SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V13I5P117

Volume 13 Issue 5, 210-225, May 2026

Original Article

An Adaptive Deep Learning Framework for DDoS Attack Detection under Concept Drift in IoT Networks

Jyotsna A Nanajkar[1], Sudhir B Lande[2], Sandhya A Shirsat[3], Anil S Shirsat[4], Vinay J Nagalkar[5]

[1]*Department of Electronics & Telecommunication, SVPM's College of Engineering, Malegaon, Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India.*
[2]*Department of Electronics & Telecommunication, VP's Kamalnayan Bajaj Institute of Engineering and Technology, Baramati, Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India.*
[3]*Department of Electronics & Telecommunication, Sinhgad College of Engineering, Pune, Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India.*
[4]*Department of Electronics & Telecommunication, PES's Modern College of Engineering, Pune, Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India.*
[5]*Department of Electronics & Telecommunication, Ajeenkya DY Patil School of Engineering, Pune, Savitribai Phule Pune University (SPPU), Pune, Maharashtra, India.*

[2]*Corresponding Author : jyotsna.nanajkar@gmail.com*

Received: 25 February 2026 Revised: 24 March 2026 Accepted: 23 April 2026 Published: 30 May 2026

Abstract *- DDoS attacks are a constant menace to Internet of Things (IoT) networks because they are growing in magnitude, changing their methods of attack, and can bypass fixed detection systems. This difficulty is also compounded by concept drift, in which the statistical characteristics of network traffic change with time, causing the performance of fixed or periodically retrained machine learning models to degrade quickly. This paper suggests a self-adaptive deep representation learning model to overcome these constraints to detect concept-drift-resilient DDoS attacks in IoT settings. The suggested solution trains small, behavior-sensitive latent network traffic representations and updates them in a prequential streaming protocol. It uses a drift-sensitive stability regularization mechanism to reduce catastrophic forgetting and allow continuous adaptation to changing traffic distributions. The framework is tested on chronologically ordered experimental protocols on recent IoT intrusion datasets, such as CIC IoT DIAD 2024 and IoT-DH, which capture long-term traffic evolution due to device behavior and adaptive attack dynamics. In both datasets, the proposed method has better detection performance than classical and deep learning baselines, with detection accuracies of 95.1% and 94.4%, respectively, and statistically significant improvements ($p < 0.01$) and a 10.5% increase in Matthews Correlation Coefficient over the best adaptive baseline. The model also shows that there is less temporal performance degradation, and the retraining cost is reduced by 82 % with incremental adaptation. These findings suggest that adaptive latent representation learning offers a strong and computationally efficient approach to maintaining long-term DDoS detection performance in non-stationary IoT network settings.*

Keywords *- Concept Drift, DDoS Detection, IoT, Network Security, Adaptive Intrusion Detection.*

1. Introduction

Things (IoT) networks have emerged as an essential element of contemporary digital infrastructures, linking heterogeneous devices in industrial automation, smart cities, healthcare, and critical services. The massive use of resource-limited devices and their constant connectivity to dynamic network conditions contribute to the attack surface of IoT ecosystems to a considerable degree. DDoS attacks are one of the most disruptive types of cyber threats because of their scalability, flexibility, and the possibility of using compromised IoT devices as botnet agents [1]. The latest developments in attack techniques have turned the DDoS attacks into complex multi-vector attacks as opposed to the previous volumetric flooding. These are low-rate covert traffic, protocol exploitation, and adaptive attack patterns that are very similar to legitimate IoT communication. This has rendered traditional signature-based and rule-based intrusion detection systems ineffective in detecting these emerging threats.

In order to resolve these drawbacks, the machine learning discriminative methods of detection have been proposed to automatically derive discriminative patterns by analyzing network traffic data. Initial research has shown that deep learning can be used to detect distributed attacks in an IoT setting with effectiveness [2]. Later, extensive research on

decision-making in relation to anomaly detection demonstrated the significance of good feature representation in network security systems [3]. More improvements used deep autoencoder models to detect the activities of the IoT botnets based on behavior modeling [4]. Although these enhancements have been made, the vast majority of the current methods are based on such datasets as UNSW-NB15 [5] and CIC-based benchmarks [6], in which training and testing data are assumed to have a similar distribution. In real IoT applications, this assumption is not an exact one as there are constant changes in device behavior, network features, and attack plans. This is called concept drift and poses serious challenges to intrusion detection systems. Early research in concept drift has demonstrated that the performance of a model can be reduced significantly when the data distributions change as time progresses [7]. Adaptive windowing methods have been suggested to deal with time-varying data streams, but they mainly deal with the statistical adaptation as opposed to deep representation learning [8].

Even the recent publications presented secure IoT architectures and a distributed learning paradigm [9], and federated learning methods to decentralized intrusion detection [10], these approaches do not directly consider temporal drift in feature representation. The results of survey studies of DDoS detection with deep learning also prove that the majority of the current models are offline trained and do not have the ability to be continuously adapted [11]. Also, zero-trust architecture focuses on constant monitoring of network activity but uses detection systems that are not necessarily drift-sensitive [12].

Although these progressions have been made, there are three basic constraints that exist in the present-day IoT intrusion detection systems. To begin with, most models are based on either static training or regular retraining, which cannot be effective in capturing continuous distributional variations in streaming traffic. Second, current adaptive strategies are mainly on updating classifier decision boundaries only, without stabilizing learned latent feature representations, resulting in catastrophic forgetting. Third, most studies use randomly shuffled data, which creates an unrealistic assessment not based on time dependencies and changing attack patterns.

The most important one is to find a way of designing a detection framework that can respond to non-stationary traffic distributions to maintain the structure of latent representations that is discriminative over time. To have long-term sustaining performance under concept drift, a system like this should be flexible and stable at the same time.

To overcome this difficulty, the present work suggests a self-adaptive deep representation learning framework to detect DDoS in the IoT networks. The given method, in contrast to the current methods, centralizes on decision-boundary adaptation and provides representation-level adaptation, in which latent features undergo refinement. It introduces a drift-sensitive stability regularization schedule, which helps to avoid catastrophic forgetting, and a prequential chronological evaluation protocol, which helps to evaluate the situation realistically in streaming conditions.

1.1. Motivation

Traditional intrusion detection systems are based on predefined decision boundaries or periodic retraining policies, neither of which is suitable in the dynamic IoT settings. With the changing traffic distributions, the simple update of classifier parameters does not suffice since the feature representations are obsolete. This encourages the necessity of adaptation at the feature representation level, where latent representations undergo constant changes without any temporal discrepancy.

This research paper aims to come up with a framework that provides stable, continuous, and computationally viable learning under concept drift, hence allowing real-time IoT deployment to proceed continuously.

1.2. Contributions

The key findings of this work include the following:

- Representation-Level Adaptation: This model is a continuous updating of latent feature representations to ensure that they remain discriminative in the changing traffic conditions.
- Drift-Aware Stability Regularization: It is a stability-constrained learning algorithm that alleviates catastrophic forgetting in sequential adaptation.
- Prequential Temporal Evaluation: This is a realistic evaluation protocol based on chronological streams of data, and it avoids any temporal leakage.
- Joint Stability-Adaptability Optimization: A combined goal that allows adapting to new patterns and maintaining previous knowledge.
- Deployment-Oriented Design: This is a computationally efficient design that uses less retraining overhead and high inference throughput, which is appropriate to resource-constrained IoT systems.

1.3. Positioning with Respect to Existing Work

The current DDoS detection methods can be divided into three broad categories, namely: static machine learning and deep learning, periodically retrained, and partially adaptive frameworks. The former are more successful in the case of fixed distributions, but are highly unsuccessful when concept drift is introduced to the system [13]. The classical machine learning approaches also have the disadvantage of their gradual deterioration with the performance as the feature extraction functions are hand-crafted [14]. Deep learning models enhance the extraction of features but are highly reliant on the offline training assumptions [15]. More accurate

architectures based on feature engineering also provide further improvement, but do not cover the temporal drift [16]. Online policy updates can be made through adaptive strategies, like reinforcement learning, which create instability and high computational costs within the IoT setting [17]. On the same note, sequence-based models have the capability of capturing time dependence, but they are usually latent and scale dependent [18]. Representation learning models, including autoencoders, offer better generalization but are vulnerable to changing normal behavior [19]. Generative models are more robust but unpredictable in terms of training and not deployable [20]. Gradient boosting and hybrid algorithms enhance the detection rate, but also make the relatively stable data distributions [21]. The systems based on federated learning encourage decentralized learning but lack a specific way of addressing the issue of representation drift when no IID conditions are met [22]. Recent secure IoT frameworks focus on the resiliency of the system level, but do not offer learning mechanisms that are drift-aware [23].

