Computer Networks 272 (2025) 111712

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Adaptive memory replay for network intrusion detection: Tackling data drift and catastrophic forgetting

Nasreen Fathima A H [a,b], Ansam Khraisat [a], Syed Ibrahim S P [b,*], Gang Li [a]

[a] School of Information Technology, Deakin University, Burwood, Melbourne, 3125, Victoria, Australia
[b] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, India

ARTICLE INFO

Keywords:
Continual learning
Catastrophic forgetting
Task-aware learning
Evolving network attacks
Network intrusion detection systems (NIDS)
Incremental learning
Signature attacks
Dynamic network
Memory replay

ABSTRACT

Network intrusion detection aims to identify anomalous activities in network traffic, while continual learning (CL) methods strive to preserve past knowledge and adapt to evolving threats. Memory replay-based CL approaches have been widely used and proven effective at mitigating catastrophic forgetting. However, previous research has primarily focused on addressing class imbalance and has largely relied on augmented and random memory replay strategies, which introduce significant computational overhead and limit practicality in real-time applications. To overcome these challenges, we propose Task-Aware Memory Replay (TAMR), a novel framework that prioritizes past experiences based on their relevance to the current task. By dynamically adjusting the importance of replayed samples, TAMR balances the integration of new attack patterns with the retention of critical historical knowledge, ensuring resilience against evolving threats and variations in normal traffic. Unlike traditional methods that employ random selection or augmented replays, TAMR selectively replays high-impact experiences, thereby optimizing memory usage and improving adaptability. Our experiments demonstrate that TAMR achieves real-time adaptability across five distinct NIDS datasets, ultimately delivering superior performance and computational efficiency in detecting even unknown attacks in dynamic network environments. In general, we highlight the potential of memory-based replay strategies for continual learning in detecting unknown attacks using a task-aware approach.

1. Introduction

Network Intrusion Detection Systems(NIDS) for networks play a vital role in identifying and blocking malicious actions within network traffic.[1]. Most existing NIDS fail to detect subtle or novel attacks due to inefficiencies in handling diverse threat scenarios [2]. A critical challenge in this context is data drift, a phenomenon where shifts in network traffic patterns degrade detection accuracy over time [3]. This issue stems from catastrophic forgetting, where models lose previously learned attack patterns when integrating new ones [4].

The static architecture of traditional NIDS models further exacerbates these challenges [5]. Their inflexibility leads to a gradual decline in effectiveness against emerging threats, leaving networks increasingly vulnerable. To remain effective, these models require frequent retraining from scratch, a resource-intensive process that is impractical in dynamic and fast-paced cyber environments [6]. Moreover, this retraining not only demands significant computational resources but also risks erasing prior knowledge, as these models are prone to catastrophic forgetting when exposed to new attack patterns.

Continual Learning (CL) has emerged as a powerful strategy that enables models to incrementally learn from new data while mitigating the issue of catastrophic forgetting. [7]. It is imperative to transition from static NIDS frameworks to more adaptive systems capable of continuous learning [8]. Such systems would enable the integration of new threat intelligence while preserving knowledge of previously encountered attacks, thereby enhancing their resilience and effectiveness [9]. By incorporating mechanisms that support lifelong learning and adaptability, NIDS can better withstand the challenges posed by data drift, catastrophic forgetting, and the ever-evolving nature of cyber threats. This paradigm shift is essential for developing robust and future-proof cybersecurity solutions.

Recent studies highlight the importance of continual learning strategies, particularly replay-based methods [10], in mitigating catastrophic forgetting within NIDS frameworks [8,11]. By employing

* Corresponding Author.
E-mail addresses: s223992237@deakin.edu.au (N.F. A H), ansam.khraisat@deakin.edu.au (A. Khraisat), syedibrahimsp@gmail.com, syedibrahim.sp@vit.ac.in (S.I. S P), gang.li@deakin.edu.au (G. Li).

https://doi.org/10.1016/j.comnet.2025.111712
Received 15 April 2025; Received in revised form 16 August 2025; Accepted 9 September 2025
Available online 11 September 2025

memory-efficient replay mechanisms, NIDS can retain past attack patterns while integrating new threats in real time [12]. However, conventional replay methods rely heavily on outdated attack patterns and predominantly focus on augmentation sampling and random sampling to address class imbalance, offering limited practical applicability in real-world dynamic environments. Addressing these limitations requires an adaptive, task-aware replay strategy that selects and retains knowledge based on emerging attack behaviors and network conditions. Random or augmented memory replay methods struggle to adapt to shifting attacker behaviors, evolving network configurations, and newly discovered vulnerabilities. Moreover, these approaches fail to consider task-specific variations, such as differences in attack signatures, network traffic characteristics, and threat severity levels [3].

To overcome the limitations of existing replay strategies, we propose Task-Aware Memory Replay (TAMR), a novel framework that selectively replays task-relevant attack patterns while dynamically adapting to evolving threats. Unlike prior methods, TAMR introduces a task-relevance scoring mechanism that quantifies the relevance of each incoming sample x from task t by comparing it with stored experiences. This enables the system to prioritize replay based on intrinsic task relevance rather than relying on explicit task labels.

Operating within an incremental learning paradigm, TAMR leverages task-specific cues encountered during training to effectively retain past knowledge while seamlessly incorporating new information. This approach contrasts with traditional Task-Incremental Learning (TIL) methods, which assume known task boundaries, a constraint rarely met in real-world network intrusion detection deployments. By dynamically adjusting replay priorities according to the internal relevance of stored experiences to current inputs, TAMR enables targeted memory replay that both preserves critical historical attack signatures and integrates novel threat behaviors. This task-aware prioritization enhances the adaptability and long-term effectiveness of network intrusion detection systems (NIDS) operating in continuously evolving environments.

A robust NIDS must detect emerging cyber threats without compromising knowledge of well-known signature attacks, ensuring sustained resilience against evolving attack strategies. TAMR achieves this balance between stability and plasticity by selectively replaying critical tasks to prevent knowledge degradation while facilitating continuous learning. To realize this balance, TAMR comprises four core components: Task Identification, Memory Buffer Update, Experience Replay Module, and Model Refinement. Together, these components collaboratively optimize memory management, improve learning efficiency, and mitigate interference from new tasks.

This structured framework ensures effective adaptation to emerging threats while preserving previously acquired knowledge. Extensive experiments on five benchmark NIDS datasets demonstrate that TAMR consistently improves detection of both minor and novel attacks while mitigating catastrophic forgetting. The results underscore TAMR's capacity to handle persistent and emerging cyber threats in dynamic network environments, making it highly suitable for real-world NIDS applications where attack patterns continuously evolve.

The structure of this paper is as follows: Section 2 provides an overview of key concepts, the evolution of NIDS, and the underlying challenges. Section 3 examines relevant research in the field. Section 4 presents the architecture and functional process of the proposed framework. Section 5 outlines the experimental configuration and provides a detailed analysis of the results. Finally, Section 6 highlights the key findings and proposes potential avenues for future work.

2. Background

This section presents a detailed examination of the fundamental concepts necessary to grasp the challenges tackled in this study. It introduces key definitions, outlines the learning paradigms within continual learning (CL) for NIDS, reviews related work, and identifies existing research gaps.

2.1. Preliminaries

In this section, we discuss the evolution of NIDS, the core concepts of the problem and an overview of CL based NIDS for attack detection in dynamic network traffic.

2.1.1. Evolution of network intrusion detection systems (NIDS)

NIDS have transitioned from conventional rule-based and statistical techniques to more sophisticated machine learning (ML) methods [13]. Early NIDS relied on rule-based detection, which matched network traffic patterns to predefined attack signatures. While highly accurate for known threats, these systems failed to detect novel or variant attacks, limiting their adaptability in dynamic environments [14]. To overcome these limitations, statistical anomaly detection was introduced, modeling normal network behavior and flagging deviations as potential threats [15]. Although more flexible than rule-based methods, these approaches required significant computational resources and suffered from high false positive rates, particularly in complex network environments [5,16].

Contemporary NIDS now utilize hybrid detection methods that integrate signature-based and anomaly-based techniques to enhance their adaptability [17]. As cyber threats grow increasingly sophisticated, Machine Learning-based NIDS have gained attraction. Supervised learning techniques, including Naive Bayes and Support Vector Machines (SVMs), have been used to categorize network traffic into normal and malicious classes.[18]. These techniques improved detection accuracy but struggled with dynamic traffic patterns and unseen attack types [1,19]. To address these challenges, deep learning models, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been widely adopted due to their ability to capture complex and temporal patterns in network traffic [4,17,20]. While hybrid systems and deep learning significantly enhances threat detection by recognizing both known and unknown attacks, they still struggle with evolving network traffic patterns and suffer from catastrophic forgetting, where previously learned attack patterns are lost when new ones are introduced [21,22]. With each new learning phase, the model faces the risk of losing previously acquired knowledge, posing a continuous challenge for long-term adaptation.

2.1.2. Catastrophic forgetting

Definition 1 (Catastrophic Forgetting in NIDS)**.** Within NIDS, catastrophic forgetting describes the gradual erosion of earlier acquired knowledge as the model is trained on new attack patterns[2].

This occurs because the model overwrites existing representations while learning new information, causing a decline in its ability to detect earlier attack signatures. As a result, the model becomes biased toward recently seen threats, leading to reduced detection accuracy for previously encountered attacks. This challenge is particularly problematic in dynamic network environments, where new threats continuously emerge, and maintaining long-term memory of diverse attack patterns is essential. Catastrophic forgetting is quantitatively assessed by the deterioration in a model's accuracy on a previously acquired task T_i following the assimilation of a new task T_{new}:

$$\Delta P(T_i) = P(T_i|\theta_{\text{new}}) - P(T_i|\theta_{\text{old}}) \tag{1}$$

where $P(T_i)$ denotes the evaluation metric (such as accuracy or F1-score) for task T_i, θ_{old} denotes the model parameters before learning T_{new}, θ_{new} represents the model parameters after learning T_{new}, and a large negative $\Delta P(T_i)$ indicates significant forgetting of T_i.

2.1.3. Continual learning

In practical applications, data usually arrives in batches at consistent intervals, ranging from minutes to hours or days based on the specific requirements of the organization [23]. Continual Learning (CL) allows models to learn incrementally, retaining knowledge from previous experiences while adapting to new information [18]. By leveraging prior

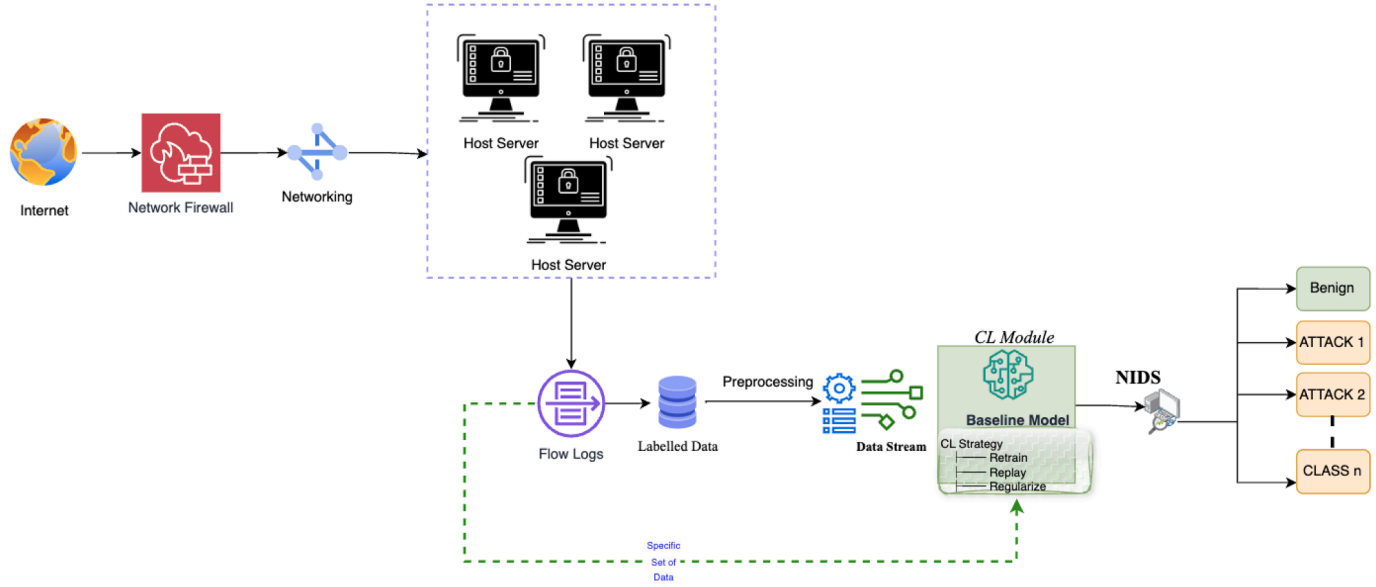


Fig. 1. Continual learning pipeline in NIDS.

knowledge, CL significantly reduces computational and storage costs, while enhancing the adaptability of machine learning models [21]. This ability to continually learn from evolving data is especially crucial in fields like cybersecurity, where new and emerging threats appear constantly.

