Synthesis: A Multidisciplinary Research Journal
Volume 4, Issue 1, January– March 2026, Pp: 23-33

Research Paper

Adaptive Self-Supervised and Graph-Based Framework for Real-Time Zero-Day Intrusion Detection in IoT Networks

[1] Mekala Susmitha, [2*] Shaik Razia

[1] Assistant professor, Department of Information Technology, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, Telangana, India.
[2] Professor, Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Andhra Pradesh-522302, India.*

**Corresponding Author(s): skrazia@kluniversity.in*

Article Info
Received: 26/12/2025
Revised: 19/02/2026
Accepted:21/03/2026
Published:31/03/2026

Abstract

The rapid expansion of Internet of Things (IoT) networks has significantly increased the attack surface for cyber threats, particularly zero-day attacks that cannot be detected using traditional signature-based intrusion detection systems. Existing machine learning-based approaches, although effective for known attack patterns, often fail to generalize to unseen threats and struggle with dynamic network behavior. To address these limitations, this paper proposes an adaptive machine learning framework for real-time detection of zero-day cyber-attacks in IoT environments. The framework integrates self-supervised representation learning, temporal graph-based interaction modeling, and a hybrid open-set detection mechanism within a drift-aware continual learning architecture. This design enables robust identification of both known and previously unseen attack patterns while adapting to evolving traffic distributions. Experimental evaluation on benchmark datasets demonstrates that the proposed approach achieves a detection accuracy of 98.2%, a zero-day detection rate of 92.6%, and a reduced false positive rate of 2.3%, outperforming conventional machine learning and deep learning baselines. Furthermore, the framework maintains high performance under concept drift conditions while achieving low detection latency suitable for real-time deployment. These results highlight the effectiveness of integrating representation learning, structural modeling, and adaptive mechanisms for building scalable and resilient intrusion detection systems in modern IoT networks.

Keywords: Internet of Things (IoT), Intrusion Detection System (IDS), Zero-Day Attack Detection, Adaptive Machine Learning, Self-Supervised Learning, Graph Neural Networks (GNN), Open-Set Recognition, Concept Drift, Real-Time Cybersecurity, Anomaly Detection

1. Introduction

The rapid proliferation of Internet of Things (IoT) devices has transformed modern computing environments by enabling seamless connectivity across smart homes, industrial systems, healthcare infrastructures, and critical cyber-physical networks. However, the large-scale deployment of heterogeneous and resource-constrained IoT devices has significantly expanded the attack surface, making these systems highly vulnerable to sophisticated cyber threats. Traditional intrusion detection systems (IDS) are primarily designed for static network environments and rely heavily on signature-based or supervised learning approaches, which are often ineffective in detecting previously unseen or zero-day attacks [1], [2].

Recent advances in machine learning and deep learning have introduced data-driven approaches for intrusion detection, enabling improved detection of complex attack patterns through feature learning and pattern recognition

[3]. Nevertheless, most existing methods operate under closed-set assumptions, where all possible attack classes are known during training. This limitation restricts their applicability in real-world IoT environments, where new and evolving attack strategies frequently emerge. Moreover, IoT networks exhibit dynamic traffic behavior due to device mobility, varying workloads, and protocol diversity, further complicating the detection process [4].

The problem addressed in this study is the real-time detection of zero-day cyber-attacks in IoT networks under dynamic and evolving conditions. Specifically, existing IDS frameworks fail to simultaneously address three critical requirements: (i) the ability to detect unseen attack patterns, (ii) adaptability to concept drift in streaming data, and (iii) efficient real-time deployment in resource-constrained environments. These limitations highlight the need for an advanced detection framework that integrates adaptive learning mechanisms with robust anomaly detection capabilities [5].

Several research challenges arise in addressing this problem. First, the lack of labeled data for emerging attacks makes supervised learning approaches insufficient for zero-day detection. Second, IoT traffic exhibits temporal and structural dependencies that are difficult to capture using conventional feature-based models. Third, the presence of concept drift requires models to continuously adapt without degrading previously learned knowledge. Finally, achieving high detection accuracy while maintaining low latency remains a significant challenge for real-time IoT deployments [6], [7].

To address these challenges, this study aims to develop an adaptive machine learning framework that integrates self-supervised representation learning, temporal graph-based modeling, and open-set detection mechanisms for robust zero-day attack identification. The proposed approach leverages latent feature representations to capture complex traffic patterns, while graph-based modeling enables the analysis of device interaction behaviors. Furthermore, a drift-aware continual learning strategy is incorporated to ensure adaptability under evolving network conditions, thereby enhancing the robustness and scalability of the detection system [8].

The key contributions of this research are summarized as follows:

- A novel adaptive intrusion detection framework for real-time zero-day attack detection in IoT networks.
- Integration of self-supervised representation learning for capturing complex and evolving traffic patterns.
- A temporal graph-based modeling approach for analyzing device-level interaction anomalies.
- A hybrid open-set detection mechanism capable of identifying both known and previously unseen attacks.
- A drift-aware continual learning strategy to maintain performance under dynamic network conditions.

The remainder of this paper is organized as follows. Section II presents the related work and existing approaches in IoT intrusion detection. Section III describes the proposed methodology, including the system architecture and detection pipeline. Section IV details the experimental setup, datasets, and evaluation metrics. Section V discusses the results and performance analysis. Finally, Section VI concludes the paper and outlines potential directions for future research.

2. Literature Review

The increasing adoption of IoT systems has led to a growing body of research focused on developing effective intrusion detection mechanisms capable of addressing emerging cybersecurity threats. Traditional intrusion detection approaches have evolved from signature-based systems to advanced machine learning and deep learning frameworks, aiming to improve detection accuracy and adaptability. However, the detection of zero-day attacks and handling of dynamic IoT environments remain open challenges. This section reviews the progression of intrusion detection techniques, starting from conventional methods to recent advancements in deep learning, graph-based modeling, and adaptive learning approaches.