In contrast, the framework presented above presents a representation-based adaptive paradigm, in which the latent feature space is dynamically defined and changed through well-stated stability requirements. This allows long-term concept drift detection performance, with computational efficiency, unlike current methods, which are primarily parameter or decision-boundary adaptation methods.

The rest of this paper will be structured in the following manner. Section 2 introduces the literature review and gaps in the research. Section 3 is a mathematical statement of the problem. The proposed methodology is presented in Section 4. The experimental results are given in Section 5. Section 6 is on limitations, Section 7 is on Practical Implications, and the conclusion is in Section 8, and future work is mentioned in Section 9.

2. Literature Review

This section critically examines the current methods of DDoS detection in IoT networks, including the methodological development, constraints in concept drift, and the gaps in representation-level adaptation.

2.1. Background and Technical Foundations

IoT networks DDoS detection is on the border of network security and data-driven learning because a heterogeneous device base, limited resources, and the dynamism of traffic streams make intrusion detection a challenging problem. As opposed to conventional networks, traffic in IoT has high non-stationarity, where statistical properties change through the progression of time because of device behavior, system changes, adversarial adaptation, among others [7]. The dynamic nature of data over time (often known as concept drift) violates the stationary assumptions underlying most traditional machine learning models. The drift can be in the form of feature distributions, class priors, and decision boundaries, eventually ruining the detection reliability and leading to more false alarms in long-run applications. A majority of the available intrusion detection systems are based on offline or batch-based learning models, in which models are trained and deployed, but not adjusted. Although deep learning does a better job at feature representation, it continues to make many assumptions about fixed data distributions, or periodically retreating, which is computationally intensive and is not applicable in a streaming IoT setting.

Recent adaptive and incremental methods of learning strive to overcome this drawback by updating models with time. Nonetheless, one of the main issues was not sufficiently tackled, i.e., the need to retain stability of the learned representations in case of sequential updates. Adaptive models without specific limitations tend to forget in a catastrophic manner, and therefore, have come to forget what they have learned previously and perform irregularly.

Hence, the IoT should have an effective DDoS detection with a unified framework that ensures:

(i) It adapts continuously to streaming data,
(ii) the maintained discriminative latent representations, and
(iii) resilience to changing and hidden patterns of attacks.

Such difficulties drive the desire to have a representation-conscious adaptive learning framework, which is further examined in the literature review that follows.

2.2. Conventional Statistical and Signature-Based Methods

Initial DDoS detection systems were based on statistical traffic analysis and threshold-based monitoring [5]. The typical indicators are packet arrival rate, traffic entropy, source-destination dispersion, and protocol usage statistics. These techniques worked well in detecting high-rate volumetric flooding attacks that are typified by sudden changes in the normal traffic [11]. Nevertheless, modern IoT settings are characterized by very dynamic and heterogeneous traffic patterns. Low-rate DDoS attacks, protocol-conformant floods, and application-layer exploitation can be similar to normal IoT communication, which diminishes the discriminative ability of fixed statistical thresholds [12]. Furthermore, thresholding is parameter-tuning sensitive and does not adapt to changing traffic distributions, which results in more false positives and false negatives in the drift case [13]. Intrusion detection systems based on signature detection systems expanded statistical techniques with predefined attack patterns. Although useful in the case of known threats, signature-based systems need to be updated manually frequently and are not capable of detecting zero-day or polymorphic attacks. Traffic obfuscation, protocol mimicry, and distributed botnet behavior also make them less effective in heterogeneous IoT deployments. Therefore, purely statistical and signature-based methods are not adaptable and scalable in non-stationary IoT settings.

2.3. Detection Based on Machine Learning

Machine Learning (ML) methods presented data-driven detection features through learning discriminative patterns based on network traffic features. SVM, Random Forests (RF), k-Nearest Neighbors, and Gradient Boosting are examples of supervised classifiers that have been shown to perform better on benchmark datasets [14, 15]. Ensemble techniques also increased strength through the integration of several decision models [16]. Although more accurate, the traditional ML methods are highly dependent on manually designed features. The feature sets that are optimized to work in particular deployment situations do not tend to generalize to heterogeneous IoT devices, communication protocols, or traffic conditions. More to the point, the majority of classical ML models presuppose stationary data distributions and are trained in batch. These models are characterized by a high rate of degradation under distributional shifts, which is why they cannot be used in long-term IoT deployment when traffic is changing over time.

2.4. Deep Learning-Based Methods

Deep LEARNING (DL) approaches address the problem of manual feature engineering by learning hierarchical representations directly on traffic data. Convolutional Neural Networks (CNNs) are able to capture spatial interactions of features, whereas Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks can capture temporal interactions of sequential traffic flows [17, 18]. Anomaly detection models based on autoencoders are trained on benign traffic representations and detect anomalies as possible attacks [19]. It has also been investigated that Generative Adversarial Networks (GANs) can be used to model traffic distributions and enhance resistance to adversarial patterns [20].

Table 1 summarizes representative deep learning–based approaches for IoT DDoS detection, highlighting their core learning strategies, strengths, and limitations with respect to concept drift and deployment feasibility.

Table 1. IoT DDoS Detection Representative Deep Learning Models

Model Category	Core Idea	Strength	Limitation Under Drift
CNN-based IDS [17]	Spatial feature learning	High classification accuracy	No online adaptation
LSTM-based IDS [18]	Temporal modelling	Sequence awareness	High computational latency
Autoencoder IDS [19]	Unsupervised anomaly detection	Zero-day detection	Sensitive to evolving normal behavior
GAN-based IDS [20]	Traffic distribution modeling	Improved robustness	Training instability
Proposed Framework	Adaptive latent representation learning	Representation-level adaptation	Increased design complexity

Although DL models are better than classical ML in static benchmarks, the majority of them are trained offline and deployed without ongoing adaptation. The most common approach to performance degradation is periodic retraining, but this method adds computational overhead, operational disruption, and reliance on labeled data. Importantly, the current DL approaches are mainly focused on adapting decision boundaries instead of directly preserving stability in the latent representation space in the presence of drift.

2.5. Adaptive and Reinforcement Learning Methods

The adaptive learning methods are aimed at addressing non-stationarity so that the model can be modified over time. The reinforcement learning (RL) has been applied to intrusion detection in which the agents can alter detection policies according to the environmental responses [21]. Though theoretically, RL-based systems are flexible, they cannot be applied to the IoT due to pragmatic problems:

- Complexity of reward functions design.
- Exploration-induced instability
- Slow computational speed.
- Slow convergence

These features make them less suitable for latency-sensitive and resource-constrained IoT settings.

2.6. Federated, Fog, and Distributed Detection Architectures

Scalability and privacy issues have been proposed to be addressed by Federated learning (FL) and fog-based detection frameworks [22, 23]. Federated learning enables decentralized model training without sharing raw data, which enhances privacy protection.

However, the federated IDS systems are faced with issues like:

- Communication overhead
- Synchronization complexity
- Distribution of non-IID clients.
- Heterogeneous node model drift.

Equally, getting latency to the bare minimum is achieved with fog and edge-based architectures, which relocate the detection to the data sources, but they often rely on static or periodically updated models, which limit resilience to continuous concept drift.

2.7. Zero-Trust and Next-Generation Security Paradigms

Unlike perimeter-based defense, which is based on the notion of constant verification of the behavior of devices and access control, zero-trust architectures emphasize these aspects [1]. Though the principles of zero-trust can make the network safer, they are based on intelligent traffic analysis systems that will be able to adapt to the alteration of the

behavioral patterns. The current detection systems do not have the appropriate continuous representation adaptation within such architectures.

2.8. Comparative Analysis and Research Gaps

The literature review indicates that there have been some limitations that have remained:

- Dominating use of fixed or periodically retrained models.
- Weak concept drift, lifelong or streaming adaptation support.
- Very little emphasis on representation-level stability mechanisms.
- The extensive application of unrealistic random data disregards time development.
- Inadequate reporting of deployment-related metrics (latency, overhead, memory).