Incremental Learning Paradigms. However, when learning incrementally, models face the challenge of retaining previously learned information while adapting to new tasks. Task-Incremental Learning (Task-IL) addresses this by treating each task as an independent unit with well-defined boundaries, allowing the model to mitigate catastrophic forgetting [24]. This might lead attack patterns frequently overlap with previously encountered ones, making it difficult for Task-IL to generalize to new or blended attack types. While a Class-Incremental Learning (Class-IL) removes predefined task boundaries and allows the model to integrate new attack classes continuously [25,26]. While this improves adaptability by learning new attacks without forgetting older ones, it also introduces the challenge of class interference. In contrast, Domain-Incremental Learning (DIL) emphasizes adjusting to changes in the data distribution, while maintaining the task and label space constant [27].

2.1.4. Continual learning for NIDS

In the context of Network Intrusion Detection Systems (NIDS), CL plays a pivotal role in improving detection capabilities by ensuring that models remain effective against both known and emerging attack patterns [28]. The objective is to maintain high detection accuracy for both old and new attack classes without sacrificing efficiency. The primary challenge in this domain is preserving previously acquired knowledge while integrating new information. CL enables models to incrementally learn from a continuous stream of tasks. This adaptability is particularly crucial in cybersecurity, where new and evolving threats emerge dynamically [29]. At each task interval, the incoming data may include previously unseen classes or additional instances of known ones. As time progresses, new types of attacks can emerge, requiring the system to learn these classes while preserving knowledge of earlier ones. This ability is referred to as incremental class learning [30]. In addition, the characteristics of previously observed data may change due to the evolving nature of cyber threats, resulting in dataset drift. Addressing both the emergence of new classes and changes in existing ones is a central challenge for continual learning in Network Intrusion Detection Systems [23].

Fig. 1 depicts the integration of a continual learning (CL) pipeline into a Network Intrusion Detection System (NIDS) to enable real-time adaptation to emerging cyber threats. By leveraging CL strategies such as retraining, replay, and regularization, the NIDS can dynamically update its knowledge while preserving previously learned attack patterns [11]. Specifically, retraining enables the system to fine-tune on new data, adapting to recent threat trends; replay mitigates forgetting by reintroducing representative past examples during learning; and regularization constrains model updates to prevent significant deviations from previously acquired knowledge. However, integrating CL into high-velocity network environments introduces significant challenges, particularly in ensuring efficient knowledge retention, computational scalability, and adaptive decision-making to mitigate catastrophic forgetting.

2.2. Problem statement

NIDS play a vital role in cybersecurity by continuously monitoring and analyzing network traffic to identify potential threats or malicious behavior. However, traditional NIDS methods face significant challenges in adapting to the ever-evolving nature of cyberattacks and dynamic network conditions. CL has emerged as a promising solution to mitigate the problem of catastrophic forgetting by allowing NIDS to learn progressively and adapt to new threats. While CL techniques have shown potential, current methods often fail to address the task-specific characteristics of different attack patterns, which leads to the loss of critical knowledge over time. Moreover, the presence of data drift in network traffic characteristics evolve due to new attack strategies, changing protocols, or shifting network behaviors which further complicates the ability of NIDS to maintain accurate detection capabilities. Most approaches rely on static training or outdated knowledge, making them ineffective in real-world environments where threats and network behaviors constantly change. These issues hinder the performance of CL-based NIDS, especially in real-time environments where timely and accurate detection is paramount.

3. Related work

This section explores the key categories of Continual Learning (CL) strategies relevant to this work. A summarized overview of the reviewed approaches is presented in Table 1, which organizes various CL strategies based on their core principles, representative techniques, and associated limitations. These include methods based on regularization, experience replay, memory replay, and architectural adaptation, each

Table 1
Overview of continual learning methods.

Methods	Description	Strategy	Limitations	Ref.
Regularization-Based Methods	Constrains weight updates to preserve previously learned knowledge.	EWC, SI, LWF	May not fully prevent forgetting in highly dynamic environments.	[31,33]
Experience Replay Algorithms	Replays stored training data to reinforce past knowledge by repeatedly exposing the model to prior experiences.	Experience Replay	High memory usage and increased computational cost.	[7,10]
Replay-Based Methods	Stores a subset of previously learned data and rehearses past tasks to mitigate forgetting.	GEM, Latent Replay	Storage limitations and potential interference between tasks.	[34,35]
Memory Replay-Based Methods	Maintains a memory buffer of diverse past experiences to improve model retention.	MIR, CBRS, ECBRS, PAPA	Requires effective memory management to balance past and new information.	[8,36]
Architecture-Based Methods	Utilizes modular architectures to efficiently learn new tasks and adapt to changing data.	SPIDER, SOUL, CLAW	Increased model complexity and resource requirements.	[14,37,38]

Table 2
Abbreviations of CL strategies and learning paradigms.

Abbreviation	Full Form
EWC	Elastic Weight Consolidation
SI	Synaptic Intelligence
LWF	Learning Without Forgetting
GEM	Gradient Episodic Memory
MIR	Maximally Interfered Retrieval
CBRS	Class-Balanced Reservoir Sampling
ECBRS	Enhanced Class-Balanced Reservoir Sampling
PAPA	Perturbation Assistance for Parameter Approximation
SPIDER	Semisupervised Privacy-preserving Intrusion Detection with Drift-aware Continual Learning
SOUL	Semi-supervised Open-world ContinUal Learning
CLAW	Continual Learning with Adaptive Weights
TIL	Task Incremental Learning
CIL	Class Incremental Learning
DIL	Domain Incremental Learning

contributing uniquely to mitigating forgetting in evolving environments such as Network Intrusion Detection Systems (NIDS). Table 2 provides the full forms of the abbreviations used throughout this section, offering a concise reference to support clarity and readability.

3.1. Regularization-based methods

Regularization based approaches address the issue of catastrophic forgetting by limiting changes to model parameters, helping to retain knowledge acquired from earlier tasks. Strategies such as Elastic Weight Consolidation (EWC) and Synaptic Intelligence (SI) achieve this by employing task-specific regularization coefficients and dynamically adjusting regularization strength [31]. EWC stabilizes crucial weights in a neural network by penalizing changes to them. It identifies important parameters using the Fisher information matrix, ensuring that critical knowledge is retained while learning new tasks [32]. Similarly, Learning without Forgetting (LwF) [33] balances knowledge retention and adaptation by leveraging multitasking. However, these methods struggle in highly dynamic environments that require rapid and frequent updates to adapt to evolving attack patterns. The reliance on weight constraints limits model flexibility, making it difficult to integrate novel attack variations while maintaining robustness.

3.2. Experience replay methods

Catastrophic forgetting is mitigated by replaying stored training data, as demonstrated in [10] and [7]. This approach reinforces previously acquired knowledge by repeatedly exposing the model to past experiences, helping to consolidate information and maintain performance across tasks over time. However, replaying stored data can be computationally expensive and require significant storage capacity. Additionally, it may be ineffective in nonstationary environments, where outdated information could lead to overfitting. To address these limitations, strategies include prioritizing sample replay, employing diverse sampling techniques, and incorporating temporal consistency constraints.

3.3. Replay-based strategies

Replay-based strategies retain a portion of previously learned data and revisit past tasks to prevent catastrophic forgetting. These methods involve selecting representative samples, prioritizing important instances, and balancing rehearsal with new task learning [35]. By maintaining a subset of past samples, replay-based methods reinforce prior knowledge and enhance retention during training. Gradient Episodic Memory (GEM) [34] is a commonly adopted replay strategy that addresses catastrophic forgetting by retaining previous examples in an episodic memory buffer. This allows the model to review prior tasks while learning new ones, preserving critical knowledge.

Another approach, Latent Replay [35], enhances model performance by storing past activations and comparing new inputs with these stored representations. This method is particularly beneficial for anomaly detection, as it reduces false positives by contextualizing new data against previously encountered patterns. Despite their effectiveness, traditional replay methods often suffer from inefficient memory management, leading to redundant or suboptimal sample selection, which dilutes the model's focus on emerging threats [24]. Many replay techniques rely on random sampling or heuristic selection, introducing non-informative replay. Additionally, fixed replay buffers struggle to adapt to evolving attack distributions, potentially overlooking new, high-risk threats.

3.4. Memory replay-based strategies

Approaches such as Experience Replay and Memory Replay maintain a buffer of diverse past experiences to improve model retention. It is crucial for continual learning in Network Intrusion Detection Systems

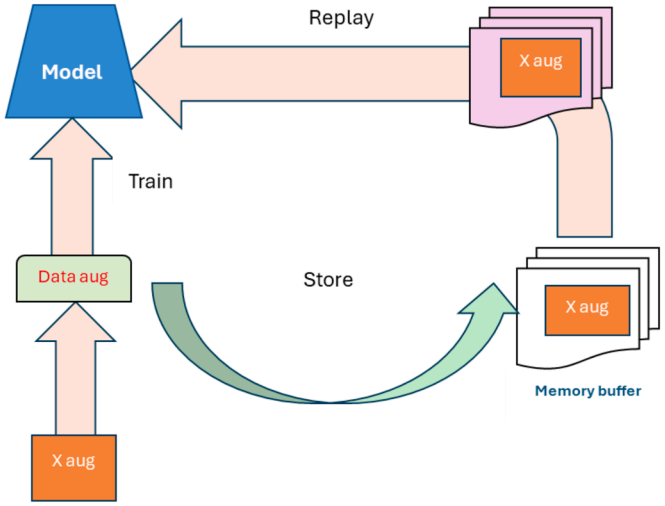


Fig. 2. Graphical representation of augmented memory replay on samples.

(NIDS) to enhance efficiency, Maximally Interfered Retrieval (MIR) [36] prioritizes samples most likely to interfere with new task learning, improving adaptation. However, its computational cost makes it less practical for real-time intrusion detection. These limitations underscore the need for scalable and adaptive replay strategies that dynamically manage memory and sample selection without excessive resource demands. Fig. 2 depicts the augmented memory replay on samples.

A key challenge in NIDS datasets is the significant class imbalance, where infrequent attack types are poorly represented. Methods like Class-Balanced Reservoir Sampling (CBRS) and its extension, Extended Class-Balanced Reservoir Sampling (ECBRS) [8], ensure adequate replay of minority classes. While these approaches improve detection of rare attacks, they rely on static heuristics and may not dynamically adjust to real-time attack distributions. To address this, the author [8] employed Perturbation Assistance for Parameter Approximation (PAPA), which prioritizes critical samples during training, significantly reducing training time without sacrificing accuracy. This makes PAPA well-suited for high-dimensional datasets like CIC-IDS. Experimental results demonstrate that CBRS and ECBRS enhance the detection of underrepresented threats, improving precision and recall. However, class imbalance is just one part of the challenge; real-time environments also require adaptive strategies that consider task-specific attack patterns. Traditional methods often face difficulty maintaining essential knowledge when adapting to emerging threats, resulting in a gradual decline in performance if a structured replay mechanism is not employed.

