Edge-Computing-Based Integrated Framework for Network Traffic Analysis and Intrusion Detection to Enhance Cyber–Physical System Security in Industrial IoT," *Sensors*, vol. 25, no. 8, p. 2395, Apr. 2025, doi: 10.3390/s25082395.

[6] M. Thirumalaisamy, S. Yogarayan, M. S. Sayeed, S. F. A. Razak, and R. Shunmugam, "Fog-Aware Hierarchical Autoencoder with Density-Based Clustering for AI-Driven Threat Detection in Smart Farming IoT Systems," *Future Internet*, vol. 17, no. 12, Dec. 2025, doi: 10.3390/fi17120567.

[7] S. K. S *et al.*, "An Unsupervised Cloud-Centric Intrusion Diagnosis Framework Using Autoencoder and Density-Based Learning," *Future Internet*, vol. 18, no. 1, p. 54, Jan. 2026, doi: 10.3390/fi18010054.

[8] Z. Sun, A. M. H. Teixeira, and S. Toor, "GNN-IDS: Graph Neural Network based Intrusion Detection System," in *ACM International Conference Proceeding Series*, Association for Computing Machinery, Jul. 2024, pp. 1-12, doi: 10.1145/3664476.3664515.

[9] D. H. Tran and M. Park, "FN-GNN: A Novel Graph Embedding Approach for Enhancing Graph Neural Networks in Network Intrusion Detection Systems," *Applied Sciences (Switzerland)*, vol. 14, no. 16, Aug. 2024, doi: 10.3390/app14166932.

[10] T.-T. Nguyen and M. Park, "EL-GNN: A Continual-Learning-Based Graph Neural Network for Task-Incremental Intrusion Detection Systems," *Electronics (Basel)*, vol. 14, no. 14, p. 2756, Jul. 2025, doi: 10.3390/electronics14142756.

[11] A. Haque and H. Soliman, "A Transformer-Based Autoencoder with Isolation Forest and XGBoost for Malfunction and Intrusion Detection in Wireless Sensor Networks for Forest Fire Prediction," *Future Internet*, vol. 17, no. 4, Apr. 2025, doi: 10.3390/fi17040164.

[12] M. Sajid *et al.*, "Enhancing intrusion detection: a hybrid machine and deep learning approach," *Journal of Cloud Computing*, vol. 13, no. 1, Dec. 2024, doi: 10.1186/s13677-024-00685-x.

[13] W. Villegas-Ch, J. Govea, A. M. Navarro, and P. P. Játiva, "Intrusion Detection in IoT Networks Using Dynamic Graph Modeling and Graph-Based Neural Networks," *IEEE Access*, vol. 13, pp. 65356–65375, 2025, doi: 10.1109/ACCESS.2025.3559325.

[14] J. Ghadermazi, S. Hore, A. Shah, and N. D. Bastian, "GTAE-IDS: Graph Transformer-Based Autoencoder Framework for Real-Time Network Intrusion Detection," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 4026–4041, 2025, doi: 10.1109/TIFS.2025.3557741.

[15] Q. Gao, S. Kausar, and H. X. Zhang, "Incremental-learning-based graph neural networks on edge-forwarding devices for network intrusion detection," *Alexandria Engineering Journal*, vol. 126, pp. 81–89, Jul. 2025, doi: 10.1016/j.aej.2025.03.102.

[16] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00892-y.

[17] M. Karthikeyan, D. Manimegalai, and K. RajaGopal, "Firefly algorithm based WSN-IoT security enhancement with machine learning for intrusion detection," *Scientific Reports*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-50554-x.

[18] M. Paricherla, M. Ritonga, S. R. Shinde, S. M. Chaudhari, R. Linur, and A. Raghuvanshi, "Machine learning techniques for accurate classification and detection of intrusions in computer network," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 4, pp. 2340–2347, Aug. 2023, doi: 10.11591/eei.v12i4.4708.