Recent IoT datasets, such as IDSIoT2024 [24], provide realistic traffic traces, but are not generally tested using systematic drift-aware benchmarking; instead, they are generally tested using robustness evaluation Although some of the previous studies have studied deep learning-based detection, adaptive learning, and distributed frameworks separately, none of them have combined (i) continual concept drift, (ii) latent representation stability, and (iii) realistic time analysis. The suggested framework is the first of its kind to encompass these three dimensions into a coherent learning paradigm, thus filling a major gap between what the theory knows and the reality of the application needs in terms of IoT deployment.

Table 2 highlights major distinctions between current methods.

Table 2. IoT DDoS Detection Approaches Comparison

Method Type	Learning Strategy	Concept Drift Support	Adaptation Level	Evaluation Realism	Key Limitation
Signature-based IDS [13]	Static rules	No	None	Low	Cannot detect unseen attacks
Traditional ML [14, 15]	Batch supervised	No	Decision boundary	Low	Rapid degradation under drift
Deep Learning [17, 18]	Offline training	Limited	Decision boundary	Moderate	High retraining cost
Autoencoder IDS [19]	Offline anomaly learning	Limited	Latent features	Moderate	Drift sensitivity
Reinforcement Learning [21]	Online policy learning	Partial	Policy-level	Moderate	Slow convergence
Federated IDS [22]	Distributed parameter updates	Partial	Model-level	Moderate	Non-IID data issues
Proposed Framework	Lifelong incremental learning	Yes	Representation-level	High (temporal prequential)	Increased architectural complexity

2.9. Research Hypothesis and Research Questions

2.9.1. Hypothesis

Self-adaptive deep representation learning is able to sustain a high level of DDoS detection performance in the face of concept drift by continually updating behavior-sensitive latent representations whilst limiting temporal instability.

2.9.2. Research Questions

1. RQ1: How well can representation-level adaptation maintain detection accuracy in changing IoT traffic distributions?
2. RQ2: To what extent does the framework generalize to heterogeneous IoT data and attack conditions?
3. RQ3: How does the trade-off between adaptation overhead and long-term detection stability in streaming IoT deployments look like?

These questions can only be answered through formal problem modeling and drift-aware optimization, which is discussed in the next section.

3. Problem Formulation and Mathematical Modeling

3.1. Network Traffic Representation

Think of the IoT network traffic as a time-ordered sequence of flow records produced by monitoring systems. A feature vector is used to represent each traffic flow at time index t:

$$x_t \in R^d \quad (1)$$

Where d is the dimensionality of the feature space and $x_t = [x_t^1, x_t^2, \dots, x_t^d]$ has statistical, temporal, and protocol-level information, including the number of packets, the rate of bytes, inter-arrival times, header flags, and flow duration.

Every case is linked to a binary label:

$$y_t \in \{0,1\} \quad (2)$$

Where:

$y_t = 0$ represents malignant traffic

$y_t = 1$ represents DDoS attack traffic.

The streaming data is characterized by:

$$D = \{(x_t, y_t)\}_{t=1}^{T} \quad (3)$$

This model is a reflection of real-world implementation where traffic is received in a sequence, and decisions have to be made online without the ability to observe in the future.

3.2. Deep Representation Learning

An encoder network to model nonlinear relationships in traffic behavior $f_\theta: R^d \to R^k$ is a parameterized map of the input feature vector to a latent representation:

$$z_t = f_\theta(x_t), z_t \in R^k, k \ll d \quad (4)$$

The latent space aims to:

- Maintain a discriminative pattern between benign and attack traffic.
- Eliminate redundant correlations and noise.
- Enhance generalization to changing or novel attack patterns.

A classifier

$$g_\phi: R^k \to [0,1]$$

Where ϕ is a parameter, gives the probability of detection: $\hat{y}_t = g_\phi(z_t)$, and $\hat{y}_t$ is the forecasted probability of a DDoS attack.

The composite detection model is:

$$h_\Theta(x_t) = g_\phi\big(f_\theta(x_t)\big), \Theta = (\theta, \phi) \quad (5)$$

3.3. Concept Drift Modeling

The joint distribution of features and labels changes with time in dynamic IoT environments. The concept drift is formally defined as:

$$P_t(x, y) \neq P_{t+\Delta}(x, y) \quad (6)$$

Drift may arise from:

- Alterations in benign device behavior.
- Development of attack tactics.
- Reconfiguration of the network or infrastructure.

In the case of streaming learning, the dataset is divided into windows that are ordered in time:

$$D = \bigcup_{i=1}^{N} D^{(i)} \quad (7)$$

With every window $D^{(i)}$ having a possibly different distribution $P^{(i)}(x, y)$.

In the case of the static models, it is assumed that there is stationarity. $P^{(1)}(x, y) = P^{(2)}(x, y) = \cdots = P^{(N)}(x, y)$ which is barely accurate in IoT deployments.

3.4. Self-Adaptive Learning Under Drift

Let the model parameters at window i be $\Theta^{(i)} = \big(\theta^{(i)}, \phi^{(i)}\big)$. Therefore, the parameters are updated in a step-by-step manner as new data comes in:

$$\Theta^{(i+1)} = \Theta^{(i)} - \eta \nabla_\Theta L^{(i)} \quad (8)$$

Where, $L^{(i)}$ is the window-specific loss and η is the learning rate.

Loss Function: The objective function is a combination of classification accuracy and latent stability:

$$L^{(i)} = L_{cls}^{(i)} + \lambda L_{reg}^{(i)} \quad (9)$$

Where:
$L_{cls}^{(i)}$ is binary cross-entropy loss
$L_{reg}^{(i)}$ enforces representation stability
$\lambda > 0$ controls the adaptation–stability trade-off

The stability term prevents catastrophic forgetting by punishing sudden latent shifts:

$$L_{reg}^{(i)} = E\big[\| f_{\theta^{(i)}}(x_t) - f_{\theta^{(i-1)}}(x_t) \|_2^2\big] \quad (10)$$

This formulation maintains the discriminative structure and allows gradual adaptation.

3.5. Zero-Day and Generalization Perspective

In real-world implementation, new variants of attacks might arise that were not used in training.

Let $Y_{train} \subset Y_{test}$, where Y_{train} refers to the types of attacks that are seen during training and Y_{test} refers to unknown variants.

Thus, the detection task should not be limited to particular attack signatures, but should be based on behavioral deviation coded in the latent representation.

The self-adaptive representation model seeks to ensure that the separability of benign and malicious behavior can be maintained as $P(x, y)$ changes.

3.6. Dataset-Specific Modeling

3.6.1. CIC IoT DIAD 2024

- Feature dimension: $d = 78$,
- Latent dimension: $k = 24$
- Chronological split:

- 60% initial training
- 20% adaptation
- 20% sequential testing

Zero-day setup does not include the training phase of selected attack categories, and instead, they are introduced during testing to test generalization.

3.6.2. IoT-DH Dataset

- Feature dimension: $d = 70$
- Latent dimension: $k = 20$
- Strict chronological division.

Temporal segmentation removes leakage and models real-world streaming deployment.

The formal optimization problem is as follows:
Assuming a time-ordered stream of data.

$$D = \{(x_t, y_t)\}_{t=1}^{T} \quad (11)$$

Where $P_t(x, y)$ is a non-stationary distribution, the aim is to learn parameters Θ so that:

$$\min_{\Theta^{(1)},\ldots,\Theta^{(N)}} \sum_{i=1}^{N} L^{(i)} \quad (12)$$

subject to:

- Sequential updates with past and current windows only.
- Stability conditions to avoid disastrous forgetting.
- Streaming IoT efficiency.

The aim is to reduce cumulative detection error and limit temporal degradation with changing traffic distributions.

3.7. Problem Statement

Given a temporally evolving network traffic stream (D) exhibiting concept drift and emerging DDoS attack patterns, this work aims to design a self-adaptive deep representation learning framework that dynamically refines its latent feature representations and classification parameters to ensure accurate DDoS detection, robustness against distributional variations, and minimal degradation in performance over time.

4. Proposed Methodology

4.1. Overview of the Self-Adaptive Framework

In this section, the authors introduce a drift-sensitive self-adaptive deep representation learning model to detect DDoS in streaming IoT systems. The framework is intended to be used in non-stationary traffic conditions where the behavior of the devices, communication patterns, and attack strategies change with time. It is this design that sets the proposed framework apart, superimposed on existing IDS models since it explicitly couples representation learning, drift adaptation, and stability preservation in a unique optimization structure. In contrast to the case of the static models or periodic retraining strategies, the proposed methodology conducts the representation-level incremental adaptation, maintaining the previously learned discriminative structure and updating the latent space in the case of distributional shifts.