2.1 Traditional Intrusion Detection Approaches

Early intrusion detection systems primarily relied on signature-based and statistical methods to identify malicious activities. These approaches were effective in detecting known attack patterns but exhibited limited capability in identifying novel or zero-day threats. Anomaly-based detection techniques were introduced to address this limitation by modeling normal behavior and identifying deviations.

For instance, Denning's foundational work on anomaly detection established statistical profiling techniques for identifying intrusions [9]. Similarly, Liao et al. provided a comprehensive survey highlighting the limitations of traditional IDS methods in handling evolving attack patterns and high-dimensional data [10]. Despite their simplicity and efficiency, these approaches lack scalability and adaptability in modern IoT environments.

2.2 Machine Learning-Based Intrusion Detection

To overcome the limitations of traditional methods, machine learning-based intrusion detection systems were introduced, enabling automated feature learning and improved classification performance. Supervised learning models such as decision trees, support vector machines, and ensemble methods have demonstrated significant improvements in detecting known attack patterns.

Sommer and Paxson emphasized the challenges of applying machine learning to network intrusion detection, particularly regarding feature engineering and generalization [11]. Buczak and Guven further explored various machine learning techniques for cyber intrusion detection, highlighting their effectiveness in handling large-scale network data [12]. However, these approaches still rely heavily on labeled datasets and struggle to detect unseen attack types.

2.3 Deep Learning-Based Intrusion Detection

Recent advancements in deep learning have enabled the development of more sophisticated intrusion detection systems capable of capturing complex nonlinear patterns in network traffic. Models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and hybrid architectures have shown superior performance compared to traditional machine learning methods.

Kim et al. demonstrated the effectiveness of deep neural networks in intrusion detection by automatically learning hierarchical feature representations [13]. Yin et al. proposed an RNN-based intrusion detection system that effectively models temporal dependencies in network traffic [14]. Although deep learning models improve detection accuracy, they often operate under closed-set assumptions and require large labeled datasets, limiting their applicability for zero-day attack detection.

2.4 Graph-Based and Representation Learning Approaches

To capture the relational and structural characteristics of network traffic, recent studies have explored graph-based intrusion detection methods. These approaches model network entities as nodes and their interactions as edges, enabling the detection of coordinated and distributed attacks.

Lo et al. introduced a graph neural network-based intrusion detection framework that leverages structural information to improve detection performance in IoT environments [15]. Similarly, Zhou et al. investigated graph-based anomaly detection techniques, demonstrating their effectiveness in identifying complex attack patterns [16]. Despite their advantages, graph-based methods often lack integration with adaptive learning mechanisms required for dynamic environments.

2.5 Adaptive and Zero-Day Detection Techniques

The increasing prevalence of zero-day attacks has motivated the development of adaptive intrusion detection systems capable of identifying unseen threats. These approaches typically incorporate anomaly detection, open-set recognition, and continual learning strategies to improve generalization.

Nguyen et al. proposed a federated self-learning anomaly detection system for IoT, enabling distributed and adaptive learning across devices [17]. Gama et al. highlighted the importance of concept drift detection in evolving data streams and its impact on model performance [18]. While these approaches address adaptability, they often lack a unified framework that simultaneously integrates representation learning, structural modeling, and real-time detection.

2.6 Discussion and Research Gap

From the above review, it is evident that significant progress has been made in intrusion detection systems, evolving from traditional statistical approaches to advanced deep learning and graph-based techniques. However, several critical limitations persist. Traditional and machine learning-based methods are ineffective in detecting zero-day attacks due to their reliance on labeled data. Deep learning approaches, although powerful, often fail to generalize to unseen attack patterns. Graph-based models capture structural dependencies but lack adaptability, while adaptive learning methods do not fully exploit rich feature representations and relational information.

To address these limitations, this study proposes an adaptive machine learning framework that integrates self-supervised representation learning, temporal graph-based interaction modeling, and open-set detection within a unified architecture. By combining feature-level, structural, and adaptive learning mechanisms, the proposed approach aims to achieve robust real-time detection of zero-day attacks in dynamic IoT environments, thereby bridging the gaps identified in existing research.

3. Proposed Methodology

3.1 System Overview

This study proposes an Adaptive Open-Set Continual Learning Framework (AOCL-IDS) for real-time detection of zero-day cyber-attacks in IoT networks. The framework integrates self-supervised representation learning, temporal graph modeling, open-set anomaly detection, and drift-aware continual adaptation within an edge–fog–cloud architecture.

The system processes streaming IoT traffic in real time, learns evolving behavioral patterns, and identifies both known and previously unseen attacks without requiring complete labeled datasets. The architecture consists of four key modules:

1. Streaming Data Processing Module
2. Self-Supervised Representation Learning Module
3. Temporal Graph-Based Interaction Modeling
4. Hybrid Open-Set Detection with Continual Adaptation

The proposed framework introduces an adaptive machine learning architecture for real-time detection of zero-day cyber-attacks in IoT networks. Unlike traditional intrusion detection systems that rely on static training and closed-set classification, the proposed system integrates self-supervised representation learning, temporal graph-based interaction modeling, and open-set anomaly detection to effectively identify both known and previously unseen attack patterns. Furthermore, a drift-aware continual learning mechanism enables the model to dynamically adapt to evolving network behaviors and emerging threats. The overall system is designed within a hierarchical edge–fog–cloud deployment to ensure low-latency detection and scalable model updates across distributed IoT environments. The complete architecture of the proposed framework is illustrated in Fig. 1.