3.5. Architecture-based methods

Architecture-based approaches primarily aim to adjust intrusion detection models to evolving attack patterns while preserving knowledge of previously encountered threats. SPIDER is a semi-supervised, privacy-preserving intrusion detection framework with Drift-aware Experience Replay [39] . This approach enables adaptation to evolving attack patterns while retaining past knowledge. While such frameworks help mitigate challenges like class imbalance, they tend to focus more on labeling issues than on addressing the dynamic nature of real-time network traffic. As new attack vectors emerge and normal traffic patterns evolve, models must continuously learn while preserving the ability to detect previously identified threats. Incorporating techniques like dynamic modular architectures, Semi-supervised Open-world Continual Learning (SOUL), and adaptive learning rates can improve model flexibility [14],[38],[37]. However, balancing the learning of new threats with the retention of prior knowledge remains a significant challenge due to the unpredictable nature of real-time network traffic. Strategies such as GPM and Experience Replay are useful in reducing catastrophic forgetting, but they require careful tuning to ensure effectiveness and scalability. If a model adapts too quickly, it may forget earlier threats, while if it adapts too slowly, it may fail to recognize new attacks. Thus, task-specific replay strategies are crucial for ensuring both adaptability to evolving threats and the retention of essential knowledge.

3.6. Research gaps and motivation

Traditional replay strategies in continual learning, such as random sampling, uniform selection, and semi-supervised methods like CBRS, ECBRS, PAPA, MIR, and SPIDER, often fail to account for the structured nature of network traffic in intrusion detection. These approaches either treat all stored instances equally or focus primarily on class imbalance and labeling, leading to several key challenges.

First, random sampling neglects the prioritization of critical attack samples, increasing the likelihood of forgetting rare but high-risk threats. While data augmentation can mitigate scarcity, it introduces additional computational overhead and risks overfitting when synthetic samples fail to accurately represent real-world attacks. Moreover, conventional replay methods disregard differences in attack patterns, packet structures, and traffic behaviors, resulting in concept drift, where newly encountered attacks overwrite previously learned information.

Class imbalance further exacerbates the problem. Common attack types dominate memory, while rare but high-impact threats are often underrepresented, leaving systems vulnerable. Static memory selection is ineffective in dynamic environments, as it retains redundant data while discarding crucial threat indicators. Without a task-aware mechanism, these limitations significantly hinder the detection of emerging attacks, increasing the risk of undetected intrusions.

To address these challenges, an adaptive, task-aware replay strategy is essential. Such an approach should:

- Prioritize relevant attack patterns rather than relying on random selection.
- Retain rare yet high-impact threats while preventing memory saturation from redundant data.
- Adapt dynamically to network traffic variations without excessive computational overhead.

These considerations drive the need for a more intelligent and efficient replay mechanism to enhance intrusion detection in continually evolving cyber environments.

4. Task-aware memory replay (TAMR): design and implementation

This section presents our proposed method, *Task-Aware Memory Replay* (TAMR), which incorporates a memory replay strategy within a continual learning (CL) framework. TAMR is designed to address the limitations of traditional replay techniques by introducing a task-aware memory buffer combined with model refinement mechanisms. These components enable the model to adapt effectively to changes in the dataset and evolving network behaviors. By selectively replaying task-relevant information, TAMR mitigates catastrophic forgetting and strengthens the continual learning pipeline, directly addressing the challenges and research gaps highlighted in the preceding section.

4.1. Framework overview

The proposed framework is organised into four components, illustrated in Fig. 3. TAMR utilizes a replay memory buffer ($\mathcal{M}_t$) to selectively retain representative samples from previously learned attack patterns. Unlike conventional NIDS approaches, which rely on static training or periodic retraining, TAMR continuously integrates new task

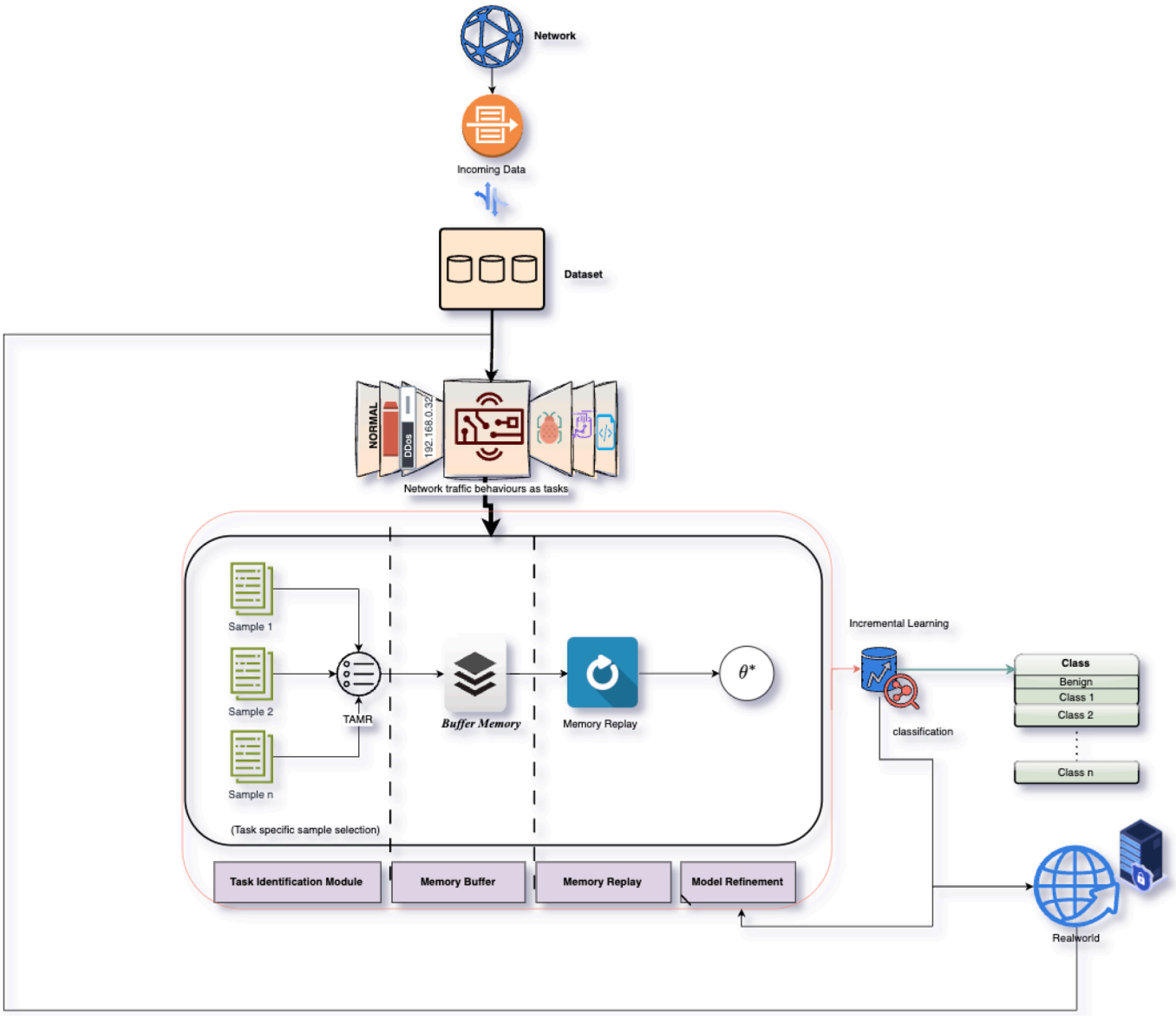


Fig. 3. System architecture of the proposed TAMR framework.

data (T_{t+1}) with past experiences. This integration is achieved by replaying stored instances alongside incoming data during training, reinforcing historical knowledge and mitigating catastrophic forgetting. Traditional NIDS models typically use fixed architectures that require full retraining to adapt to new threats. In contrast, TAMR's dynamic replay mechanism allows for incremental updates, ensuring the system retains essential historical attack patterns while incorporating new information. This continuous learning paradigm overcomes the limitations of static models in evolving cyber environments.

In TAMR, the labeling process is tightly integrated with the continual learning framework. The system reinforces the association between attack patterns and their corresponding labels by replaying past samples during training, ensuring that labels remain consistent and relevant as network conditions and threat landscapes evolve. While standard classification models are designed to assign labels based on learned features in a fixed, one-time manner, TAMR continuously refines its decision boundaries. Continuous refinement is essential for handling label drift, allowing the model to adjust to new threats while preserving its ability to recognize previously learned classes. TAMR incorporates iterative model refinement based on performance evaluations, aiming to minimize cumulative loss across all tasks. This iterative process ensures the model remains robust and effective in real-time environments, eliminating the need for complete retraining. To summarize, TAMR presents an innovative and flexible framework for network intrusion detection. The four components of the proposed framework detailed as follows. The detailed workflow and operational phases are explained in the subsequent sections and visually represented in Fig. 4.

4.1.1. *Task identification*

The Task Identification Module (TAM) forms the foundation of our continual learning framework by organizing raw, pre-labeled network traffic into structured tasks. This organization enables incremental model updates while preserving contextual information vital for distinguishing between benign and malicious behaviors. Each incoming traffic sample is assigned a unique Task ID corresponding to its category (benign or specific attack type), which is propagated downstream to maintain per-task metadata and support targeted learning.

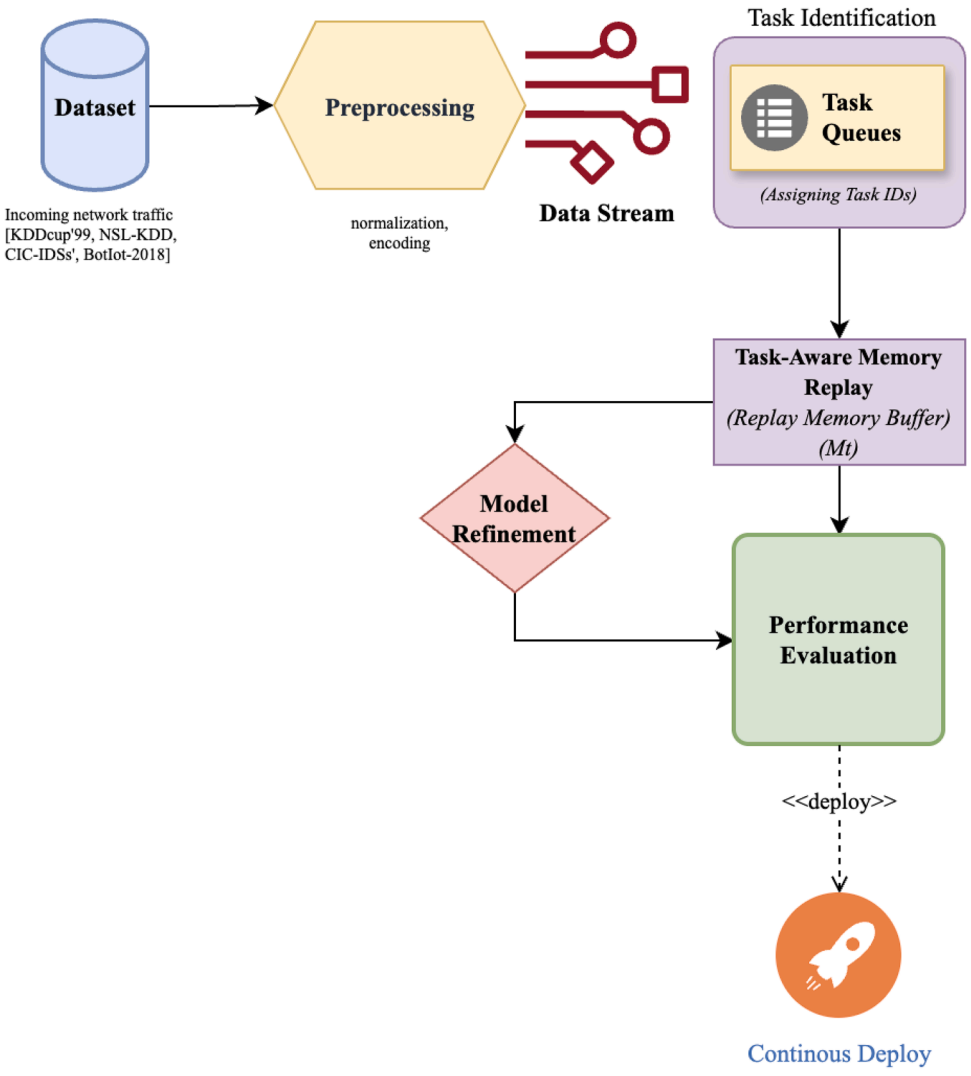


Fig. 4. Workflow overview of the proposed framework.