The general pipeline is composed of:

- Preprocessing of traffic and time sequence.
- Latent representation learning through an encoder network.
- Incremental adaptation that is constrained by stability.
- DDoS inference based on behavior.

Figure 1 shows the overall architecture of the proposed framework and how raw IoT traffic flows are converted into adaptive latent representations to be used in robust DDoS detection.

4.2. Traffic Preprocessing and Temporal Structuring

4.2.1. Data Cleaning and Feature Normalization

IoT flow-level traffic data are usually incomplete, noisy, and monitoring artifacts. In order to make learning stable:

- Flow corruption and incompleteness are eliminated.
- Categorical fields that are not informative are eliminated.
- Redundant records are removed.

Continuous variables are normalized to a z-score:

$$x' = \frac{x-\mu}{\sigma} \quad (13)$$

Where μ and σ are only calculated using training data to avoid leakage.

One-hot encoding is used to encode categorical features where necessary (e.g., protocol types in CIC IoT DIAD 2024). The IoT-DH characteristics are already numerically aggregated at the flow level. Normalization provides numerical stability and avoids high-magnitude attributes dominating the optimization process.

4.2.2. Temporal Windowing

The traffic records are ordered by time in the strict sense prior to processing to mimic the realistic streaming deployment. The ordered stream is split into windows, which are discontinuous:

$$D = \{D^{(1)}, D^{(2)}, \ldots, D^{(N)}\} \quad (14)$$

Flow counts in each window are constant (75,000 in experiments), and can be updated in an incremental fashion with chronological integrity. This not only removes the time leakage but also does not involve the unrealistic random shuffling of data that was being used in earlier works.

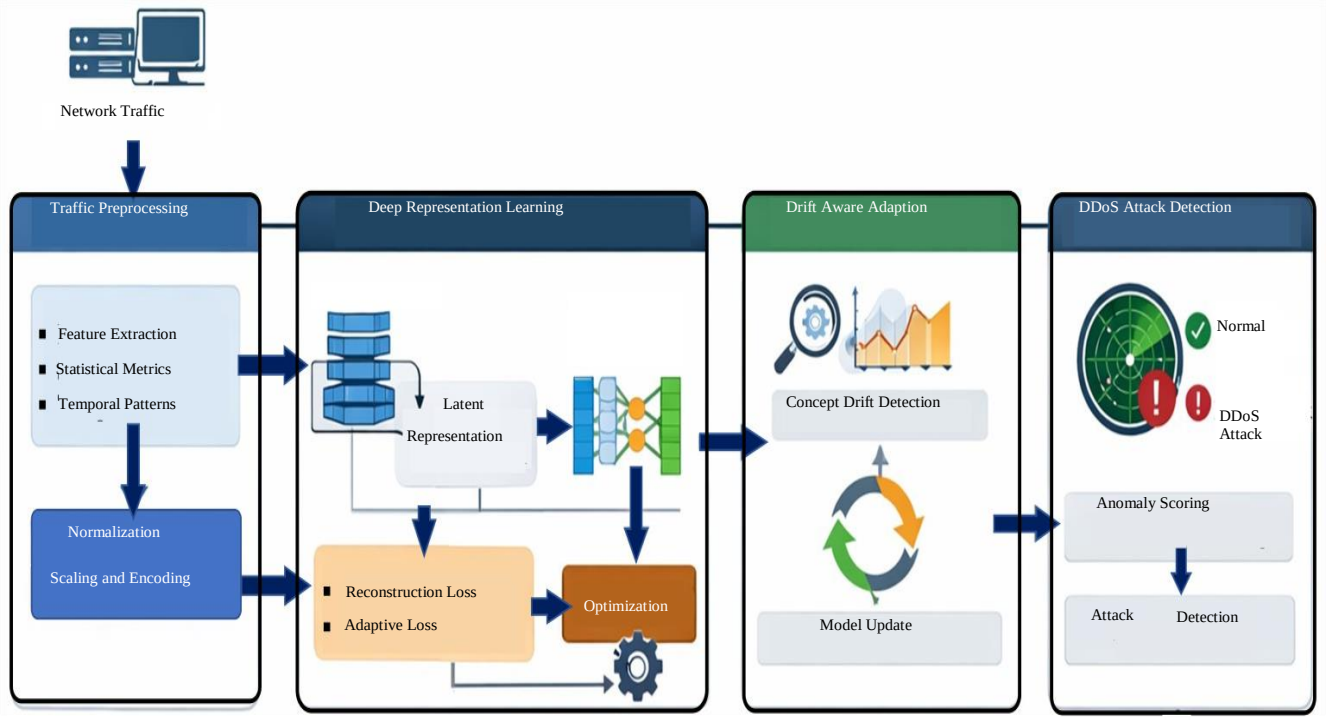

Fig. 1 General structure of the suggested self-adaptive deep representation learning system

4.3. Deep Representation Learning Module

4.3.1. Encoder Architecture

The encoder network predicts high-dimensional traffic properties into a low-dimensional latent space:

$$z = f_\theta(x) \tag{15}$$

The architectural design principles include:

- Fully connected feed-forward layers.
- ReLU activation functions
- Regularization dropout (rate = 0.3).
- Bottleneck layer of dimension k.

Dataset-specific configurations:

- CIC IoT DIAD 2024: $d = 78$, $k = 24$
- IoT-DH: $d = 70$, $k = 20$

The bottleneck imposes abstraction, which promotes the model to capture behavior-level properties instead of dataset-specific artifacts.

4.3.2. Latent Stability Regularization

Sequential adaptation may result in disastrous forgetting. In order to reduce this, a latent consistency constraint is added:

$$L_{reg} = E\left[\| f_{\theta^{(i)}}(x_t) - f_{\theta^{(i-1)}}(x_t) \|_2^2\right] \tag{16}$$

This penalty discourages abrupt latent alterations between consecutive windows and allows smooth adaptation. Decision geometry in embedding space is preserved, and long-term learning is stabilized by the mechanism

4.4. Drift-Aware Incremental Learning

4.4.1. Initial Training Phase

Training under baseline manages the initial 60% of chronologically ordered traffic:

$$L_{init} = L_{cls} \tag{17}$$

This creates pre-emptive discriminative latent images of benign and familiar attack behavior. In order to update the parameters of the next window $D^{(i)}$ is used:

$$L^{(i)} = L_{cls}^{(i)} + \lambda L_{reg}^{(i)} \tag{18}$$

Where $\lambda = 0.1$ is a tradeoff between adaptation and stability.

Optimization details:

- Adam optimizer
- Initial learning rate: 0.001
- Adaptation learning rate: 0.0001
- 15 epochs per window
- Early termination using validation loss.

This plan will guarantee a controlled adaptation without complete retraining. Figure 2 shows how traffic is temporally segmented and how incremental updates can be used to allow continuous learning in the presence of concept drift.