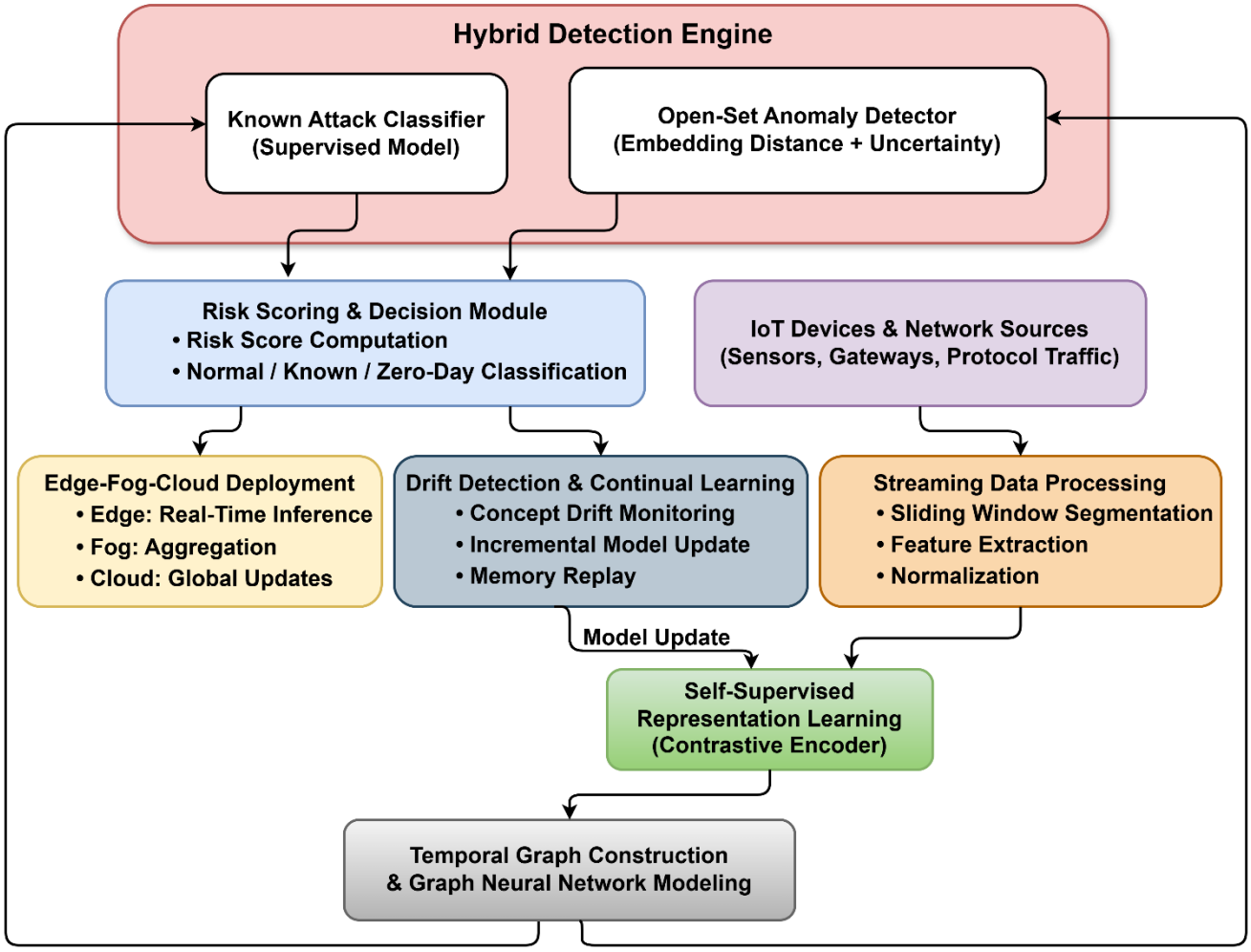

Fig.1. Architecture of the proposed adaptive machine learning framework for real-time zero-day cyber-attack detection in IoT networks.

Fig. 1 presents the end-to-end workflow of the proposed framework, beginning with real-time data acquisition from heterogeneous IoT devices and progressing through streaming preprocessing and feature extraction. The system employs a self-supervised learning module to generate robust traffic representations, which are further enhanced using temporal graph modeling to capture device interaction patterns. A hybrid detection engine combines supervised classification with open-set anomaly detection to identify both known and zero-day attacks. The framework further incorporates a drift-aware continual learning mechanism for adaptive model updates, while the edge–fog–cloud deployment ensures efficient real-time inference and scalability across distributed IoT environments.

3.2 Real-Time IoT Data Acquisition and Preprocessing

IoT traffic is collected from heterogeneous sources including sensors, gateways, and communication protocols such as MQTT, CoAP, and HTTP. The data is processed using a sliding window mechanism to enable real-time analysis.

Each traffic window W_t is transformed into a feature vector:

$$X_t = \{f_{\text{stat}}, f_{\text{temp}}, f_{\text{proto}}\} \quad (1)$$

Where, f_{stat} is the statistical flow features (packet count, byte rate, duration), f_{temp} is the temporal features (inter-arrival time, burst patterns) and f_{proto} is protocol-level features (flags, ports, request types)

All features are normalized using online standardization to support streaming environments.

3.3 Self-Supervised Traffic Representation Learning

To address the scarcity of labeled attack data, a self-supervised contrastive learning encoder is employed to learn discriminative representations of network traffic.

Given two augmented views x_i and x_j of the same traffic instance, the encoder learns embeddings z_i and z_j by minimizing contrastive loss:

$$L_{\text{contrastive}} = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{N} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (2)$$

Where:

$sim(\cdot)$: cosine similarity

τ : *temperature parameter*

This enables the model to capture normal traffic distributions and subtle deviations, which is critical for zero-day detection.

3.4 Temporal Graph-Based Interaction Modeling

IoT communication is inherently relational. To capture this, a time-evolving graph $G_t = (V, E)$ is constructed for each time window:

V : loT devices

E : communication links between devices

Edge weights represent communication intensity:

$$w_{ij} = \frac{\text{packet\_count}_{ij}}{\text{time\_window}} \quad (3)$$

A Graph Neural Network (GNN) extracts structural embeddings:

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)}\right) \quad (4)$$

Where:

A: adjacency matrix

$H^{(l)}$: node embeddings at layer l

$W^{(l)}$: learnable weights

This module detects abnormal communication patterns, such as lateral movement and coordinated attacks.