Task Segmentation: Pre-labeled network traffic is partitioned into distinct tasks based on their unique class labels. Each task represents a homogeneous group of samples belonging to a specific category, such as benign traffic or an attack type (e.g., DoS, Probe, R2L/U2R, Botnet). This segmentation leverages inherent data attributes to cluster samples, enabling focused learning on category-specific characteristics.

Task Representation: Following segmentation, each task $\mathcal{T}_{t+1}$ is formalized as a dataset:

$$\mathcal{T}_{t+1} = \{(x_j, y_j)\}_{j=1}^{|\mathcal{T}_{t+1}|}, \quad x_j \in \mathbb{R}^d, \quad y_j \in \{0, 1, \ldots, k-1\}, \tag{2}$$

where x_j denotes the feature vector for a network traffic sample, and y_j its class label. Here, k is the total number of distinct categories. Assigning a unique Task ID T_t to each dataset enables TAM to distinguish new from prior knowledge, allowing the learning process to concentrate on one task at a time. This organization helps reduce interference between tasks and mitigates catastrophic forgetting.

Task-Relevance Score. To guide replay selection, we compute a task-relevance score $\mathrm{Rel}(x, t)$ for each sample $x \in \mathbb{R}^d$ with label $y \in \{0, \ldots, k-1\}$ in task $\mathcal{T}_t$:

$$\mathrm{Rel}(x,t) = w_s \cdot \frac{1}{1 + \|x - \pi_t\|_2} + w_r \cdot \frac{1}{\mathrm{freq}_t(y) + \varepsilon} + w_u \cdot \exp\big(-\lambda(T_{\mathrm{now}} - T_x)\big), \tag{3}$$

where π_t is the prototype (mean) vector of task t, $\mathrm{freq}_t(y)$ counts how often class y appears in the memory for task t, T_x and T_{now} are sample and current timestamps respectively, $\varepsilon > 0$ prevents division by zero, and $\lambda > 0$ controls the recency decay rate. Weights $w_s, w_r, w_u \geq 0$ balance similarity (favoring samples close to the task prototype), rarity (emphasizing underrepresented classes), and recency (prioritizing newer samples), with $w_s + w_r + w_u = 1$. This relevance score helps prioritize samples that are representative, rare, and recent, thereby improving replay effectiveness and reducing forgetting.

4.1.2. Memory buffer update

The Memory Buffer plays a pivotal role in preserving historical knowledge by storing representative samples from tasks encountered before. This mechanism is essential for continuous learning systems, as it prevents the model from overfitting to the most recent data and forgetting earlier tasks (shown in Fig. 3). For each new task $\mathcal{T}_{t+1}$, the system selects a subset $S_{t+1} \subset \mathcal{T}_{t+1}$ that captures the diversity of the task's data. These representative samples are then integrated into the existing memory buffer according to:

$$\mathcal{M}_{t+1} = \mathcal{M}_t \cup S_{t+1}. \tag{4}$$

In Algorithm 1, for each incoming sample in $\mathcal{T}_t$, we calculate $\mathrm{Rel}(x, t)$ using Eq. (3) and select the highest scoring samples for inclusion in the memory buffer $\mathcal{M}$. This ensures a balanced replay buffer representing

Algorithm 1 Memory buffer update in TAMR.

Input: Current memory buffer $\mathcal{M}$, task data $\mathcal{T}_t$, buffer capacity C
For each sample in $\mathcal{T}_t$, compute a task-relevance score using prototype alignment, rarity, and recency indicators (optionally uncertainty)
Select the most relevant samples from $\mathcal{T}_t$ and add them to $\mathcal{M}$
if $|\mathcal{M}| > C$ **then**
    Remove the oldest samples from $\mathcal{M}$ (FIFO) until $|\mathcal{M}| = C$
end if
Output: Updated memory buffer $\mathcal{M}$

task-relevant, rare, and recent samples. Representative samples are selectively retained in a fixed-capacity buffer to preserve critical historical information. On arrival, each candidate is assigned a *task-relevance score* computed. When the buffer reaches capacity, an eviction policy removes the lowest-relevance items, either within the affected task or globally, depending on the buffer policy, so the retained set remains representative of historically important examples.

To maintain a manageable size and ensure computational efficiency, the replay buffer is constrained by a fixed capacity C. When $|\mathcal{M}_{t+1}| > C$, TAMR applies a First-In-First-Out (FIFO) replacement policy to remove the oldest samples, ensuring simplicity and low computational overhead. FIFO offers deterministic behavior and avoids additional complexity in decision-making, making it well-suited for real-time intrusion detection scenarios. However, a limitation of FIFO is its potential to prematurely discard critical samples, particularly when task relevance varies significantly over time. Hence, replay selection is guided by task-relevance scores as discussed in Eq. 3, allowing the buffer to dynamically preserve high-importance samples while still respecting capacity constraints.

4.1.3. Experience replay module

Once the memory buffer is updated, the Experience Replay Module reintroduces the stored samples into the training process. The Experience Replay Module is designed to reinforce previously acquired knowledge by reintroducing historical samples into the training process. This module ensures that, while the model adapts to new data $\mathcal{T}_{t+1}$, it also revisits past information by combining it with a set of previously selected samples, thereby balancing learning and reducing the risk of forgetting earlier tasks.

Let

$$\mathcal{M}_{\text{replay}} = \{(x_i, y_i)\}_{i=1}^{|\mathcal{M}_t|} \tag{5}$$

denote the set of replayed samples extracted from the memory buffer. During training, these samples are combined with the new task data $\mathcal{T}_{t+1}$ to form a comprehensive training batch. This approach facilitates a learning environment in which the model is continuously exposed both new and historical examples. Stored samples are reintroduced into training according to task-aware selection criteria that prioritise high-relevance items. Algorithm 2 describes the training phase, which iterates over each task, shuffles the data, processes mini-batches containing both new and replayed samples, and updates the model parameters. After completing training on each task,

4.1.4. Model refinement

The Model Refinement stage integrates the previous modules into a cohesive learning strategy, enabling the system to iteratively update and optimize model parameters. The model parameters are iteratively optimised to integrate new information while preserving past knowledge. By combining the memory buffer update and the experience replay module, the framework is capable of adapting to new tasks while retaining critical information from previous ones. The optimization objective is to minimize the cumulative loss across all tasks. This objective is formulated as:

$$\theta^* = \arg\min_{\theta} \frac{1}{|\mathcal{T}^*_{t+1}|} \sum_t \Big(\mathcal{L}_t(f_{\theta_t}) + \mathcal{L}'_t(f_{\theta_t}) \Big), \tag{6}$$

where $\mathcal{T}^*_{t+1}$ represents the merged dataset comprising new task data and replayed samples. This loss function ensures that the model not only learns from current inputs but also maintains proficiency on historical tasks.

Algorithm 2 Training phase of task-aware memory replay (TAMR).

Input:
    Task sequence data $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_T\}$
    Initial model parameters θ (initialized using He initialization)
    Optimizer: Adam, Learning rate $\eta \leftarrow 0.001$
    Batch size: $B \leftarrow 32$, Epochs per task: $E \leftarrow 50$
    Regularization: L2 ($\lambda \leftarrow 0.001$), Dropout rate: 0.3
    Activation: ReLU, Loss: Cross-Entropy, Early Stopping: Yes
    Memory buffer capacity C (e.g., 1000 or 1500)
Initialize: Memory buffer $\mathcal{M} \leftarrow \emptyset$
for each task $t = 1$ **to** T **do**
    Step 1: Incorporate current task data $\mathcal{T}_t$ (see Algorithm 1)
    for epoch $e = 1$ **to** E **do**
        Shuffle $\mathcal{T}_t$
        for each mini-batch B sampled from $\mathcal{T}_t$ **do**
            Compute loss on new data:

$$L_{\text{new}} = \sum_{(x_j, y_j) \in B} \ell(f_\theta(x_j), y_j)$$

            Sample a mini-batch $\mathcal{M}_{\text{replay}}$ from memory $\mathcal{M}$
            Compute loss on replayed data:

$$L_{\text{replay}} = \sum_{(x_i, y_i) \in \mathcal{M}_{\text{replay}}} \ell(f_\theta(x_i), y_i)$$

            Combine losses:

$$L_{\text{combined}} = L_{\text{new}} + \lambda L_{\text{replay}}$$

            Update parameters:

$$\theta \leftarrow \theta - \eta \nabla_\theta L_{\text{combined}}$$

        end for
    end for
    Step 2: Update the memory buffer with samples from $\mathcal{T}_t$ (Algorithm 1)
end for
Output: Optimized model parameters θ

The proposed framework integrates key components such as task identification, memory buffer management, experience replay, and iterative model refinement to enhance the adaptability and robustness of NIDS in dynamic network environments. Task identification ensures accurate classification of incoming traffic, distinguishing known from novel threats. Memory buffer management retains critical past-task instances, optimizing storage and preserving essential knowledge. Experience replay revisits stored data to mitigate catastrophic forgetting, while iterative refinement updates decision boundaries, enabling adaptation to emerging attacks without loss of accuracy. Balancing new knowledge acquisition with historical retention enhances the resilience and long-term effectiveness of NIDS. As shown in Table 3, the key notations used throughout the proposed framework are summarized.

4.1.5. Workflow

Fig. 4, illustrates the Task-Aware Memory Replay (TAMR) workflow is designed to enable continual learning in a NIDS. The workflow begins with continuous monitoring of network traffic from structured datasets. The raw traffic undergoes preprocessing, where feature extraction, normalization, and encoding are applied. Labeled data, including benign

Table 3
Summary of key notations.

Notation	Description
$\mathcal{X}$	Input space representing network traffic features.
$\mathcal{Y}$	Output space including attack categories and benign activities.
t	Task identifier corresponding to a specific attack type or network event.
$\mathcal{M}_t$	Memory buffer after task t, storing past samples for replay.
$\mathcal{M}_{\text{replay}}$	Subset of $\mathcal{M}_t$ used during training for replay.
θ_t	Model parameters after training on task t.
$\ell(\theta;(x,y))$	Loss function for the input-label pair (x, y).
C	Maximum capacity of the memory buffer.
B	Mini-batch size for training.
L_{new}	Loss computed on new task data.
L_{replay}	Loss computed on replayed samples.
L_{total}	Combined loss used for model update.
$f_\theta(x)$	Model prediction function parameterized by θ.

Table 4
Overview of datasets used in the study.

Dataset	Description
KDD Cup 99 [40,41]	A widely used benchmark dataset for Intrusion Detection Systems (IDS), covering various attack types.
NSL-KDD [1,15,42]	An enhanced version of KDD Cup 99, addressing redundancy issues to improve IDS evaluation.
CIC-IDS 2017 [8,43]	Captures real-world network traffic and diverse attack scenarios to assess IDS performance.
CIC-IDS 2018 [17,18,43]	An extended version of CIC-IDS 2017 with additional attack types and features for comprehensive IDS evaluation.
Bot-IoT 2018 [4,44,45]	Specializes in IoT-based network attacks, aiding in intrusion detection for IoT environments.

traffic and various attack categories (e.g., DoS, Probe, R2L/U2R, Botnet), is then fed into the continual learning pipeline for incremental updates. Instead of training the model in a static manner, TAMR follows a task-aware incremental learning approach. The model initially learns from a subset of tasks, and new attack types (tasks) are introduced sequentially over time, mimicking real-world emerging threats. This approach allows the model to retain knowledge of previously learned attacks while adapting to new ones.