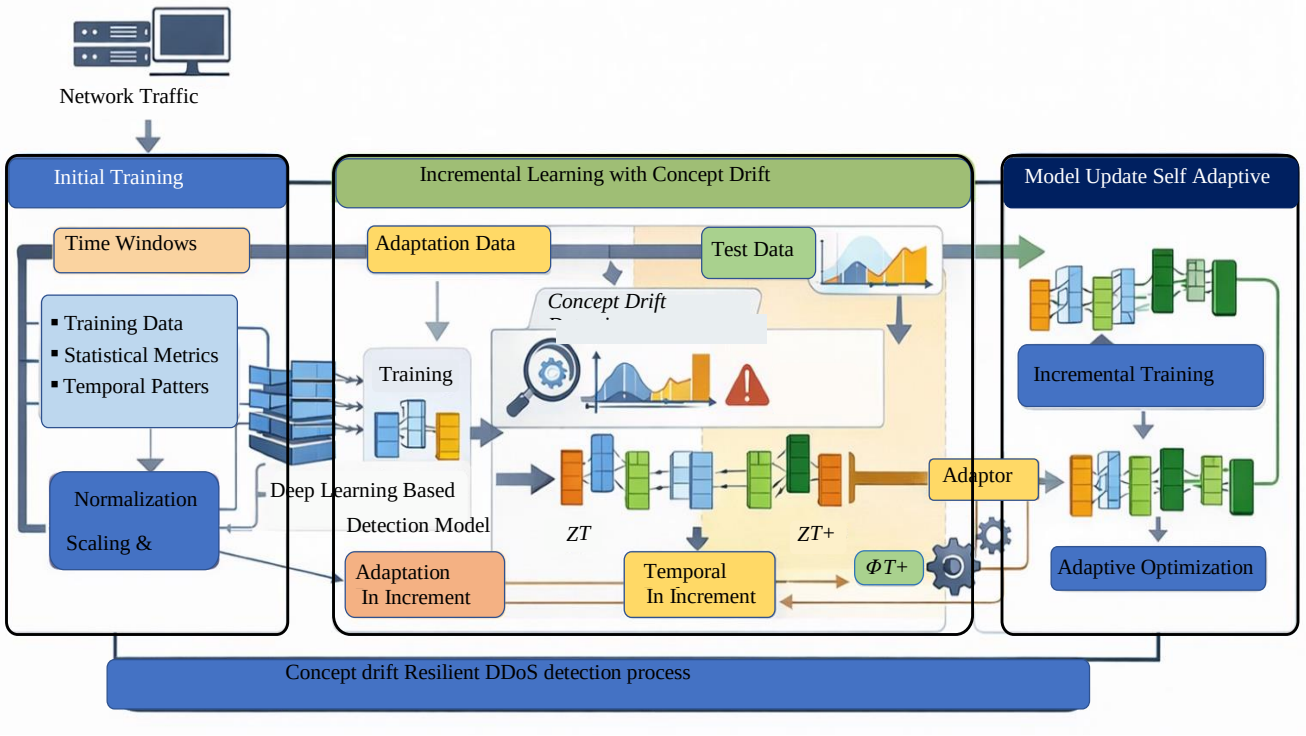

Fig. 2 Temporal learning and self-adaptive model update process

This adaptive update mechanism enables the system to acquire changing attack features without having to retrain afresh.

4.4.2. DDoS Inference Mechanism

During inference:

$$\hat{y}_t = g_\phi\left(f_\theta(x_t)\right) \tag{19}$$

Validation data is used to choose a decision threshold τ to keep the false alarm rate under control. In contrast to signature-based systems, detection is based on behavioral deviation coded in the latent representation, which allows it to be resistant to changing attack strategies.

4.5. Experimental Protocol and Dataset Configuration

Evaluation is performed on:

- CIC IoT DIAD 2024
- IoT-DH

Both datasets are chronologically split:

- 60% initial training
- 20% adaptation
- 20% testing

Windows contain 75,000 flows.

This protocol is a realistic streaming deployment and not a random split evaluation. Table 3 gives a summary of the datasets used for experimental evaluation.

Table 3. Dataset Characteristics

Dataset Name	Data Source Type	Traffic Nature	Attack Characteristics	Duration / Scope	Relevance to Study
CIC IoT DIAD 2024 [5]	Controlled testbed (flow-based)	Synthetic IoT traffic	Multi-class DDoS, scanning, brute-force, botnet, zero-day-like behaviors	Large-scale, long-term	Enables controlled evaluation of concept drift and zero-day detection
IoT-DH [25]	Real-world honeypot	Naturally occurring IoT traffic	Real attack evolution, unknown and emerging threats	Continuous, real deployment	Validates robustness under realistic, non-stationary attack conditions
IDSIoT2024 [26]	Physical IoT deployment	Long-term device traffic	Diverse real attacks	Generalization test	

4.5.1. Cross-Dataset Generalization

IDSIoT2024 is applied to the qualitative robustness evaluation in the naturally changing traffic conditions. Even though it is not applied as a primary benchmarking tool, it offers further evidence of stability in heterogeneous IoT deployment conditions.

4.5.2. Implementation and Reproducibility

The model is written in Python (3.10) and PyTorch (2.x).

Reproducibility measures:

- Deterministic random seeds (NumPy, PyTorch, OS-level)
- Strict chronological data partitioning.
- Training-statistics-only normalization
- Statistical reporting of five independent runs.

Figure 3 shows the entire pipeline of the experiment, with a strong focus on the chronological processing of traffic data.

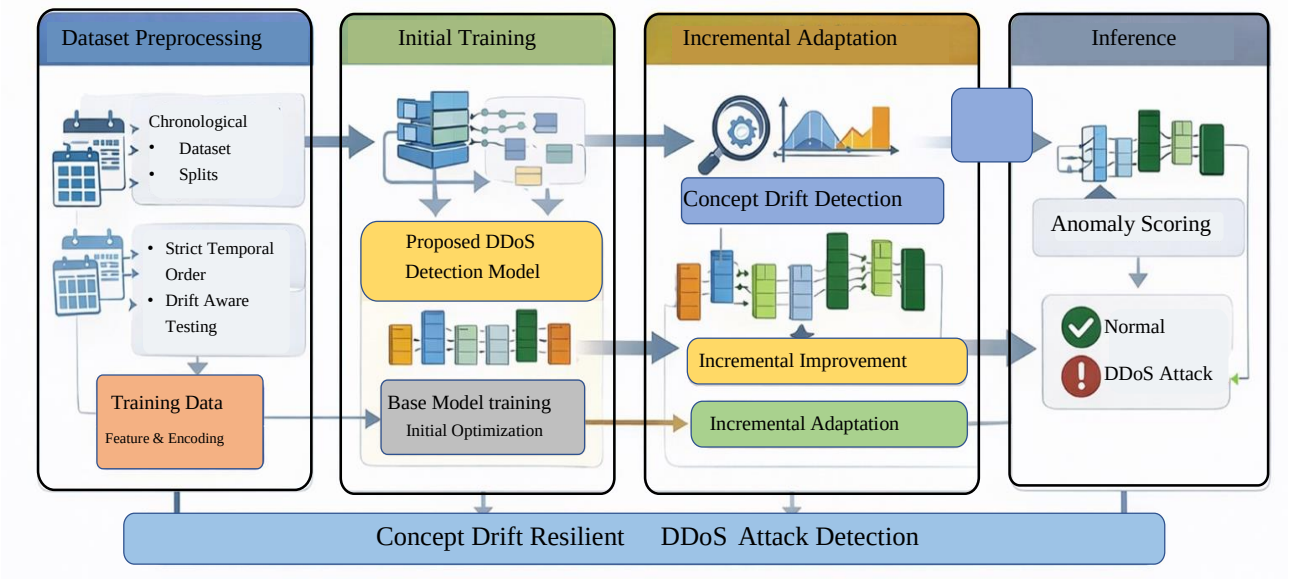

Fig. 3 Dataset preprocessing, training, adaptation, and inference workflow

4.6. Methodological Summary

The suggested methodology contrasts with the previous research in three important ways:

1. Representation-level adaptation instead of decision-boundary retraining.
2. Latent stability constraint of drift mitigation.
3. Prequential chronological analysis to model actual deployment.

This architecture allows computationally efficient, stable, and scalable DDoS detection in non-stationary IoT traffic.

5. Experimental Setup and Results

5.1. Implementation and Experimental Environment

Experiments were performed in a prequential streaming evaluation protocol, in which each temporal window $D^{(i)}$ was initially assessed with the model parameters $\Theta^{(i-1)}$ and subsequently used for incremental adaptation. Formally, prediction precedes update:

$$\hat{y}_t = h_{\Theta^{(i-1)}}(x_t), x_t \in D^{(i)} \quad (20)$$

Then update the parameters:

$$\Theta^{(i)} \leftarrow \Theta^{(i-1)} - \eta \nabla L^{(i)} \quad (21)$$

This protocol is a realistic simulation of a streaming IoT deployment, in which future observations are unavailable during inference and adaptation can only be based on past and current traffic.The suggested self-adaptive deep representation learning model was developed in Python 3.10 and PyTorch 2.x. The experiments were performed on a workstation with an Intel Xeon processor at 3.6 GHz, 64GB of RAM, and an NVIDIA RTX 3090 graphics card with 24GB of VRAM.

Training and adaptation phases were performed with the use of GPU acceleration, and the inference latency was measured on the CPU to replicate the conditions of the edge and gateway-level IoT deployment. Random seeds were fixed in NumPy, PyTorch, and system-level operations to guarantee reproducibility. The temporal sequence of traffic data was maintained strictly during preprocessing, training, adaptation, and testing. There was no random shuffling at any point, so the temporal leakage was removed and the long-term deployment scenarios were faithfully simulated.

The findings are presented in the form of mean $\pm$ standard deviation. Also, 95 % Confidence Intervals (CI) were calculated as:

$$CI_{95\%} = \mu \pm 1.96 \frac{\sigma}{\sqrt{n}}, n = 5 \quad (22)$$

Table 4 lists the hyperparameter settings and training configurations adopted for the proposed framework on each dataset to ensure reproducibility and fair evaluation.