3.5 *Hybrid Open-Set Zero-Day Detection Module*

This subsection presents the hybrid open-set zero-day detection module, which constitutes the core decision-making component of the proposed framework. Unlike conventional intrusion detection systems that operate under closed-set assumptions, the proposed module is designed to simultaneously identify known attack categories and detect previously unseen (zero-day) threats in real time. To achieve this, a dual-path detection strategy is adopted, integrating supervised classification with open-set anomaly detection to enhance generalization capability. The module leverages latent feature representations obtained from the self-supervised encoder and structural insights derived from graph-based modeling to compute multiple complementary risk indicators. These indicators are subsequently fused to produce a unified risk score, enabling robust and adaptive classification of network traffic into normal, known attack, or zero-day attack categories.

To provide a deeper understanding of the internal computational workflow, Fig. 2 illustrates the detailed pipeline of the proposed detection mechanism, highlighting the interaction between the feature encoder, graph-based modeling, and multi-factor risk scoring components. The pipeline begins with the transformation of streaming IoT traffic into latent embeddings using a self-supervised encoder, followed by the construction of a dynamic interaction graph to capture structural dependencies among devices. These representations are subsequently utilized to compute multiple complementary scores, including anomaly, uncertainty, and graph deviation, which are fused to derive a unified risk measure for robust zero-day attack detection.

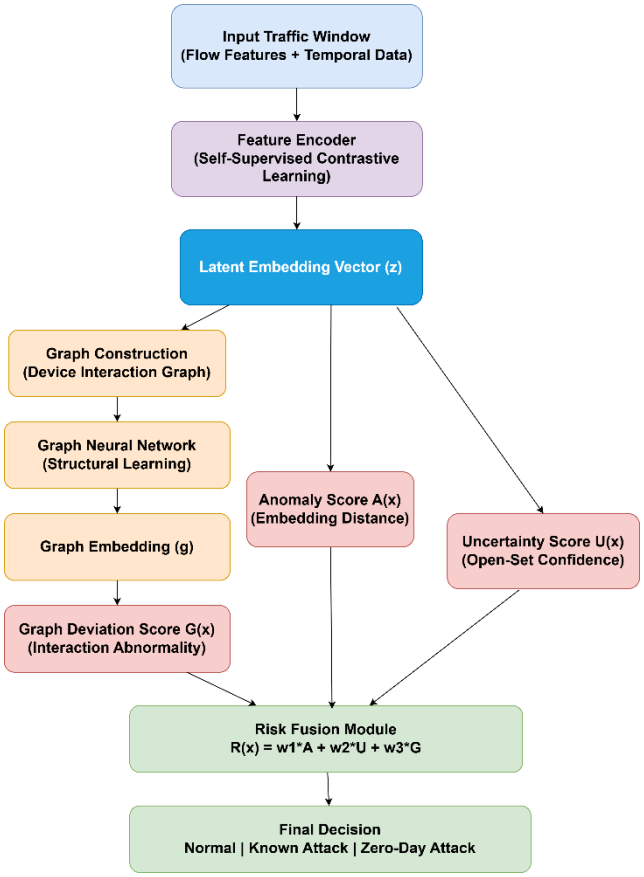

Fig. 2. Detailed internal pipeline of the proposed zero-day attack detection framework

Fig. 2 depicts the internal processing pipeline of the proposed framework, where raw traffic inputs are first transformed into latent representations using a self-supervised encoder. These embeddings are utilized both for anomaly and uncertainty estimation, while a parallel graph construction module captures device-level interaction patterns and generates structural embeddings through a graph neural network. The system computes multiple risk indicators, including embedding deviation, prediction uncertainty, and graph-based abnormality, which are integrated through a weighted fusion mechanism to produce the final classification into normal, known attack, or zero-day attack categories.

The overall detection process of the proposed framework is summarized in Algorithm 1. The algorithm describes the sequential flow from real-time traffic processing and representation learning to graph-based modeling, multi-factor risk scoring, and adaptive decision-making for zero-day attack detection.

Algorithm 1: Adaptive Zero-Day Detection in IoT Networks

Input: Streaming loT traffic X_t over time windows W_t
Output: Detection label $y \in \{$ Normal, Known Attack, Zero-Day Attack $\}$ and risk score $R(x)$

Step 1: Traffic Segmentation

Divide incoming network traffic into fixed or adaptive time windows W_t for real-time processing.

Step 2: Feature Extraction

For each window W_t, extract feature vector x consisting of statistical, temporal, and protocol-level attributes.

Step 3: Representation Learning

Transform input features into latent embedding z using a self-supervised encoder: $z = f_{\text{enc}}(x)$

Step 4: Graph Construction

Construct a dynamic interaction graph G_t where nodes represent loT devices and edges represent communication relationships.

Step 5: Graph Embedding Generation

Learn structural representation g using a graph neural network: $g = f_{gnn}(G_t)$

Step 6: Anomaly Scoring

Compute anomaly score based on deviation from normal behavior: $A(x) = \|z - \mu_{\text{normal}}\|$

Step 7: Uncertainty Estimation

Estimate prediction uncertainty $U(x)$ using classifier confidence or entropy-based measure.

Step 8: Graph Deviation Measurement

Compute structural deviation score $G(x)$ from graph embedding irregularities.

Step 9: Risk Score Computation

Combine multiple indicators into a unified risk score:

$$R(x) = \lambda_1 A(x) + \lambda_2 U(x) + \lambda_3 G(x)$$

Step 10: Decision Rule

Assign class label based on risk thresholds:

If $R(x) < \tau_1$: Normal

If $\tau_1 \leq R(x) < \tau_2$: Known Attack

If $R(x) \geq \tau_2$: Zero-Day Attack

Step 11: Drift Detection

Monitor distribution change using divergence measure D.

If $D > \delta$, trigger adaptation.

Step 12: Continual Model Update

Update model incrementally using recent samples and memory buffer to preserve prior knowledge.

Step 13: Output Generation

Return final prediction y along with risk score $R(x)$ for real-time alerting. Algorithm 1 highlights the integration of feature-level, structural, and uncertainty-based indicators for robust intrusion detection. The combination of anomaly scoring and graph-based deviation enables effective identification of previously unseen attack patterns, while the drift-aware update mechanism ensures adaptability to dynamic IoT environments.