5. Results and analysis

This section evaluates the proposed framework through experiments conducted on multiple benchmark NIDS datasets. The primary objective is to assess the framework's ability to detect both known and emerging threats while maintaining stable performance over time, adaptability, and mitigating catastrophic forgetting. We first outline the dataset preparation process, followed by the experimental setup and key performance metrics used to demonstrate the framework's efficacy.

5.1. Datasets

To comprehensively evaluate the proposed framework, we utilize multiple benchmark NIDS datasets, each representing diverse intrusion scenarios. These datasets encompass a range of attack types, ensuring a robust assessment of the framework's ability to detect both known and emerging threats. Table 4 provides a detailed summary of the datasets used in this study.

Each of these datasets plays a crucial role in advancing research in network intrusion detection, offering diverse and comprehensive data that facilitate the development of adaptive IDS models [46]. Unlike traditional machine learning paradigms that rely on predefined training and testing splits (e.g., 80-20 or 70-30), our approach adopts a continuous learning framework. In this setting, data is processed as an evolving stream, where the model incrementally learns from new instances while preserving previously acquired knowledge. This method ensures that the IDS remains adaptive to emerging threats without catastrophic forgetting. Performance evaluation is conducted dynamically by assessing the model's capability to detect previously acquired attack patterns, along with novel threats encountered in the incoming data stream.

5.2. Preserving class distribution:

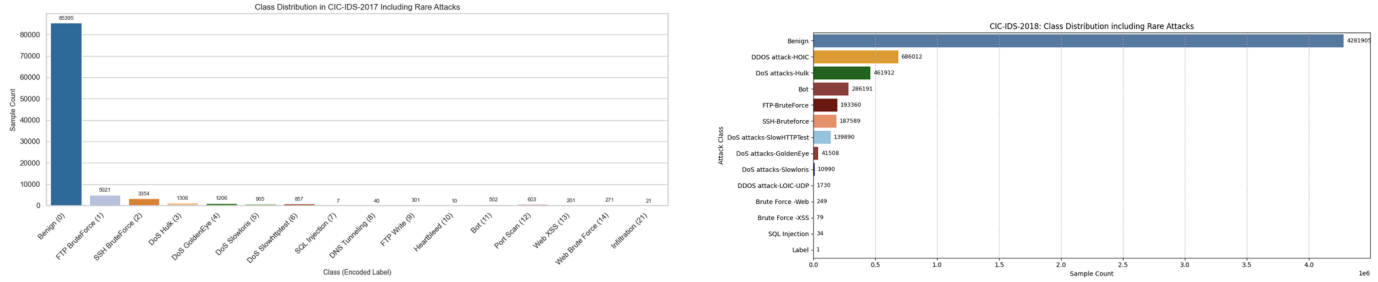
In scenarios where preserving the original class distribution was critical, no balancing techniques were applied to retain the dataset's natural class proportions. This approach was particularly relevant for assessing the model's ability to handle imbalanced data effectively. By maintaining the inherent class distribution, the evaluation provided a realistic perspective on the model's robustness in real-world conditions, where imbalanced datasets are common. To ensure consistent and fair evaluation of the proposed model across various benchmark datasets, several key hyperparameters were carefully selected and applied uniformly across all experiments. These hyperparameters were determined based on best practices and the unique characteristics of each dataset, with minimal adjustments made when necessary to align with specific requirements.

Fig. 5 presents the class distributions across the four benchmark datasets used in this study: CIC-IDS-2017 and CIC-IDS-2018. These distributions reveal substantial class imbalances and the presence of rare attack types, which are critical factors influencing the design and evaluation of intrusion detection models, especially under continual and imbalanced learning settings.

Table 5 shows that each dataset is characterized by the number of classes per task, total tasks, and an average number of samples per task. Common training settings include a sequential (experience-wise) training stream, a minibatch size of 256, evaluation on the full test set after each experience, a fixed memory buffer capacity (1000 or 1500), and the Task-Aware Memory Replay (TAMR) strategy.

5.3. Baselines

We evaluate TAMR against several established continual-learning replay strategies under a unified experimental setup on five intrusion-detection benchmarks: KDDCup'99, NSL-KDD, CIC-IDS-2017, CIC-IDS-2018, and BotIoT-2018. The baselines are Random Memory Replay (RMR), SPIDER [39], Class-Balanced Replay Sampling (CBRS) and ECBRS [8], and Maximally Interfered Retrieval (MIR) [36]. These methods span random, class-balanced, confidence-based, and interference-driven selection strategies; comparing them highlights TAMR's advantages in task-incremental NIDS scenarios.



(a) CIC-IDS-2017 Class Distribution (b) CIC-IDS-2018 Class Distribution

Fig. 5. Normalized class distributions across the CIC-IDS-2017 and CIC-IDS-2018 datasets.

Table 5
Continual learning task configuration across datasets.

Dataset	Classes/Task	Tasks	Samples/Task	Memory Buffer	Iterations per Task
NSL-KDD	5	5	2500	1000	10
KDDCUP'99	5	5	3000	1000	12
CIC-IDS-2017	15	15	2000	1000	8
CIC-IDS-2018	16	16	2100	1000	9
Bot-IoT 2018	9	9	1800	1500	7

Table 6
Comparison of TAMR with related continual learning replay methods.

Method	Memory Efficiency[1]	Computational Complexity[2]	Task Awareness[3]
MIR	Medium	High	Indirect
PAPA	Medium	Medium	Partial
CBRS	Medium	Medium	Partial
ECBRS	Medium	Medium	Partial
TAMR	High	Low-Medium	Direct

Table 7
Comparison of TAMR with representative replay-based continual-learning methods.

Method	Selection strategy	rarity/drifts	Per-sample cost
RMR (Random)	Random	Poor	Very low
MIR	Estimated gradient interference	Limited	High (simulated gradients)
CBRS / ECBRS	Class-balanced / confidence-based	Moderate	Moderate
SPIDER	(method-specific)	Moderate-High	Moderate-High
TAMR (ours)	Task relevance score	**High**	Low-Moderate (no per-candidate gradient sims)

As summarized in Table 6, we compare TAMR with prominent continual learning replay methods across several dimensions. TAMR demonstrates higher memory efficiency by retaining a diverse and representative set of samples per unit memory, while maintaining low to moderate computational complexity during replay sample selection. Unlike MIR and PAPA, which incorporate task information either implicitly or partially, TAMR employs explicit task-awareness through relevance scoring. These distinctions position TAMR as a practical and effective framework for continual learning in network intrusion detection systems.

Additionally, Table 7 summarises key differences. Notably, TAMR reduces per-epoch overhead by avoiding repeated per-candidate gradient simulations (i.e., forward + backward passes to estimate each sample's gradient interference). Instead, relevance ranking is computed from stored metadata and lightweight measures, substantially lowering computational cost. Empirical results in Tables 11 and 12 (Ablation) demonstrate TAMR's improved retention of rare classes, greater stability under drift, and favourable runtime trade-offs.

5.3.1. Experiment settings:

To ensure that our evaluation reflects real-world conditions, we designed a rigorous experimental framework where task separation is based on realistic network traffic patterns and anomaly distributions. Maintaining uniform configurations across all experiments is essential, as it ensures that any performance differences arise from the models themselves rather than variations in hyperparameters. The experimental setup, summarized in Table 8, outlines the key hyperparameters along with the hardware and software specifications. Each model is trained under identical conditions to guarantee a fair comparison.

For continual learning, we employ a New Class (NC) Benchmark, where each dataset is divided into multiple experiences (tasks). Each experience introduces new attack types or network behaviors based on a fixed class order specific to the dataset. This approach ensures that the model learns new classes incrementally, rather than being exposed to all classes at once. The training stream progresses through sequential experiences, with each experience consisting of a subset of classes that the model incrementally learns. Mini-batches of size 256 are used

Table 8
Benchmark setup, hyperparameters, and hardware/software requirements.

Component	Description
Benchmark	NC_benchmark (Config:n_experiences, fixed_class_order, per_exp_classes (for custom class splitting))
Training set	train_set
Testing set	test_set
Dataset	NSL-KDD, KDDCUP'99, CIC-IDS-2017, CIC-IDS-2018, Bot-IoT 2018
Model	SimpleDNN (Input Size and Output Size (Classes) : Varies by Dataset)
Optimizer	Adam
Learning Rate	0.001
Minibatch Size	Training: 256, Evaluation: 300
Epochs per Experience	1 (7-12 iterations per task, dataset-dependent)
Dropout Rate	0.3
Activation Function	ReLU
Loss Function	Cross-Entropy
Regularization	L2 (0.001)
Memory Buffer Size	1000 (default) / 1500 (Bot-IoT 2018)
Early Stopping	Yes
Chip	Apple M3
Memory	16 GB
GPU	10Core
RAM	16 GB
OS	macOS Sequoia
Framework	PyTorch, Avalanche [47]

to balance training efficiency and stability. To combat catastrophic forgetting, the Task-Aware Memory Replay (TAMR) mechanism selectively retrieves and replays task-relevant samples from previous experiences, based on task IDs. A fixed test stream is employed to assess the model's ability to recognize both previously seen and novel attack types. After each training experience, the model is evaluated using a batch size of 300, to determine its effectiveness in retaining past knowledge while adapting to emerging threats.

5.3.2. Evaluation metrics

To comprehensively evaluate the effectiveness of the proposed framework, and following established works [8,11], we adopt several standard evaluation metrics. These metrics are computed from the confusion matrix, which consists of four fundamental components: True Positives (TP), defined as the number of attack instances correctly identified by the model; True Negatives (TN), representing the number of benign instances correctly classified as non-attacks; False Positives (FP), which are benign instances incorrectly classified as attacks (i.e., false alarms); and False Negatives (FN), the number of attack instances incorrectly classified as benign (i.e., missed detections). Using these components, the evaluation metrics are formally defined as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{8}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

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

Precision measures the proportion of detected attacks that are actual attacks, reflecting the model's ability to minimize false alarms. Recall quantifies the model's ability to detect all actual attacks, minimizing missed detections. Accuracy provides the overall correctness of the classification, and F1-score balances precision and recall into a single harmonic mean metric, especially useful when class distributions are imbalanced. Specifically, we employ these metrics in conjunction with the Precision-Recall Area Under the Curve (PR-AUC) to evaluate performance on both benign and attack data. The PR-AUC scores are reported separately for each task, denoted as PR-AUC Task 1, PR-AUC Task 2, and PR-AUC Task 3, respectively. PR-AUC is particularly well-suited for imbalanced datasets, as it emphasizes the trade-off between precision and recall rather than overall accuracy, making it a robust indicator of detection performance, especially for rare attack types. A higher PR-AUC score reflects improved detection capability and a stronger ability to distinguish between attack and benign traffic.

The PR-AUC is computed as

$$\text{PR-AUC} = \int_0^1 P(R)\, dR \tag{11}$$

where $P(R)$ denotes the precision as a function of recall R, and the integral represents the area under the precision-recall curve.

Besides, to evaluate the statistical significance of performance differences between the TAMR and RMR models, a paired t-test was conducted for Task 1 and Task 2. The paired t-test is computed using the following formula:

$$t = \frac{\bar{D}}{s_D / \sqrt{n}} \tag{12}$$

where $\bar{D}$ represents the mean difference between paired observations, s_D denotes the standard deviation of the differences, and n is the number of paired observations.

5.4. Comparative experiments

To assess the effectiveness of the proposed Task-Aware Memory Replay (TAMR) framework, the following subsections provide detailed analyses of task learnability, model stability, and detection performance across multiple benchmark datasets.