Table 4. Model Hyperparameters

Parameter	IoT-DH	CIC IoT DIAD 2024
Latent dimension k	16	24
Initial learning rate	0.001	0.001
Adaptive learning rate	0.0001	0.0001
Batch size	256	512
Regularization weight λ	0.1	0.1
Epochs per window	15	15

The hyperparameters were selected by showing stability during the validation and sensitivity analysis. Training is done in two phases, whereby a first supervised step is used in which the first 60% of the chronologically ordered traffic is used, and then window-wise progressive adaptation in the remaining 40%. This final model contains 0.83 million trainable parameters, and this corresponds to approximately 32 MB memory footprint (32-bit precision). This parameter scale facilitates the implementation of IoT nodes at gateways. Figure 4 introduces the experimental design as a practice used to apply, observe, and test the proposed framework in concept-drifting IoT traffic conditions.

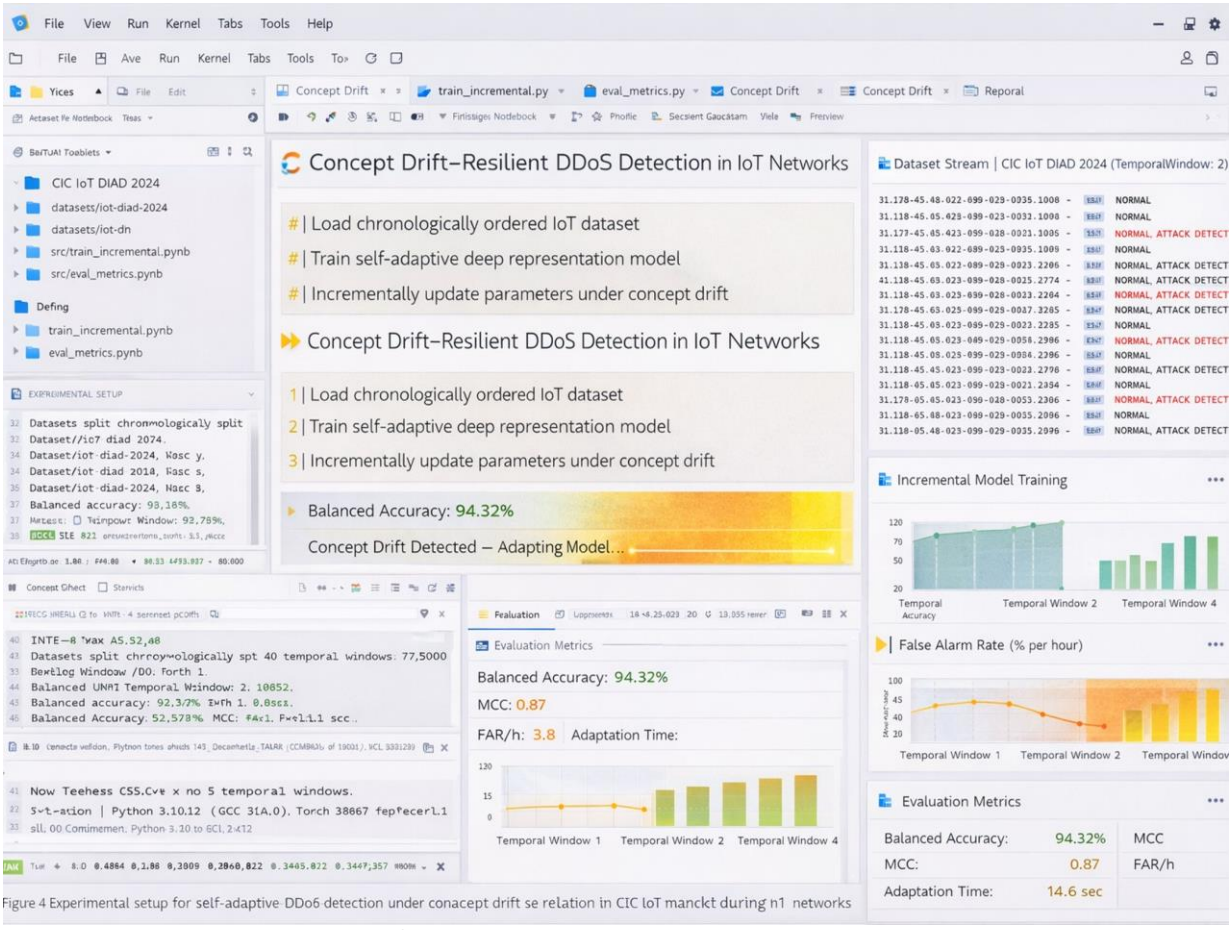

Fig. 4 Self-adaptive DDoS detection in IoT networks experimental setup

The experimental environment is an embodiment of an integrated programming and monitoring dashboard that logs the whole lifecycle of data ingestion, model training, incremental adaptation, and inference. The structure enables the latent traffic representations to be dynamically updated, the detection performance to be dynamically measured, and the operational measures of the detection accuracy, false alarm rate, and adaptation overhead to be tracked across consecutive time windows. This framework is concerned with realistic performance assessment, ongoing learning behavior, and realistic deployability in the evolving IoT traffic.

5.2. Baseline Models

Representative static and adaptive baselines were applied to ensure fair comparison with the same preprocessing and chronological splits:

- Support Vector Machine (RBF kernel; grid-searched C, γ)
- Random Forest (200 trees)
- Static DNN (128-64-32 fully connected layers, ReLU, dropout 0.3)
- Autoencoder + classifier (symmetric encoder-decoder)
- Periodic Retraining DL (retrained after every window on cumulative data)

Each baseline was trained on the same number of epochs per window and optimized with validation segments only to avoid contamination of the test. All deep learning baselines were optimized using Adam with comparable learning rate tuning and early stopping to ensure training parity. Parameter counts of deep learning baselines were maintained within ±10% of the proposed model to avoid capacity bias.

5.3. Evaluation Metrics

The evaluation of performance was based on:

- Accuracy
- Precision
- Recall
- F1-score
- Area Under ROC Curve (AUC)
- Matthews Correlation Coefficient (MCC)

MCC is defined as:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (23)$$

MCC was selected due to its robustness under class imbalance.

Deployment-based measures are:

- Balanced Accuracy
- Detection latency (ms)
- False alarm rate (per 10^5 flows)
- The rate of performance degradation.
- Adaptation overhead

Performance degradation is characterized as:

$$\Delta_{deg} = \frac{1}{N}\sum_{i=1}^{N}(Acc_1 - Acc_i) \quad (24)$$

The overhead of adaptation is calculated as:

$$O_{adapt} = \frac{T_{adapt}}{T_{retrain}} \quad (25)$$

5.4. Results on CIC IoT DIAD 2024

5.4.1. Performance of Detection in the Presence of Concept Drift

Table 5 presents the comparative detection performance of all evaluated models on the CIC IoT DIAD 2024 dataset under chronological prequential evaluation, reporting mean ± standard deviation across five independent runs.

Table 5. CIC IoT DIAD 2024 Results (Mean ± Std)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC	MCC
SVM	82.6 ± 0.8	79.4 ± 0.9	77.1 ± 1.0	78.2 ± 0.9	0.846	0.61
RF	86.1 ± 0.7	83.7 ± 0.8	81.5 ± 0.9	82.6 ± 0.8	0.883	0.66
Static DNN	89.4 ± 0.6	87.2 ± 0.7	85.6 ± 0.8	86.4 ± 0.7	0.912	0.71
AE+Classifier	90.2 ± 0.6	88.6 ± 0.7	86.9 ± 0.8	87.7 ± 0.7	0.919	0.73
Periodic DL	91.6 ± 0.5	89.8 ± 0.6	88.1 ± 0.7	88.9 ± 0.6	0.931	0.76
Proposed	95.1 ± 0.4	94.3 ± 0.5	93.7 ± 0.6	94.0 ± 0.5	0.967	0.84

Relative MCC improvement over periodic retraining:

$$\frac{0.84-0.76}{0.76} = 10.5\%$$

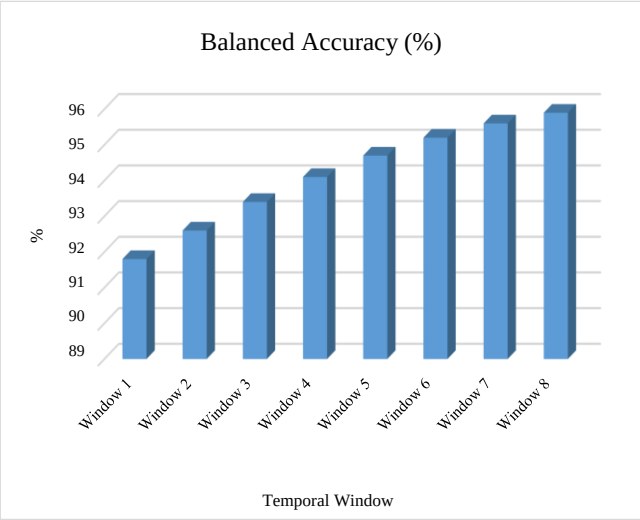

Fig. 5 Balanced accuracy variation across temporal windows on CIC IoT DIAD 2024 Dataset

Statistical testing was conducted across five independent runs, and statistical significance ($p < 0.01$) is confirmed using a paired t-test.