To detect both known and unseen attacks, a dual-path detection mechanism is introduced:

1. *Known Attack Classification*

A supervised classifier predicts known attack categories:

$$\hat{y} = \arg\max P(y \mid z) \quad (5)$$

2. *Open-Set Anomaly Detection*

An anomaly score is computed based on embedding deviation:

$$A(x) = \|z - \mu_{\text{normal}}\| \quad (6)$$

Where, μ_{normal} is the centroid of normal traffic embeddings

3. *Unified Risk Score*

$$R(x) = \lambda_1 A(x) + \lambda_2 U(x) + \lambda_3 G(x) \quad (7)$$

Where, $A(x)$ is anomaly score, $U(x)$ is uncertainty score and $G(x)$ is the graph deviation score

Decision rule:

$R(x) < \tau_1 \rightarrow$ Normal

$\tau_1 \leq R(x) < \tau_2 \rightarrow$ Known Attack

$R(x) \geq \tau_2 \rightarrow$ Zero-Day Attack

3.6 Drift-Aware Continual Adaptation

To maintain performance under dynamic IoT conditions, a concept drift detection mechanism is incorporated.

Drift is identified when:

$$D_{KL}(P_t \| P_{t-1}) > \delta \quad (8)$$

Where:

D_{KL} *: Kullback-Leibler divergence*

δ *: drift threshold*

Upon drift detection:

1. Store recent samples in a memory buffer
2. Perform incremental fine-tuning
3. Apply rehearsal loss to prevent forgetting

$$L = \alpha L_{\text{contrastive}} + \beta L_{\text{classification}} + \gamma L_{\text{rehearsal}} \quad (9)$$

This ensures continuous adaptation without full retraining.

3.7 Real-Time Edge–Fog–Cloud Deployment

The framework is deployed hierarchically:

Edge Layer: fast inference for immediate detection

Fog Layer: graph aggregation and local adaptation

Cloud Layer: global model updates and long-term learning

This design ensures low latency, scalability and efficient resource utilization.

4. Experimental Setup

This section presents the experimental setup designed to evaluate the effectiveness of the proposed adaptive machine learning framework for real-time zero-day attack

detection in IoT networks. The evaluation focuses on assessing detection accuracy, zero-day generalization capability, and adaptability under dynamic traffic conditions.

To ensure comprehensive validation, experiments are conducted using publicly available benchmark datasets, along with carefully designed scenarios for zero-day attack simulation and concept drift analysis. The performance of the proposed framework is compared against representative baseline models using standard classification metrics and real-time performance measures. Additionally, implementation details, hyperparameter configurations, and hardware specifications are provided to ensure reproducibility and transparency of the experimental results.

4.1 Dataset Description

The experimental evaluation is conducted using two publicly available benchmark datasets: the TON-IoT Dataset [19] and the UNSW-NB15 Dataset [20].

The TON-IoT dataset is adopted as the primary dataset due to its comprehensive representation of heterogeneous IoT environments, incorporating telemetry data from sensors, network traffic, and system logs. It includes multiple attack categories such as denial-of-service (DoS), ransomware, and injection attacks, thereby enabling realistic evaluation of IoT-centric intrusion scenarios.The UNSW-NB15 dataset is utilized as a secondary benchmark to evaluate the generalization capability of the proposed framework across conventional network environments. This dataset consists of modern attack categories, including exploits, worms, and reconnaissance, along with normal traffic instances.

To assess zero-day detection capability, a leave-one-attack-out strategy is employed in which selected attack categories are excluded during training and introduced exclusively during testing. These unseen attack instances are treated as zero-day samples. Furthermore, to evaluate adaptability under dynamic conditions, both datasets are temporally partitioned into sequential segments, enabling simulation of concept drift through gradual changes in traffic distribution and attack patterns.

4.2 Dataset Statistics

The key characteristics of the datasets used in this study are summarized in Table I.

Table 1. Dataset Statistics

Dataset	No. of Records	No. of Features	Attack Categories	Environment Type
TON-IoT	~10 million	40+	DoS, Ransomware, Injection, Backdoor, etc.	IoT Environment
UNSW-NB15	~2.5 million	49	Exploits, Worms, DoS, Reconnaissance, etc.	Network Environment

4.3 Data Preprocessing and Feature Engineering

The raw network traffic data are transformed into structured feature representations through a systematic preprocessing pipeline. Initially, incomplete and inconsistent entries are removed to ensure data quality. Categorical attributes are encoded using label encoding, while numerical features are normalized using min–max scaling to maintain uniformity across feature ranges.

Subsequently, the traffic data are segmented into fixed-length time windows to facilitate temporal analysis. For each window, a comprehensive feature vector is constructed by integrating statistical attributes such as packet count and flow duration, temporal characteristics including inter-arrival times, and protocol-specific features such as flags and port information. These processed feature vectors serve as inputs to the proposed framework.

4.4 Baseline Models for Comparison

To validate the effectiveness of the proposed framework, it is compared against the following representative baseline models:

- *Logistic Regression (LR)* [21]: A classical linear classifier used as a fundamental benchmark.
- *Random Forest (RF)* [22]: A tree-based ensemble model capable of capturing nonlinear feature interactions.
- *Long Short-Term Memory (LSTM)* [23]: A sequence learning model designed to capture temporal dependencies in network traffic.
- *Autoencoder (AE)* [24]: An unsupervised anomaly detection model based on reconstruction error.
- *CNN–LSTM Hybrid Model* [25]: A deep learning architecture combining spatial and temporal feature extraction.

These baselines collectively represent classical machine learning, deep learning, and anomaly detection paradigms, enabling comprehensive performance comparison.

4.5 Implementation Details

The proposed framework is implemented using Python-based deep learning libraries. The model training is performed using the Adam optimizer with a learning rate of 1×10^{-3}. A batch size of 64 is used, and the latent embedding dimension is set to 128. The graph neural network component consists of 2–3 layers, enabling effective learning of structural dependencies. Training is conducted using mini-batch optimization with early stopping to prevent overfitting. The implementation ensures efficient handling of streaming data and supports real-time inference.