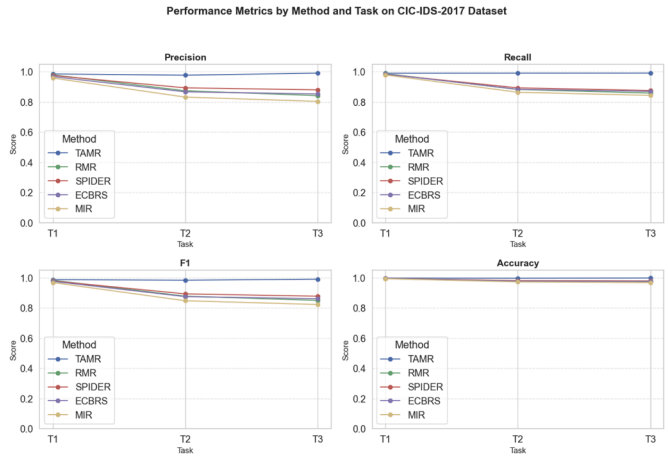
5.4.1. Evaluation on performance metrics by method and task cross benchmarks

Figs. 6a–c illustrate the comparative performance of different methods evaluated on the CIC-IDS-2017, CIC-IDS-2018, and Bot-IoT-2018 datasets, respectively. These results comprehensively present key metrics including False Positive Rate (FPR), Precision, Recall, and Accuracy, segmented by task to highlight the methods' effectiveness in diverse attack scenarios. On the CIC-IDS-2017 dataset (Fig. 6a), our proposed approach consistently demonstrates superior accuracy and recall across the majority of tasks. Notably, the false positive rate remains comparatively low, indicating a robust balance between sensitivity and specificity. Similarly, the results on CIC-IDS-2018 (Fig. 6b) confirm the method's adaptability to newer and more diverse attack vectors, where it outperforms baseline approaches in precision and detection rates. This trend underscores the model's capacity to effectively generalize across evolving threat landscapes.

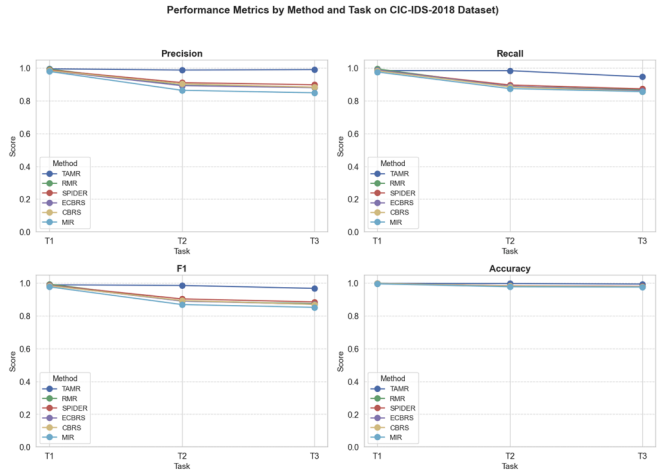
In the Bot-IoT-2018 benchmark (Fig. 6c), which contains highly heterogeneous IoT attack types, the method maintains competitive performance with stable precision and recall, showcasing its suitability for IoT security applications where anomaly patterns are often subtle and diverse. On the whole, the comparative analysis across these benchmarks highlights the proposed method's consistent ability to preserve detection performance while minimizing false alarms across multiple task domains, thereby validating its effectiveness for real-world continual learning in network intrusion detection systems.

5.4.2. Evaluation of task learnability across benchmarks

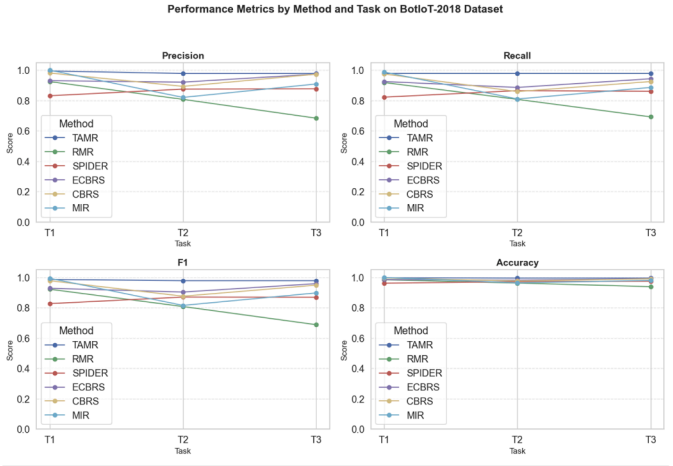
As presented in Table 9, TAMR consistently delivers high and stable PR-AUC scores across Tasks 1–3 on all evaluated datasets, demonstrating strong retention and adaptability under sequential learning conditions. In contrast, RMR experiences significant performance degradation beyond the initial task; for instance, on NSL-KDD, RMR's PR-AUC sharply



(a) CIC-IDS-2017 Dataset



(b) CIC-IDS-2018 Dataset



(c) Bot-IoT-2018 Dataset

Fig. 6. Comparison of performance metrics (FPR, Precision, Recall, Accuracy) by method and task across three benchmark datasets.

declines from 0.9245 (Task 1) to 0.5944 (Task 3), whereas TAMR maintains stability within a narrow range (0.9138–0.9105).

TAMR's advantages are particularly evident in more complex and dynamic scenarios. On CIC-IDS-2018 Task 2, TAMR achieves a PR-AUC of 0.8397, representing a 20.0 % relative improvement over RMR's 0.6994. Similarly, on BotIoT-2018 Task 2, TAMR scores 0.9121 compared to RMR's 0.8234, a gain of 10.8 %. On CIC-IDS-2017 Task 2, TAMR attains 0.9136, surpassing RMR's 0.9018 and MIR's 0.84, underscoring TAMR's capacity to effectively incorporate new information while preserving prior knowledge.

These performance gains stem from TAMR's task-aware selective replay mechanism, which strategically allocates buffer capacity to samples most relevant to each task rather than employing uniform or random sampling. By prioritizing representative and rare instances, TAMR reduces redundancy, mitigates catastrophic forgetting, and enhances both task-specific accuracy and overall generalization.

While the current evaluation validates TAMR's robustness as task complexity increases, future work could explore interpretability analyses such as SHAP, LIME, or feature-attribution methods [26] to identify which samples or features are prioritized at various stages. Such insights would enhance model transparency and potentially inform further refinements to the replay strategy, supporting deployment in dynamic cybersecurity environments.

5.4.3. *Training time comparison*

To evaluate computational efficiency, we compare the training times (in seconds) of different methods across datasets. This analysis highlights the trade-off between model performance and computational cost when using task-aware memory replay Table 10.

TAMR achieves competitive training times while efficiently managing memory updates, reducing redundant replay, and prioritizing new information. Its adaptive memory strategy enables efficient learning of emerging threats without excessive resource use. Additionally, TAMR maintains strong performance across tasks while optimizing memory allocation, making it a robust solution for real-time cybersecurity. These features efficient task-specific memory management and adaptability to evolving threats distinguish TAMR from traditional models in network intrusion detection.

5.5. *Ablation experiments*

This section compares Random Memory Replay (RMR) and Task-Aware Memory Replay (TAMR), two distinct strategies for experience replay in dynamic environments like NIDS. RMR selects past experiences randomly, potentially leading to poor retention of critical but infrequent attack patterns. In contrast, TAMR prioritizes task-relevant samples, ensuring better retention of both common and rare threats. This targeted

Table 9
PR-AUC of various methods on different datasets.

Dataset	TAMR			RMR			SPIDER			ECBRS			CBRS			MIR		
	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3	T1	T2	T3
NSL-KDD	0.9138	0.9134	0.9105	0.9245	0.8215	0.5944	0.934	0.901	0.9311	0.970	0.967	0.9491	0.929	0.969	0.9812	0.923	0.861	0.891
CIC-IDS-2017	0.9035	0.9136	0.9109	0.8298	0.9018	0.7708	0.89	0.901	0.9231	1.0	0.999	0.9182	0.999	0.999	0.9918	0.785	0.84	0.787
CIC-IDS-2018	0.8294	0.8397	0.8236	0.7099	0.6994	0.8189	0.77	0.8012	0.9102	0.999	0.999	0.9013	0.999	0.999	0.982	0.737	0.762	0.87
KDDCUP '99	0.8991	0.9059	0.9033	0.7103	0.8351	0.9034	0.943	0.9245	0.9731	1.0	0.982	0.933	1.0	0.896	0.883	1.0	0.767	0.813
BotIoT-2018	0.9013	0.9121	0.9113	0.9241	0.8234	0.7124	0.812	0.891	0.903	0.931	0.931	0.983	0.98	0.912	0.983	1.0	0.837	0.931

approach enhances model stability, minimizing forgetting while adapting to evolving attack landscapes.

5.5.1. Evaluation of tasks: TAMR vs. RMR

To quantify the benefit of TAMR's task-aware memory buffer under strict memory constraints, we compare its PR-AUC against Random Memory Replay (RMR) on the first two tasks across five benchmark datasets. Tables 11 and 12 summarize these results. On Task 1 (Table 11), TAMR achieves a PR-AUC of 0.9035 on CIC-IDS-2017, representing an absolute gain of 0.0737 (8.9 %) over RMR's 0.8298. Across all datasets, TAMR's Task 1 scores remain tightly clustered (mean = 0.8894, SD = 0.0342), whereas RMR exhibits greater variance with several pronounced drops (e.g., CIC-IDS-2018: 0.8294 vs. 0.7099). However, statistical tests indicate that the performance difference on Task 1 is not significant (e.g., p-value = 0.1549), which is unsurprising given the relative simplicity of attacks in this initial task where baseline methods already perform strongly.

By contrast, TAMR's advantage becomes more pronounced on Task 2 (Table 12). For example, on CIC-IDS-2017, TAMR attains a PR-AUC of 0.9136 compared to RMR's 0.9018, and on CIC-IDS-2018, the improvement is even larger at +0.1403 (0.8397 vs. 0.6994). Overall, TAMR's Task 2 performance (mean = 0.8950, SD = 0.0401) significantly outperforms RMR (mean = 0.7920, SD = 0.0653) by an average of 10.3 % across datasets. These results compellingly demonstrate that TAMR's task-aware selective replay strategy excels at preserving critical and often subtle attack patterns, thereby delivering consistently superior detection accuracy as tasks grow increasingly complex. Although performance on early tasks with simpler attack types is comparable to baseline methods, TAMR's pronounced advantages on subsequent, more challenging tasks underscore its superior adaptability and robustness in handling evolving, real-world intrusion scenarios. This highlights TAMR's significant potential to enhance continual learning frameworks in dynamic cybersecurity environments.

5.6. Statistical significance analysis

To assess the statistical significance of the observed differences, a paired *t*-test was conducted for Tasks 1 and 2 using PR-AUC scores across the five datasets. Table 13 reports the results.

For Task 1, the p-value of 0.1549 exceeds the standard significance threshold of 0.05, indicating that the observed performance difference between TAMR and RMR is not statistically significant. This is expected because Task 1 typically involves simpler and more distinct attack patterns, which both methods can detect effectively without advanced replay strategies. In such initial learning scenarios, baseline methods like RMR are often sufficient, as the model has fewer competing tasks and less complex distributional shifts to manage.

In contrast, the p-value for Task 2 is 0.0176, which is well below the 0.05 threshold, indicating a statistically significant improvement in performance favoring TAMR. The accompanying higher t-statistic and lower variance of the differences suggest that TAMR's advantage is consistent and robust across datasets. This is particularly important because later tasks tend to involve more subtle, complex, or evolving attack types that increase the risk of catastrophic forgetting and require sophisticated memory replay mechanisms.

These findings highlight that TAMR's task-aware selective replay strategy becomes increasingly beneficial as the complexity and volume of incoming data grow. By explicitly prioritizing task-relevant and rare samples, TAMR more effectively preserves critical historical knowledge and adapts to distributional changes, which are common challenges in sequential task learning for network intrusion detection. This validates TAMR's design rationale and its practical significance for deploying adaptive, resilient intrusion detection systems in dynamic real-world networks.