The effect size (Cohen's d) between the proposed model and the periodic retraining baseline was 1.27, indicating a large practical effect. Balanced accuracy variance across temporal windows remained below 0.8%, indicating stability under gradual drift. Normality assumption for the paired t-test was verified using the Shapiro-Wilk test.

In order to examine the stability of time, balanced accuracy trends between successive windows are presented in Figure 5. The findings indicate that there is low variance across temporal windows over time, which indicates the robustness of the framework to concept drift.

5.5. Results on IoT-DH

Table 6 shows the detection accuracy, robustness measures, and latency of the proposed and baseline models over the IoT-DH data set, which represents the performance of the models in the honeypot traffic of a real-world environment.

Table 6. IoT-DH Results (Mean ± Std)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	AUC	Latency (ms)
SVM	80.9 ± 0.9	77.8 ± 1.1	75.6 ± 1.2	76.7 ± 1.0	0.839	1.9
RF	84.7 ± 0.8	81.9 ± 0.9	79.6 ± 1.0	80.7 ± 0.9	0.874	2.3
Static DNN	88.5 ± 0.7	86.1 ± 0.8	84.2 ± 0.9	85.1 ± 0.8	0.907	1.6
AE+Classifier	88.2 ± 0.5	85.6 ± 0.4	84.5 ± 0.5	85.5 ± 0.8	0.899	1.8
Periodic DL	90.1 ± 0.6	88.3 ± 0.7	86.7 ± 0.8	87.5 ± 0.7	0.921	2.0
Proposed	94.4 ± 0.5	93.6 ± 0.6	92.8 ± 0.7	93.2 ± 0.6	0.963	1.4

The throughput of inference was 71,000 flows/sec on CPU, which is appropriate for use in real-time IoT implementation. The proposed framework achieves the highest detection accuracy and F1-score among evaluated models. These findings indicate that they are suitable for low-latency deployment scenarios.

Latency measurements were averaged over 100 inference batches. Figure 6 presents the model adaptation overhead that is experienced over time windows. The findings indicate that the computational cost of incremental updates is low compared to full retraining, which validates the scalability of high-volume IoT traffic.

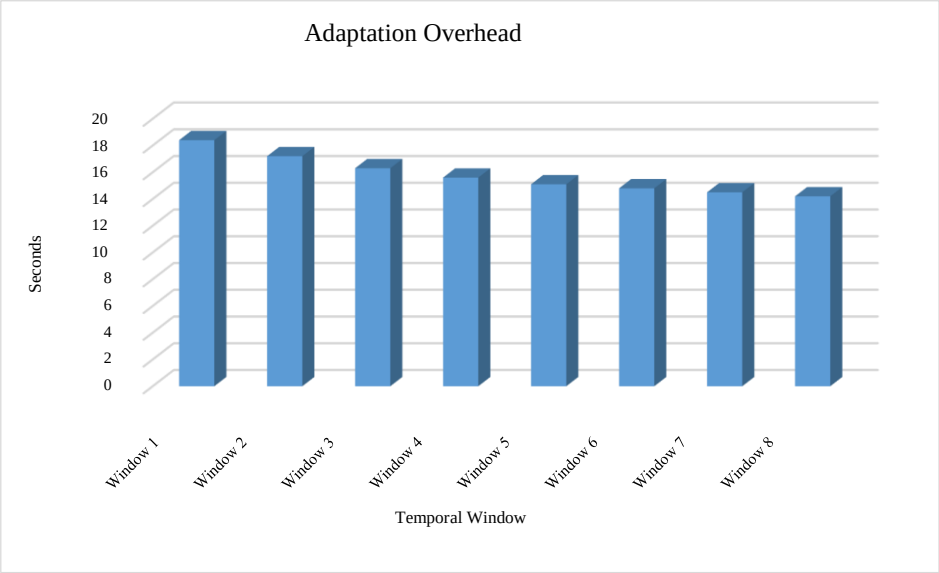

Fig. 6 Adaptation overhead of models on temporal windows of IoT traffic

5.6. Adaptation Overhead and Sensitivity Analysis

Incremental adaptation took about 18 % of the time needed to complete retraining:

$$O_{adapt} = 0.18$$

Table 7 reports the impact of the latent stability regularization coefficient on MCC and temporal degradation rate, illustrating the trade-off between adaptability and representation stability under concept drift.

Table 7. Sensitivity analysis for λ

λ	MCC	Degradation Rate
0.01	0.80	3.4%
0.05	0.83	2.1%
0.1	0.84	1.5%
0.2	0.82	2.7%

Window size variation between 50k and 100k flows resulted in MCC variation within ±0.7%, indicating robustness to segmentation granularity.

5.7. Ablation Study

Latent stability regularization removal decreased MCC by 0.84 to 0.79 (-6.1%) and increased the false alarm rate by 38 %, which is consistent with its contribution to catastrophic forgetting alleviation. Incremental adaptation was disabled, which doubled the rate of degradation (1.5% → 3.2%), highlighting the necessity of continuous adaptation under drift.

5.8. Analysis of Computational Complexity

Let d and k denote the input dimension and latent dimension, respectively.

- Forward pass complexity is given by: $O(dk + k^2)$
- Per window incremental update is given by: $O(\mid D^{(i)} \mid \cdot k)$
- Full retraining complexity is given by: $O(T \cdot d \cdot k)$

Thus, incremental adaptation reduces the cost of computation according to dependence upon the overall dataset size T, to the window size $|D^{(i)}|$, and leads to empirical efficiency improvements. Complexity analysis presupposes a constant number of epochs per window. In both datasets, the proposed framework can be statistically seen to exhibit statistically significant performance gains ($p < 0.01$, large effect size), as well as a 10.5% gain in MCC compared to periodic retraining. Temporal degradation under drift is substantially reduced, while incremental adaptation lowers retraining cost by approximately 82%. The compact model footprint (3.2 MB) and high inference throughput enable real-time deployment, confirming that adaptive latent representation learning improves robustness under non-stationary IoT traffic while maintaining computational efficiency.

6. Validity Threats and Limitations

Although its empirical performance in various datasets and streaming evaluation protocols is very high, there are a number of limitations that one should consider. The analysis is based mainly on CIC IoT DIAD 2024 and IoT-DH, which, despite being organized in time and acting in a realistic manner, fail to reflect the heterogeneity of functioning IoT ecosystems. Many different feature distributions may occur with encrypted traffic environments, industrial control protocols (e.g., Modbus, OPC-UA), ultra-low-power communications (e.g., LoRaWAN), and highly application-specific device behaviors. In addition, although further evaluation of IDSIoT2024 enhances robustness analysis to some extent, not all of the datasets are Internet-scale, carrier-grade, or multi-domain DDoS campaigns across the globe. Therefore, wider external validation is still required to generalize findings to large backbone networks.

The chronological segmentation and window-based incremental adaptation modeling of concept drift are mainly used to model gradual or moderate distributional changes. Sudden flash crowds, or coordinated transitions of multi-vector attacks, massive firmware releases, or adversarial distribution manipulation can cause more severe degradation before performance becomes stabilized by adaptation. Moreover, the framework presupposes the availability of labeled or weakly labeled data at adaptation, potentially restricting its use in entirely autonomous settings unless it is supported by unsupervised or self-supervised systems. Incremental updating can greatly decrease retraining overhead, but even ultra-constrained edge devices may be non-negligible in terms of computational costs, and may need model compression or hardware-aware optimization to be deployed to microcontrollers.