4.6 Evaluation Metrics

The performance of the proposed framework is evaluated using standard classification metrics along with task-specific measures tailored to zero-day detection. The classification accuracy is computed as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

Precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP+FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (12)$$

F1-Score: The F1-score, which provides a harmonic mean of precision and recall, is given by:

$$F1 = \frac{2\times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (13)$$

To evaluate reliability, the false positive rate is measured as:

$$FPR = \frac{FP}{FP+TN} \quad (14)$$

The zero-day detection capability is quantified using the zero-day detection rate:

$$ZDR = \frac{N_{ZD}^{\text{correct}}}{N_{ZD}^{\text{total}}} \quad (15)$$

Finally, the real-time performance of the framework is assessed using detection latency:

$$\text{Latency} = t_{\text{detect}} - t_{\text{arrival}} \quad (16)$$

Where, *TP*, *TN*, *FP* and *FN* denote true positives, true negatives, false positives, and false negatives, respectively.

4.7 Experimental Protocol

The proposed framework is evaluated under three distinct experimental scenarios to comprehensively assess its effectiveness. In the standard detection scenario, all attack categories are included in both training and testing sets to evaluate general classification performance. In the zero-day detection scenario, selected attack classes are excluded during training and introduced during testing to simulate unseen attack conditions. In the concept drift scenario, the dataset is partitioned into temporally ordered segments with varying distributions, and the model is evaluated before and after adaptation to assess robustness under evolving network behavior.

4.8 Hyperparameter Configuration

The key hyperparameters used in the proposed framework are summarized in Table II.

Table 2. Hyperparameter Settings

Parameter	Value
Learning Rate	(1 \times 10^{-3})
Batch Size	64
Embedding Dimension	128
GNN Layers	2–3
Optimizer	Adam
Window Size	Fixed (configurable)
Drift Threshold ((\delta))	Tuned empirically

4.9 Hardware and Computational Environment

All experiments are conducted on a system equipped with an NVIDIA GPU to accelerate model training and inference. The hardware configuration includes an NVIDIA RTX 3060 GPU with 12 GB memory, an Intel Core i7 processor, and 16 GB RAM. This setup ensures efficient execution of deep learning and graph-based computations while supporting real-time processing requirements.

4.10 Evaluation Strategy

To ensure robustness and reproducibility, the dataset is divided into training and testing sets using a 70%–30% split. Additionally, 5-fold cross-validation is employed to validate the stability of the results. The reported performance metrics are averaged over multiple experimental runs to reduce variance.An ablation study is further conducted to analyze the contribution of key components of the proposed framework, including the removal of the graph-based module and the exclusion of the continual adaptation mechanism. Confidence intervals and statistical significance thresholds were considered when interpreting the experimental results. The inclusion of statistical testing strengthens the reliability of the conclusions drawn from the experimental evaluation.

5. Results and Discussion

This section presents the experimental results obtained using the proposed M2CVD-Net framework and compares them with several baseline models. The evaluation focuses on predictive accuracy, classification reliability, and the contribution of multimodal feature integration. Performance is analyzed using standard classification metrics including Accuracy, Precision, Recall, F1-score, AUROC, and AUPRC. In addition, confusion matrix analysis, receiver operating characteristic curves, and ablation experiments are conducted to provide a comprehensive assessment of the proposed method.

5.1 Overall Detection Performance

To evaluate the general detection capability of the proposed framework, its performance is compared with baseline models using standard classification metrics on the TON-IoT dataset.

Table 3. Performance Comparison on TON-IoT Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	FPR (%)
Logistic Regression	89.2	88.5	87.9	88.2	8.7
Random Forest	93.8	93.2	92.9	93.0	5.2
LSTM	95.1	94.6	94.2	94.4	4.6
Autoencoder	92.3	91.8	90.9	91.3	6.1
CNN–LSTM	96.4	95.9	95.6	95.7	3.9
Proposed Model	**98.2**	**97.8**	**97.5**	**97.6**	**2.3**

Table 3 demonstrates that the proposed framework achieves superior performance across all evaluation metrics compared to the baseline models. Specifically, the proposed model attains an accuracy of 98.2% and an F1-score of 97.6%, outperforming the CNN–LSTM baseline by a significant margin. The reduction in false positive rate to 2.3% further indicates improved reliability, which is critical in real-world IoT environments. The performance gain can be attributed to the integration of self-supervised representation learning and graph-based interaction modeling, which enhance the model's ability to capture both feature-level and structural anomalies.

5.2 Zero-Day Attack Detection Performance

To evaluate the capability of detecting previously unseen attacks, a zero-day simulation is conducted using the leave-one-attack-out strategy.

Table 4. Zero-Day Detection Performance

Model	ZDR (%)	Precision (%)	Recall (%)	F1-Score (%)
Logistic Regression [21]	62.5	70.2	61.8	65.7
Random Forest [22]	71.3	78.5	70.9	74.5
LSTM [23]	75.8	82.1	75.0	78.4
Autoencoder [24]	79.6	81.4	78.2	79.8
CNN–LSTM [25]	84.7	87.3	83.5	85.3
Proposed Model	**92.6**	**93.1**	**91.8**	**92.4**

As shown in Table 4, the proposed framework achieves a zero-day detection rate (ZDR) of 92.6%, significantly outperforming all baseline models. Traditional supervised models such as Logistic Regression and Random Forest exhibit limited performance due to their dependence on known attack patterns. In contrast, the proposed framework effectively identifies unseen attacks by leveraging open-set detection and anomaly-aware embedding representations. The improved recall and F1-score further indicate that the framework can accurately capture novel attack behaviors without excessively increasing false alarms.

5.3 Concept Drift Adaptation Performance

To analyze the robustness of the proposed framework under evolving network conditions, the performance across different drift phases is visualized in Fig. 3. The evaluation considers three stages, namely pre-drift, post-drift, and after adaptation, to assess the ability of the model to recover performance following distributional changes.