Table 10
Comparison of training times (in seconds) of various methods across different datasets.

Methods	KDDCUP'99	NSL-KDD	CICIDS-2017	CIC-IDS-2018	Bot-IoT2018
CBRS	823	846	630	474	2294
ECBRS	982	964	913	1093	2901
MIR	872	744	749	923	1203
SPIDER	724	978	890	490	2041
RMR	990	748	994	781	2301
TAMR	761	**844**	633	**467**	**1131**

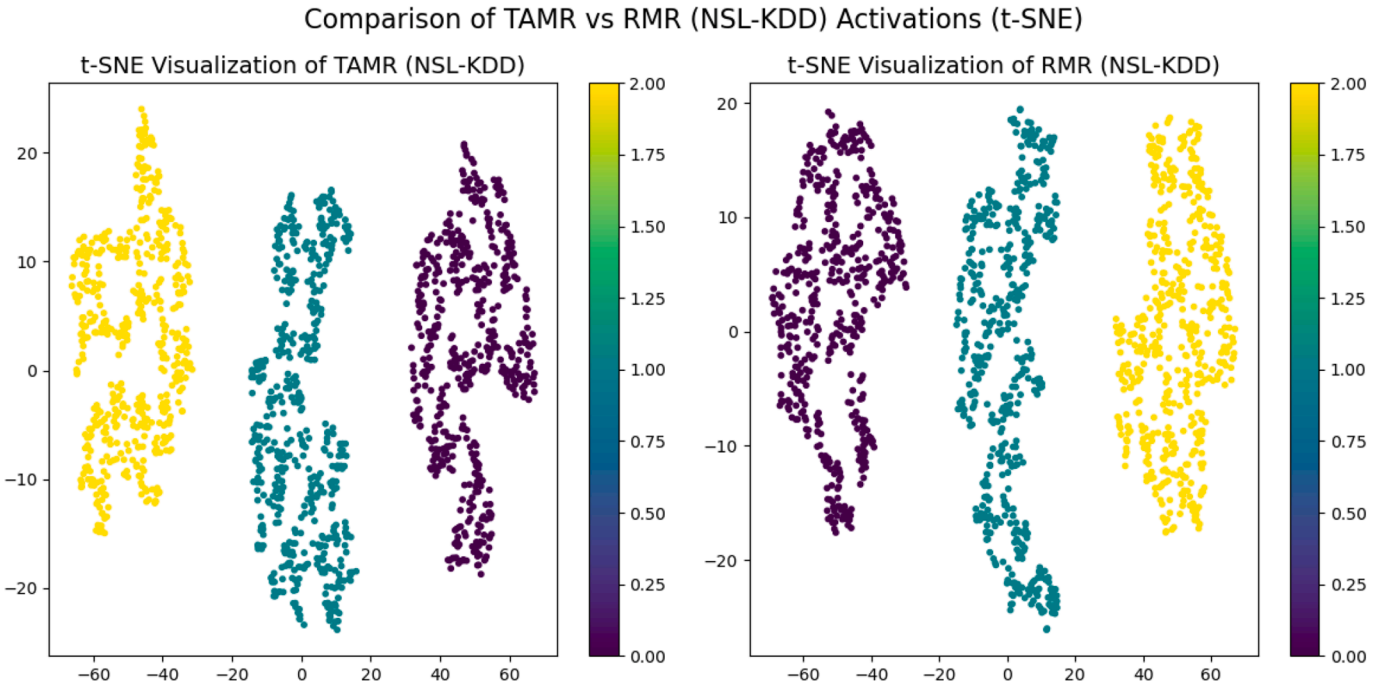


Fig. 7. t-SNE visualization of memory replay and learning on the NSL-KDD dataset.

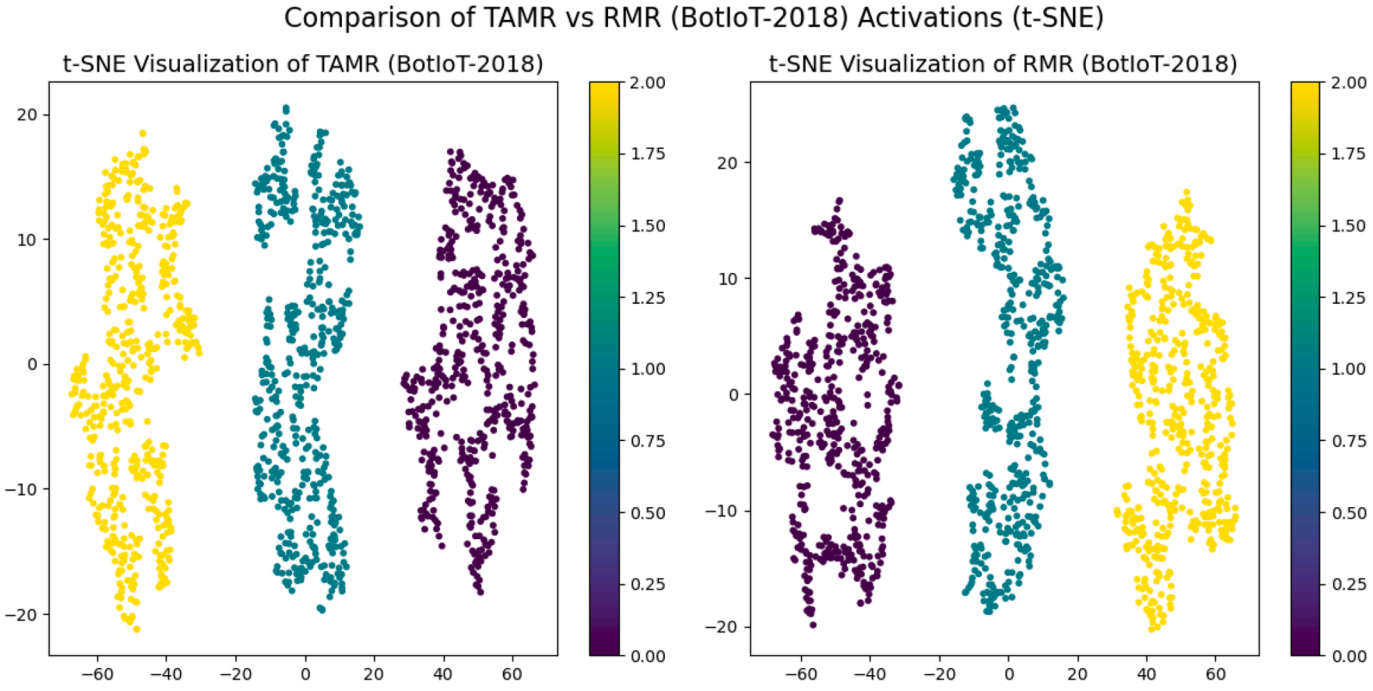


Fig. 8. t-SNE visualization of memory replay and learning on the BotIoT-2018 dataset.

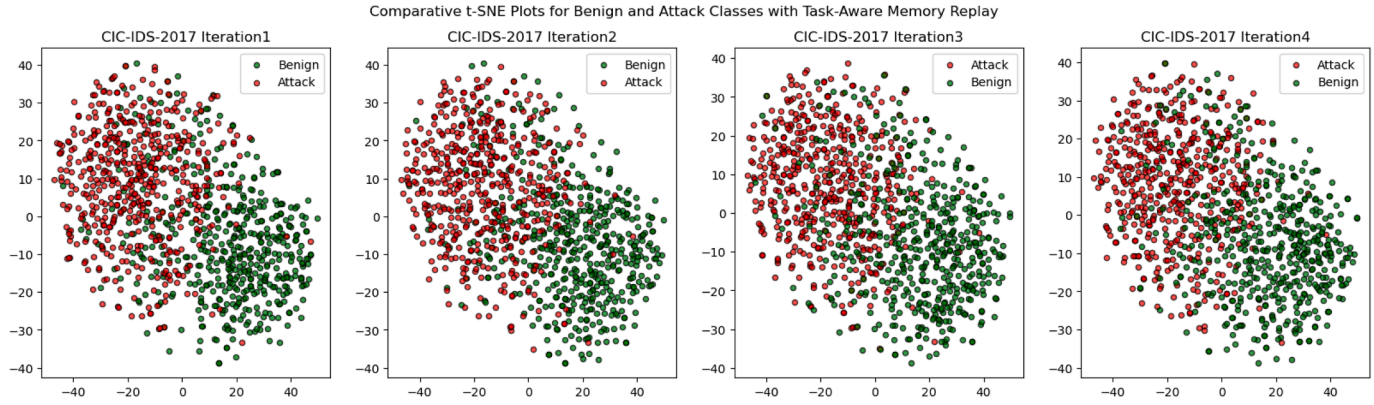


Fig. 9. t-SNE visualization of memory replay and learning on the CIC-IDS-2017 dataset.

Table 11
PR-AUC scores for Task 1 comparing (TAMR) and RMR across datasets.

Dataset	TAMR	RMR
NSL-KDD	0.9138	0.9245
CIC-IDS-2017	0.9035	0.8298
CIC-IDS-2018	0.8294	0.7099
KDDCUP '99	0.8991	0.7103
BotIoT-2018	0.9013	0.9241

Table 12
PR-AUC scores for Task 2 (TAMR) comparing TAMR and RMR across datasets.

Dataset	TAMR	RMR
NSL-KDD	0.9134	0.8215
CIC-IDS-2017	0.9136	0.9018
CIC-IDS-2018	0.8397	0.6994
KDDCUP '99	0.9059	0.8351
BotIoT-2018	0.9121	0.8234

Table 13
Paired t-Test results for Task 1 and Task 2.

Task	t-Statistic	p-Value	Mean Difference	Standard Deviation of Difference
Task 1	1.7507	0.1549	0.0697	0.0890
Task 2	3.8947	0.0176	0.0807	0.0463

5.7. T-SNE visualizations

t-Distributed Stochastic Neighbor Embedding (t-SNE) is employed to visualize the feature representations learned by TAMR and RMR, providing insight into how well each method preserves task-specific information and separates different attack classes in a continual learning context. These visualizations serve as an intuitive means to assess the structural organization of the learned feature space, particularly the model's ability to maintain meaningful decision boundaries over time.

Fig. 7 illustrates the t-SNE projection of the NSL-KDD dataset. The representation learned by TAMR shows well-separated clusters, with benign samples (depicted in green) clearly distinguishable from various attack types (in red). In contrast, RMR exhibits substantial class overlap, indicating diminished class separability and a less effective representation of decision boundaries. This suggests that RMR struggles to maintain task-specific features as new classes are introduced. A similar pattern is observed in the BotIoT-2018 dataset, as shown in Fig. 8. TAMR continues to produce compact, distinct clusters for each class, which reflects strong representational consistency and effective knowledge retention. Conversely, the clusters generated by RMR overlap significantly, particularly among certain attack types, further highlighting its limitations in capturing discriminative features under continual learning conditions. The comparative analysis between the two methods underscores TAMR's ability to mitigate interference between tasks and preserve class-specific structure.

The visualization for the CIC-IDS-2017 dataset, shown in Fig. 9, captures the evolution of feature representations across multiple learning iterations. TAMR maintains stable and well-delineated clusters, even as novel classes are introduced, suggesting a robust capacity for preserving prior knowledge while assimilating new information. Minimal overlap between classes throughout the learning process indicates that TAMR effectively mitigates catastrophic forgetting and sustains task-specific decision boundaries over time.

Collectively, these visualizations demonstrate that TAMR consistently outperforms RMR in preserving the integrity of the learned feature space across diverse datasets. The ability to form well-separated, coherent clusters reflects TAMR's superior capacity for continual representation learning. The distinct positioning of benign and attack samples, coupled with the model's ability to adapt its internal representations in response to novel threats, highlights its effectiveness in maintaining a balance between knowledge stability and plasticity. Furthermore, the task-aware memory replay mechanism enables TAMR to selectively reinforce relevant past experiences, contributing to more efficient learning and reduced interference. These findings support the conclusion that TAMR offers a robust and memory-efficient approach for real-time intrusion detection in dynamic and evolving network environments.