[19] A. Aldaej, T. A. Ahanger, and I. Ullah, "Deep Learning-Inspired IoT-IDS Mechanism for Edge Computing Environments," *Sensors*, vol. 23, no. 24, Dec. 2023, doi: 10.3390/s23249869.
[20] H. Huang, J. Yang, H. Zeng, Y. Wang, and L. Xiao, "Self-Organizing Maps-Assisted Variational Autoencoder for Unsupervised Network Anomaly Detection," *Symmetry (Basel)*, vol. 17, no. 4, p. 520, Mar. 2025, doi: 10.3390/sym17040520.
[21] Y. Ren, K. Feng, F. Hu, L. Chen, and Y. Chen, "A Lightweight Unsupervised Intrusion Detection Model Based on Variational Auto-Encoder," *Sensors (Basel)*, vol. 23, no. 20, Oct. 2023, doi: 10.3390/s23208407.
[22] W. Gouda, S. Tahir, S. Alanazi, M. Almufareh, and G. Alwakid, "Unsupervised Outlier Detection in IOT Using Deep VAE," *Sensors*, vol. 22, no. 17, Sep. 2022, doi: 10.3390/s22176617.
[23] F. T. Liu, K. M. Ting and Z. -H. Zhou, "Isolation Forest," in *2008 Eighth IEEE International Conference on Data Mining*, Pisa, Italy, 2008, pp. 413-422, doi: 10.1109/ICDM.2008.17.
[24] W. Chua *et al.*, "Web Traffic Anomaly Detection Using Isolation Forest," *Informatics*, vol. 11, no. 4, Dec. 2024, doi: 10.3390/informatics11040083.
[25] M. Bachar, A. Khiat, and K. El Guemmat, "Hybrid Autoencoder and Isolation Forest for IoT Anomaly Detection with a Novel Model," *Engineering, Technology & Applied Science Research*, vol. 16, no. 1, pp. 31123–31129, Feb. 2026, doi: 10.48084/etasr.15288.
[26] P. Appiahene *et al.*, "Network intrusion detection using a hybrid graph-based convolutional network and transformer architecture," *PLoS One*, vol. 21, no. 1, p. e0340997, Jan. 2026, doi: 10.1371/journal.pone.0340997.

BIOGRAPHIES OF AUTHORS



Rajakumar Ponnumani is an assistant professor in the Department of Computer Science and Engineering at Parul Institute of Technology, Parul University, Vadodara, India, with 17 years of teaching experience. He is pursuing a Ph.D. at Galgotias University and has published over 25 international journal and conference papers. His research interests include artificial intelligence, cybersecurity, and cloud computing. He can be contacted at email: prajakumarkk@gmail.com.



Dr. Anusuya Venkatasamy is working as professor in the Department of Computer Science and Engineering, Ramco Institute of Technology, Tamilnadu, India. She received her bachelor degree in computer science and engineering from in 1999 and her master's degree from Government College of Engineering, Tirunelveli in 2006 and she received Ph.D. from Anna University, Chennai in 2020. She has published over 30 technical papers in international journals, international/national conferences. Her current research includes data science, augmented reality, and deep learning. She is a life member of Indian Society for Technical Education (ISTE) and Annual Member of ACM. She can be contacted at email: pgkrishanu@gmail.com.



Selvaraj Palanisamy assistant professor in Department of Computer Science and Engineering-AI at Mother Theresa Institute of Engineering and Technology, Andhra Pradesh, holds 17 years of teaching experience with an M.E. from Anna University, he has published 3 patents and 4 research papers, specializing in machine learning, networks, cloud computing, and cyber security. He can be contacted at email: selvajkf@gmail.com.



Padmanabhan P is an associate professor in the Department of Computer Science and Engineering at the Greater Noida Institute of Technology, Greater Noida, Uttar Pradesh, India. He has completed his M.Tech. and Ph.D. from reputed institutions. His research thrust is mainly focused on networks, wireless sensor networks, cloud computing, IoT, artificial intelligence, machine learning, deep learning, GenAI, NLP, computer vision, expert systems, and robotics. He can be contacted at email: padmanabhan5312@gmail.com.



Dr. Thenmozhi Elumalai is a professor in the Department of Information Technology at Panimalar Engineering College. With 24 years of teaching experience, she holds a Ph.D. and has authored 10 patents, 12 book chapters, and 30 research papers in renowned international journals and conferences. Her areas of expertise include cyber security, networks, and machine learning. She can be contacted at email: ethenmozhi22.pec@gmail.com.



Rajakumar Perumal assistant professor in Department of Computer Science and Applications, Sharda School of Computing Science and Engineering, Sharda University, Uttar Pradesh, India, holds 23 years of teaching experience and is pursuing a Ph.D. in Computer Science and Engineering at Shri Venkateshwara University. With an M.E. Computer Science and Engineering from Anna University, he has published 8 patents and 10 research papers, specializing in networks, cloud computing, software engineering, and machine learning. He can be contacted at email: rajkumar.jcet@gmail.com.