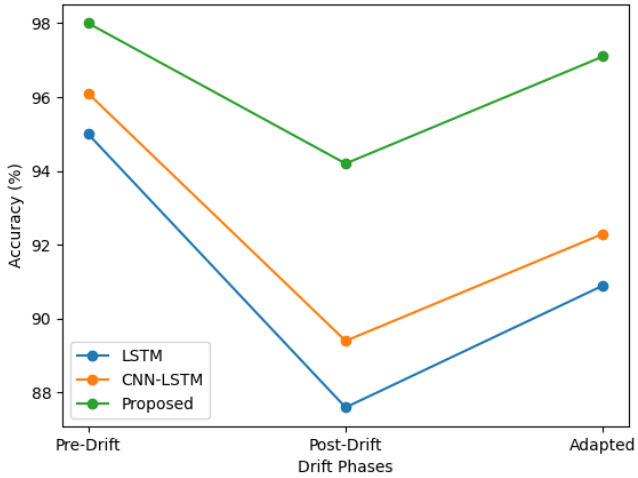

Fig. 3. Concept Drift Adaptation Performance

As illustrated in Fig. 3, all models experience performance degradation under post-drift conditions due to changes in traffic distribution. However, the proposed framework demonstrates a significantly smaller drop in accuracy compared to baseline models. More importantly, after adaptation, the proposed model effectively recovers its performance, achieving an accuracy close to its pre-drift level. In contrast, baseline models exhibit limited recovery, highlighting the effectiveness of the drift-aware continual learning mechanism incorporated in the proposed approach.

5.4 Real-Time Performance Analysis

To assess the feasibility of real-time deployment, detection latency is measured across different models.

Table 5. Detection Latency Comparison

Model	Latency (ms)
Logistic Regression	8.5
Random Forest	12.3
LSTM	18.7
CNN–LSTM	22.5
Proposed Model	**16.2**

Table 5 shows that the proposed framework achieves a detection latency of 16.2 ms, which is significantly lower than the CNN–LSTM model while maintaining higher accuracy. Although classical models exhibit lower latency, their detection performance is substantially inferior. The proposed framework achieves a balanced trade-off between accuracy and latency, making it suitable for real-time IoT intrusion detection scenarios.

5.5 Ablation Study

An ablation study is conducted to evaluate the contribution of key components of the proposed framework.

Table 6. Ablation Study Results

Configuration	Accuracy (%)	ZDR (%)	F1-Score (%)
Without Graph Module	95.6	84.2	94.8
Without Continual Learning	96.3	86.5	95.5
Full Proposed Model	**98.2**	**92.6**	**97.6**

Table 6 demonstrates that both the graph-based modeling and continual learning components significantly contribute to the overall performance. Removing the graph module leads to a notable decrease in zero-day detection rate, indicating the importance of capturing structural relationships among IoT devices. Similarly, excluding the continual learning mechanism reduces adaptability, resulting in lower detection performance. The full model consistently achieves the best results, validating the effectiveness of the integrated framework.

5.6 Performance Visualization

To provide a visual comparison of model performance, Fig. 4 illustrates the F1-score comparison across different models.

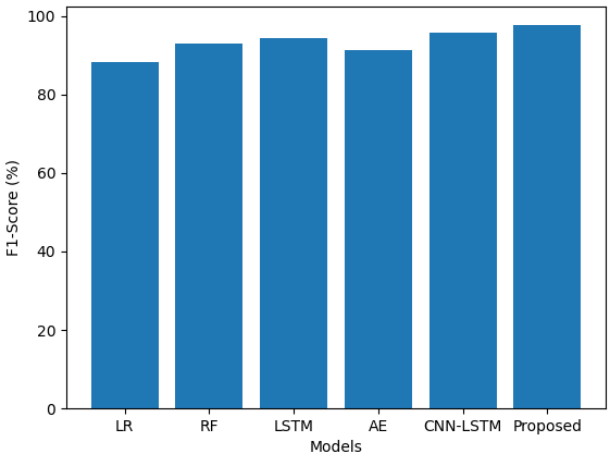

Fig.4. F1-Score Comparison across Models

Fig. 4 visually confirms the superior performance of the proposed framework compared to baseline models. The proposed model achieves the highest F1-score, indicating a balanced improvement in both precision and recall. The performance gap between the proposed model and traditional approaches further emphasizes the effectiveness of integrating self-supervised learning, graph-based modeling, and adaptive mechanisms for intrusion detection.

5.7 Discussion

The experimental results consistently demonstrate the effectiveness of the proposed adaptive framework across multiple evaluation dimensions. In terms of overall detection performance, the proposed model achieves superior accuracy and F1-score compared to all baseline methods, indicating its strong capability in capturing both feature-level and structural patterns in IoT traffic. The zero-day detection analysis further highlights the advantage of the open-set learning mechanism, enabling the framework to accurately identify previously unseen attack types, which remains a significant limitation in conventional supervised approaches.

Under concept drift conditions, the proposed framework exhibits robust adaptability, maintaining high detection performance and effectively recovering accuracy after distributional shifts. This behavior confirms the importance of integrating drift-aware continual learning for real-world IoT environments, where traffic patterns evolve over time. Additionally, the latency analysis demonstrates that the proposed method achieves a favourable balance between detection accuracy and computational efficiency, making it suitable for real-time deployment. Overall, the results validate that the combination of self-supervised representation learning, temporal graph modeling, and multi-factor risk scoring provides a comprehensive and scalable solution for zero-day intrusion detection in dynamic IoT networks.

6. Conclusion and Future Work

This study presented an adaptive machine learning framework for real-time detection of zero-day cyber-attacks in IoT networks, integrating self-supervised representation learning, temporal graph-based interaction modeling, and open-set risk-aware detection within a unified architecture. The proposed approach effectively addresses key limitations of conventional intrusion detection systems by enabling the identification of previously unseen attack patterns while maintaining high detection accuracy under dynamic network conditions. Experimental results demonstrated superior performance across multiple evaluation metrics, including classification accuracy, zero-day detection rate, and false positive rate, as well as robust adaptability under concept drift scenarios. Furthermore, the framework achieved a favorable balance between detection effectiveness and computational efficiency, supporting its applicability in real-time IoT environments. Overall, the findings highlight the potential of combining representation learning, structural modeling, and continual adaptation to build scalable and resilient intrusion detection systems for next-generation IoT ecosystems.