6. Conclusions

This work introduces Task-Aware Memory Replay (TAMR), a novel continual learning approach designed to mitigate catastrophic forgetting and enhance adaptability in NIDS. Traditional NIDS often struggle to retain knowledge of previously learned attack patterns while adapting to evolving cyber threats, primarily due to their static architectures. Although conventional replay-based methods attempt to address this by rehearsing outdated attack patterns and handling class imbalance, they frequently fall short in adaptability and practicality within dynamic network environments.TAMR overcomes these limitations by optimizing memory usage through selective prioritization of past experiences based on their relevance to the current task.

Experimental results on five benchmark NIDS datasets demonstrate that TAMR significantly improves detection accuracy, robustness, and computational efficiency compared to standard replay techniques. By preserving critical historical knowledge while seamlessly incorporating new attack patterns, TAMR effectively reduces catastrophic forgetting. An ablation study further confirms its superiority under both task-aware and random replay scenarios, reinforcing its effectiveness in real-time, dynamic network settings.

However, TAMR assumes access to accurate task labels for memory replay. In practical streaming NIDS applications, explicit task segmentation is often unavailable. To address this, integrating unsupervised or weakly supervised task inference methods, such as clustering feature representations or change point detection can dynamically identify task boundaries, enabling TAMR to adapt without predefined labels. This extension would significantly broaden TAMR's applicability to more realistic, label-sparse environments.

A notable limitation of TAMR is its reduced capability to detect zero-day attacks, which involve previously unknown vulnerabilities. Since TAMR relies mainly on replaying historical task data, it may struggle to recognize novel attack patterns not seen during training. Future research will focus on optimizing TAMR for high-throughput, real-time network traffic, integrating it with zero-trust security architectures, and enhancing resilience against emerging zero-day threats. Additionally, incorporating advanced concept drift detection and improving model interpretability will be prioritized. These enhancements will further strengthen TAMR' s role in addressing the evolving cybersecurity landscape, ensuring robust defense against sophisticated and previously unseen attacks.

CRediT authorship contribution statement

Nasreen Fathima A H: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization, Software; **Ansam Khraisat:** Writing – review & editing, Validation, Supervision, Formal analysis, Investigation, Methodology, Project administration, Conceptualization, Data curation; **Syed Ibrahim S P:** Writing – review & editing, Validation, Supervision, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration; **Gang Li:** Writing – review & editing, Validation, Supervision, Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Project administration.

Data availability

Data will be made available on request.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P. Mishra, V. Varadharajan, U. Tupakula, E.S. Pilli, A detailed investigation and analysis of using machine learning techniques for intrusion detection, IEEE Commun. Surv. Tutorials 21 (1) (2018) 686–728.

[2] Z. Wang, E. Yang, L. Shen, H. Huang, A comprehensive survey of forgetting in deep learning beyond continual learning, IEEE Trans. Pattern Anal. Mach. Intell. 47 (2025) 1464–1483.

[3] M.A. Shyaa, N.F. Ibrahim, Z. Zainol, R. Abdullah, M. Anbar, L. Alzubaidi, Evolving cybersecurity frontiers: a comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems, Eng. Appl. Artif. Intell. 137 (2024) 109143.

[4] A. Alazab, A. Khraisat, S. Singh, S. Bevinakoppa, O.A. Mahdi, Routing attacks detection in 6lowpan-based internet of things, Electronics 12 (6) (2023) 1320.

[5] Y. Chai, L. Du, J. Qiu, L. Yin, Z. Tian, Dynamic prototype network based on sample adaptation for few-shot malware detection, IEEE Trans. Knowl. Data Eng. 35 (5) (2022) 4754–4766.

[6] M. Kim, D. Lee, K. Lee, D. Kim, S. Lee, J. Kim, Deep sequence models for packet stream analysis and early decisions, in: 2022 IEEE 47th Conference on Local Computer Networks (LCN), IEEE, 2022, pp. 56–63.

[7] D. Rolnick, A. Ahuja, J. Schwarz, T. Lillicrap, G. Wayne, Experience replay for continual learning, Adv. Neural Inf. Process. Syst. 32 (2019) 348–358 .

[8] S. Channappayya, B.R. Tamma, et al., Augmented memory replay-based continual learning approaches for network intrusion detection, Adv. Neural Inf. Process Syst. 36 (2023) 17156–17169.

[9] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: defying forgetting in classification tasks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3366–3385.

[10] J. Guo, P. Schwaller, Augmented memory: Capitalizing on experience replay to accelerate de novo molecular design, arXiv preprint arXiv:2305.16160 (2023).

[11] S.K. Amalapuram, A. Tadwai, R. Vinta, S.S. Channappayya, B.R. Tamma, Continual learning for anomaly based network intrusion detection, in: 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, 2022, pp. 497–505.

[12] J.S. Smith, L. Valkov, S. Halbe, V. Gutta, R. Feris, Z. Kira, L. Karlinsky, Adaptive memory replay for continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 3605–3615.

[13] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: techniques, datasets and challenges, Cybersecurity 2 (1) (2019) 1–22.

[14] A. Ashfahani, M. Pratama, Autonomous deep learning: continual learning approach for dynamic environments, in: Proceedings of the 2019 SIAM international conference on data mining, SIAM, 2019, pp. 666–674.

[15] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. (CSUR) 41 (3) (2009) 1–58.

[16] A. Khraisat, A. Alazab, A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges, Cybersecurity 4 (1) (2021) 18.

[17] A.H.N. Fathima, S.P.S. Ibrahim, A. Khraisat, Enhancing network traffic anomaly detection: leveraging temporal correlation index in a hybrid framework, IEEE Access (2024).

[18] R.K. Malaiya, D. Kwon, S.C. Suh, H. Kim, I. Kim, J. Kim, An empirical evaluation of deep learning for network anomaly detection, IEEE Access 7 (2019) 140806–140817.

[19] M. Al-Imran, S.H. Ripon, Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models, Int. J. Comput. Intell. Syst. 14 (1) (2021) 200.

[20] A.H.N. Fathima, S.P.S. Ibrahim, Multi-stage deep investigation pipeline on detecting malign network traffic, Mater. Today Proc. 62 (2022) 4726–4731.

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A.A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al., Overcoming catastrophic forgetting in neural networks, Proc. Natl. Acad. Sci. 114 (13) (2017) 3521–3526.

[22] C. Tam, L.F. Capretz, Investigating continual learning strategies in neural networks, in: IECON 2023-49th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2023, pp. 1–7.

[23] S. Prasath, K. Sethi, D. Mohanty, P. Bera, S.R. Samantaray, Analysis of continual learning models for intrusion detection system, IEEE Access 10 (2022) 121444–121464.

[24] G.M. Van de Ven, A.S. Tolias, Three scenarios for continual learning, arXiv preprint arXiv:1904.07734 (2019).

[25] G.I. Parisi, R. Kemker, J.L. Part, C. Kanan, S. Wermter, Continual lifelong learning with neural networks: a review, Neural Netw. 113 (2019) 54–71.

[26] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, J. Jang, Online class-incremental continual learning with adversarial shapley value, in: Proceedings of the AAAI Conference on Artificial Intelligence, 35, 2021, pp. 9630–9638.

[27] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, X.-M. Wu, Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, Adv. Neural Inf. Process. Syst. 34 (2021) 6747–6761.

[28] J. Talpini, F. Sartori, M. Savi, Hierarchical Multiclass Continual Learning for Network Intrusion Detection, in: 2024 IEEE 10th International Conference on Network Softwarization (NetSoft), IEEE, 2024, pp. 263–267.

[29] X. Zhang, R. Zhao, Z. Jiang, H. Chen, Y. Ding, E.C.H. Ngai, S.-H. Yang, Continual learning with strategic selection and forgetting for network intrusion detection, in: IEEE INFOCOM 2025-IEEE Conference on Computer Communications, IEEE, 2025, pp. 1–10.

[30] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, C.H. Lampert, Icarl: Incremental classifier and representation learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2001–2010.

[31] F. Zenke, B. Poole, S. Ganguli, Continual learning through synaptic intelligence, in: International Conference on Machine Learning, PMLR, 2017, pp. 3987–3995.

[32] L. Collins, C.-T. Li, Y. Hu, B. Yan, Elastic Weight Consolidation and Transfer Learning: A New Paradigm for Deepfake Detection, in: 2024 IEEE International Conference on Imaging Systems and Techniques (IST), IEEE, 2024, pp. 1–6.

[33] Z. Li, D. Hoiem, Learning without forgetting, IEEE Trans. Pattern Anal. Mach. Intell. 40 (12) (2017) 2935–2947.

[34] G. Tyndall, K. Arizah, D. Tanaya, A. Purwarianti, D.P. Lestari, S. Sakti, Continual learning in machine speech chain using gradient episodic memory, in: 2024 27th Conference of the Oriental COCOSDA International Committee for the Coordination and Standardisation of Speech Databases and Assessment Techniques (O-COCOSDA), IEEE, 2024, pp. 1–6.

[35] L. Pellegrini, G. Graffieti, V. Lomonaco, D. Maltoni, Latent replay for real-time continual learning, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 10203–10209.

[36] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, L. Page-Caccia, Online continual learning with maximal interfered retrieval, Adv. Neural Inf. Process. Syst. 32, (2019), 11849–11860.

[37] T. Adel, H. Zhao, R.E. Turner, Continual learning with adaptive weights (claw), arXiv preprint arXiv:1911.09514 (2019).

[38] S.K. Amalapuram, S. Kumar, B.R. Tamma, S. Channappayya, SOUL: A Semi-supervised Open-world continUal Learning method for Network Intrusion Detection, arXiv preprint arXiv:2412.00911 (2024).

[39] S.K. Amalapuram, B.R. Tamma, S.S. Channappayya, Spider: A semi-supervised continual learning-based network intrusion detection system, in: IEEE INFOCOM 2024-IEEE Conference on Computer Communications, IEEE, 2024, pp. 571–580.

[40] H.-J. Liao, C.-H.R. Lin, Y.-C. Lin, K.-Y. Tung, Intrusion detection system: a comprehensive review, J. Netw. Comput. Appl. 36 (1) (2013) 16–24.
[41] Z. Zhang, Y. Zhang, D. Guo, M. Song, A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks, Int. J. Mach. Learn. Cybern. 12 (6) (2021) 1649–1665.
[42] G. Andresini, F. Pendlebury, F. Pierazzi, C. Loglisci, A. Appice, L. Cavallaro, Insomnia: towards concept-drift robustness in network intrusion detection, in: Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security, 2021, pp. 111–122.
[43] M. Soltani, K. Khajavi, M. Jafari Siavoshani, A.H. Jahangir, A multi-agent adaptive deep learning framework for online intrusion detection, Cybersecurity 7 (1) (2024) 9.
[44] F.L. de Caldas Filho, S.C.M. Soares, E. Oroski, R. de Oliveira Albuquerque, R.Z.A. Da Mata, F.L.L. De Mendonça, R.T. de Sousa Júnior, Botnet detection and mitigation model for IoT networks using federated learning, Sensors 23 (14) (2023) 6305.
[45] M.S. Alshehri, O. Saidani, F.S. Alrayes, S.F. Abbasi, J. Ahmad, A self-attention-based deep convolutional neural networks for IIot networks intrusion detection, IEEE Access 12 (2024) 45762–45772.
[46] S. Shah, P.S. Muhuri, X. Yuan, K. Roy, P. Chatterjee, Implementing a network intrusion detection system using semi-supervised support vector machine and random forest, in: Proceedings of the 2021 ACM Southeast Conference, 2021, pp. 180–184.
[47] V. Lomonaco, L. Pellegrini, A. Cossu, A. Carta, G. Graffieti, T.L. Hayes, M. De Lange, M. Masana, J. Pomponi, G.M. Van de Ven, et al., Avalanche: an end-to-end library for continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3600–3610.