7. Discussion and Practical Implications

The proposed framework is ranked as always better than both the models in the case of concept drift (when the model is not updated regularly) and periodically retrained models, mainly because of three major aspects of its design. To start with, the representation-level adaptation will allow the individual optimization of the latent feature space, preserving the class separation as the traffic distributions will change. This method is better than the conventional methods, which change only decision boundaries and minimize temporal performance degradation in non-stationary conditions.

Second, the catastrophic forgetting is reduced through drift-aware stability regularization, which ensures smooth transitions between the latent space. These results are corroborated by the decrease in MCC and the augmentation in false alarms in the ablation research, indicating that stability-constrained adaptation is essential in relation to maintaining performance.

Third, prequential chronological evaluation is used to guarantee realistic high-speed streaming assessment. The proposed framework is insensitive to temporal leakage, unlike the work done before; however, the use of random splits is susceptible to real distribution changes. In comparison, the existing models, which are based on static DLs [17, 18], are sensitive to drift, retraining on a schedule is costly, RL-based models [21] are unstable, and federated models do not directly manage representation drift. In comparison, the suggested framework achieves joint optimization of adaptation, stability, and efficiency that results in a stable increase in gains across datasets.

Incremental adaptation is cheaper to train (reducing retraining by 82 percent) and has low memory (3.2 MB) and throughput (71000 flows/sec), which allows real-time deployment on IoT gateways and edge nodes. The model is flexible and can be expanded to accommodate other activities like botnet detection and anomaly detection. A weakness is less responsiveness to abrupt or adversarial drift, where momentary degradation can take place before it reaches a level that is constant.

8. Conclusion

The paper introduced a self-adaptive deep representation learning model of concept-drift-resilient DDoS detection on streaming IoT systems. Unlike the models that are either static or periodically retrained, the proposed method does the representation-level incremental adaptation, and the latent stability across time windows is explicitly maintained. The framework reduces catastrophic forgetting and preserves discriminative structure in changing traffic distributions by jointly minimizing classification error and a regularization term that is constrained by stability. This architecture allows lifelong learning without the need to retrain completely or to access future data, which is consistent with the practical deployment limits of dynamic IoT networks. A large-scale test using a prequential streaming protocol showed statistically significant improvements compared to classical and deep

learning baselines ($p < 0.01$). The proposed framework realized a 10.5%increase in Matthews Correlation Coefficient compared with the best periodic retraining baseline, less temporal degradation between sequential windows, and less retraining overhead by about 82 percent. Measurements of inference latency ensured that real-time deployment at the edge was possible. The results show that adaptive latent representation learning offers a computationally efficient and statistically strong solution to maintain long-term DDoS detection performance in non-stationary IoT settings. Even though larger Internet-scale validation is preferable, the structure provides a conceptual basis for drift-conscious IoT intrusion detection systems.

8.1. Future Work

There are still a number of avenues to explore. To achieve complete autonomy of deployment, it is necessary to reduce dependence on labeled data, and adding unsupervised or self-supervised adaptation, including contrastive learning or reconstruction-based objectives, can allow drift-aware learning to update without constant annotation. Responsiveness to sudden changes in distribution could also be improved by integrating explicit statistical drift detection modules, which dynamically activate adaptation instead of using fixed temporal windows. It would also be beneficial to extend the framework to privacy-conserving distributed environments by means of federated or collaborative learning paradigms to make the framework applicable to heterogeneous IoT ecosystems. Further research is needed to assess the adversarial resilience in case of feature manipulation or poisoning, and to make sure that the system is resistant to advanced attack techniques. The compression techniques of bypassing the pruning, quantization, and hardware-aware optimization will be considered to make support on the resource-constrained edge devices possible. Finally, large-scale validation across multi-domain, geographically distributed IoT infrastructures would be more advantageous for scalability and operational readiness in real-world setups.

Ethics Approval and Consent to Participate

This article does not involve any studies with human participants or animals performed by any of the authors. Therefore, ethics approval and consent to participate are not applicable.

Conflicts of Interests

The author declares no financial or organizational conflicts of interest.

Funding Statement

There is no direct funding provided for this project.

Acknowledgements

The authors appreciate and are thankful to SVPM's College of Engineering as a Research Center for providing the necessary facilities to conduct the present research.

Data Availability

The data that support the findings of this study are available upon reasonable request.

References

[1] Mauro Conti et al., "Internet of Things Security and Forensics: Challenges and Opportunities," *Future Generation Computer Systems*, vol. 78, pp. 544-546, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[2] Abebe Abeshu Diro, and Naveen Chilamkurti, "Distributed Attack Detection Scheme using Deep Learning Approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, pp. 761-768, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[3] Monowar H. Bhuyan, D.K. Bhattacharyya, and J.K. Kalita, "Network Anomaly Detection: Methods, Systems and Tools," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 303-336, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[4] Yair Meidan et al., "N-BaIoT: Network-based Detection of IoT Botnet Attacks using Deep Autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-22, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Nour Moustafa, and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[6] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp. 108-116, 2018. [CrossRef] [Google Scholar]

[7] João Gama et al., "A Survey on Concept Drift Adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1-37, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[8] Albert Bifet, and Ricard Gavaldà, "Learning from Time-Changing Data with Adaptive Windowing," *Proceedings of the SIAM International Conference on Data Mining*, pp. 443-448, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[9] Saad Khan, Simon Parkinson, and Yongrui Qin, "Fog Computing Security: A Review of Current Applications and Security Solutions," *Journal of Cloud Computing*, vol. 6, no. 1, pp. 1-22, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[10] Qiang Yang et al., "Federated Machine Learning: Concept and Applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1-19, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[11] Apoorva Gupta et al., "A Review on Machine Learning Techniques for DDoS Attack Detection in IoT," *2022 4th International Conference on Artificial Intelligence and Speech Technology (AIST)*, Delhi, India, pp. 1-6, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[12] John Kindervag, "*Build Security into Your Network's DNA: The Zero Trust Network Architecture*," Forrester Research, Technical Report, pp. 1-16, 2010. [Google Scholar]

[13] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao, "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. 1-42, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[14] Arash Habibi Lashkari et al., "Characterization of Tor Traffic using Time-based Features," *Proceedings of the International Conference on Information Systems Security and Privacy (ICISSP)*, vol. 1, pp. 253-262, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[15] Chuanlong Yin et al., "A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks," *IEEE Access*, vol. 5, pp. 21954-21961, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[16] Sydney Mambwe Kasongo, and Yanxia Sun, "A Deep Learning Method with Filter-based Feature Engineering for Wireless Intrusion Detection," *IEEE Access*, vol. 7, pp. 38597-38607, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Liang Xiao et al., "PHY-Layer Spoofing Detection with Reinforcement Learning in Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10037-10047, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[18] Jihyun Kim et al., "Long Short Term Memory Recurrent Neural Network Classifier for Intrusion Detection," *2016 International Conference on Platform Technology and Service (PlatCon)*, Jeju, Korea (South), pp. 1-5, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[19] Pascal Vincent et al., "Extracting and Composing Robust Features with Denoising Autoencoders," *Proceedings of the 25th International Conference on Machine Learning*, New York, NY, United States, pp. 1096-1103, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yuancheng Li, Rong Ma, and Runhai Jiao, "A Hybrid Malicious Code Detection Method based on Deep Learning," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 205-216, 2015. [CrossRef] [Google Scholar]

[21] Zhuo Chen et al., "XGBoost Classifier for DDoS Attack Detection and Analysis in SDN-based Cloud," *2018 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Shanghai, China, pp. 251-256, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] Muhammad Umair et al., "Hierarchical Federated Learning Approach for IoT Attacks Classification," *IEEE Access*, vol. 14, pp. 65276-65291, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[23] Francesco Restuccia, Salvatore D'Oro, and Tommaso Melodia, "Securing the Internet of Things in the Age of Machine Learning and Software-Defined Networking," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4829-4842, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mahdi Rabbani et al., "Device Identification and Anomaly Detection in IoT Environments," *IEEE Internet of Things Journal*, vol. 12, no. 10, pp. 13625-13643, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[25] Syaifuddin Saif, Ridi Ferdiana, and Widyawan Widyawan, IoT-DH Dataset, Mendeley Data, vol. 1, 2024. [Online]. Available: https://data.mendeley.com/datasets/8dns3xbckv/1

[26] Manasa Koppula, and L.M.I. Leo Joseph, "A Real-World Dataset "IDSIoT2024" for Machine Learning/Deep Learning Based Cyber Attack Detection System for IoT Architecture" *2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, pp. 1757-1764, 2025. [CrossRef] [Google Scholar] [Publisher Link]