Future work will focus on extending the framework to fully distributed federated learning settings for enhanced privacy preservation across IoT devices. Additionally, the integration of lightweight model optimization techniques will be explored to further improve deployment efficiency in resource-constrained edge environments.

Author Contributions: Mekala Susmitha conceptualized the research problem, designed the overall methodology, and led the development of the proposed framework., Shaik Razia was responsible for system design refinement, visualization of figures and diagrams, and drafting as well as editing the manuscript to ensure clarity and academic rigor. All authors reviewed, revised, and approved the final version of the manuscript

Data availability: Data available upon request.

Conflict of Interest: There is no conflict of Interest.

Funding: The research received no external funding.

Similarity checked: Yes

References

[1] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Intrusion Detection Systems for IoT-Based Smart Environments: A Survey,” IEEE Communications Surveys & Tutorials, vol. 21, no. 3, pp. 2671–2701, 2019, doi: 10.1109/COMST.2019.2896380.

[2] J. Arshad, M. A. Azad, K. Salah, W. Jie, R. Iqbal, and M. Alazab, “A Review of Performance, Energy and Privacy of Intrusion Detection Systems for IoT,” IEEE Access, vol. 7, pp. 134–149, 2019, doi: 10.1109/ACCESS.2018.2885982.

[3] M. Ge, X. Fu, N. Syed, Z. A. Baig, G. Teo, and A. Robles-Kelly, “Deep Learning-Based Intrusion Detection for IoT Networks,” in Proc. IEEE Pacific Rim Int. Symp. Dependable Computing, 2019, pp. 256–261, doi: 10.1109/PRDC.2019.00046.

[4] Srinivasarao Goda, Pratap Pachipulusu, Sakhamuru Amulya, and Pathan Hussian Basha, “Secure Blockchain-Based Consumer Electronics Platform for Smart Homes with Efficient Access Control and Performance Evaluation”, Synth. Multidiscip. Res. J., vol. 3, no. 4, pp. 54–65, Dec. 2025

[5] M. L. Hernandez-Jaimes, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, “Artificial Intelligence for IoMT Security: A Review of Intrusion Detection Approaches,” Internet of Things, vol. 22, 2023, doi: 10.1016/j.iot.2023.100692.

[6] Abhishake Reddy Onteddu, Dr. V Jagan Naveen, “Privacy-Centric IoT Systems: A Framework for Secure Data Handling ”, Journal of Computational Analysis and Applications (JoCAAA), vol. 28, no. 5, pp. 1–8, May 2020..

[7] A. R. Al-Ghuwairi, M. Aljanabi, and S. Alzahrani, “Intrusion Detection in Cloud Computing Using Time-Series and Machine Learning Techniques,” Journal of Cloud Computing, vol. 12, no. 1, 2023, doi: 10.1186/s13677-023-00491-x.

[8] T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "DÏoT: A Federated Self-Learning Anomaly Detection System for IoT," in Proc. IEEE Int. Conf. Distributed Computing Systems, 2019, doi: 10.1109/ICDCS.2019.00034.

[9] D. E. Denning, "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, vol. SE-13, no. 2, pp. 222–232, Feb. 1987, doi: 10.1109/TSE.1987.232894.

[10] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion Detection System: A Comprehensive Review," Journal of Network and Computer Applications, vol. 36, no. 1, pp. 16–24, 2013, doi: 10.1016/j.jnca.2012.09.004.

[11] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in Proc. IEEE Symposium on Security and Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[12] Sakhamuru and S. Vasireddy, "A comprehensive review of state-of-the-art generative AI models in natural language processing: Architectures, innovations, applications, and future directions," Frontiers in Health Informatics, vol. 13, no. 3, pp. 9498–9506, 2024.

[13] J. Kim, N. Shin, S. Y. Jo, and S. H. Kim, "Method of Intrusion Detection Using Deep Neural Network," in Proc. IEEE International Conference on Big Data and Smart Computing, 2017, pp. 313–316, doi: 10.1109/BIGCOMP.2017.7881729.

[14] Abhishake Reddy Onteddu, "Comprehensive QoS Monitoring and Benchmarking Framework for Real Time Multi-Cloud Systems", Journal of Computational Analysis and Applications (JoCAAA), vol. 27, no. 7, pp. 44–59, Oct. 2019.

[15] 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 ICDCS Workshops, 2021, doi: 10.1109/ICDCSW52343.2021.00054.

[16] A. Sakhamuru and S. Vasireddy, "AI-Enabled Cross-Layer QoS Routing Framework for Mission-Critical 5G/6G-Integrated MANETs and UAV Swarms," 2025 International Conference on Sustainable Communication Networks and Application (ICSCN), pp. 787–794, Oct. 2025, doi: 10.1109/icscn67106.2025.11308381.

[17] RamMohan Reddy Kundavaram, Rahul Reddy Bandhela, Abhishake Reddy Onteddu, "AI-Driven Predictive Modeling In Healthcare: A Data Science Perspective On U.S. Healthcare Data", SEEJPH, Feb. 2022.

[18] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A Survey on Concept Drift Adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 44:1–44:37, 2014, doi: 10.1145/2523813.

[19] N. Moustafa, "TON_IoT Datasets: A New Generation Dataset of IoT and IIoT for Cybersecurity Research," IEEE Access, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862, [Online]. Available: https://research.unsw.edu.au/projects/toniot-datasets

[20] N. Moustafa and J. Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems," in Proc. Military Communications and Information Systems Conference (MilCIS), 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942, [Online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[21] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant, Applied Logistic Regression, 3rd ed. Hoboken, NJ, USA: Wiley, 2013, doi: 10.1002/9781118548387.

[22] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[23] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[24] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504–507, 2006, doi: 10.1126/science.1127647.

[25] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in Proc. EMNLP, 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.