SA-SOINN: A Self-Adaptive Neural Network for Continuous Intrusion Detection in Dynamic Environments

Dennis Glenn Ejeh[1,2,*], Gian Luca Foresti[2,†], Marino Miculan[2,†] and Axel De Nardin[2,†]

[1]CYSEC, IMT School for Advanced Studies Lucca, Italy
[2]Department of Mathematics, Computer Science, and Physics, University of Udine, Italy

Abstract

Intrusion detection in modern networks remains challenging due to constantly evolving traffic patterns and sophisticated attacks. Conventional intrusion detection systems (IDS), which rely on static models and periodic retraining, struggle to maintain accuracy under non-stationary data distributions. This paper introduces the Self-Adaptive Self-Organizing Incremental Neural Network (SA-SOINN), a continuous learning framework for intrusion detection. SA-SOINN extends the SOINN architecture with a variance-aware controller that profiles incoming data streams in real time and dynamically switches between unsupervised and semi-supervised learning modes. This design enables the model to autonomously adapt its structure and parameters while maintaining a balance between structural stability and adaptive responsiveness. Evaluated on the NSL-KDD and CIC-IDS-2017 datasets in a single-pass learning setting without retraining, SA-SOINN achieves balanced precision and recall, outperforming conventional supervised and unsupervised baselines. Furthermore, its CPU-aware multithreaded design ensures computational efficiency, underscoring its suitability for scalable, real-time anomaly detection in evolving network infrastructures.

Keywords

Cybersecurity, intrusion detection systems, continuous learning, self-adaptive neural networks, concept drift, SA-SOINN

1. Introduction

Modern networks continuously evolve in terms of traffic patterns, attack surfaces, and underlying data distributions. Conventional IDS [1, 2], which are typically trained offline, struggle to accommodate such dynamism and therefore often exhibit performance degradation as network characteristics change over time, including under concept drift conditions [3]. Incremental learning paradigms, particularly self-organizing networks [4], offer a promising alternative by enabling continuous learning without the need for full retraining.

Supervised learning (SL) models typically assume stationary data distributions and require frequent retraining to maintain accuracy [2, 5]. In real-world environments, both benign traffic and malicious behaviors change over time, making repeated retraining computationally demanding and operationally impractical for systems requiring real-time performance [5]. Moreover, supervised methods are likely to overfit transient patterns, which are short-lived changes in network traffic caused by workload spikes, software updates, or temporary anomalies. This, in turn, leads to false positives and unstable decision boundaries [6].

Although semi-supervised learning (SSL) aims to bridge the gap between unsupervised discovery and supervised refinement, excessive adaptability—for example, aggressive transitions from unsupervised to supervised modes—can cause overfitting and reduced generalization, especially in the presence of noisy and rapidly changing data [7, 8]. In contrast, unsupervised learning (UL) and incremental learning paradigms have gained traction due to their ability to learn autonomously from evolving data streams

Joint National Conference on Cybersecurity (ITASEC & SERICS 2026), February 09-13, 2026, Cagliari, IT
*Corresponding author.
†These authors contributed equally.
dennis.ejeh@imtlucca.it (D. G. Ejeh); gianluca.foresti@uniud.it (G. L. Foresti); marino.miculan@uniud.it (M. Miculan); axel.denardin@uniud.it (A. De Nardin)
0009-0005-5445-0503 (D. G. Ejeh); 0000-0002-8425-6892 (G. L. Foresti); 0000-0003-0755-3444 (M. Miculan); 0000-0002-0762-708X (A. De Nardin)

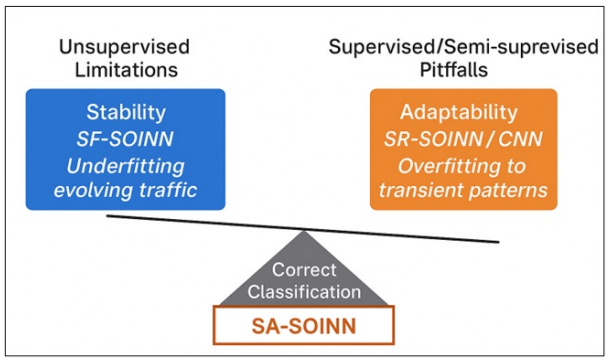

Figure 1: Trade-off between Stability and Adaptability in IDS: SA-SOINN Balancing Role

without explicit retraining, making them naturally suitable for dynamic and non-stationary environments [9]. However, they face the opposite challenge: excessive stability can result in underfitting and poor response to genuinely novel or emerging attack patterns, as the model prioritizes long-term consistency over short-term responsiveness [3]. As illustrated in Figure 1, each model class captures one side of a fundamental trade-off between stability and adaptability. Thus, achieving balance remains a fundamental challenge in the design of robust intrusion detection models—a core challenge addressed by the SA-SOINN framework.

1.1. Background Motivation

We observed that our initial model, SF-SOINN (Soft-Forgetting SOINN) [10], exhibited a noticeable performance degradation on low-variance structured datasets [11]. This behavior highlighted a deeper, more fundamental trade-off within machine learning architectures. Unsupervised methods such as SF-SOINN excel at discovering natural clusters and maintaining robustness to noise in complex, high-variance datasets [12]. However, they often lack the precision needed to establish clear decision boundaries that separate low-variance structured data patterns into well-defined groups. This limitation stems from SF-SOINN's reliance on a fixed similarity threshold (an inverse Euclidean distance of 0.7) for node creation. Although this threshold is well suited to complex data distributions, it is suboptimal for simpler, more structured datasets.

To address this limitation, an intermediate prototype variant—SR-SOINN (Self-Regulating SOINN) [13]—was developed to incorporate semi-supervised learning capabilities, introducing an adaptive similarity threshold (typically below 0.5) and with more aggressive node pruning to enhance precision on low-variance structured datasets. These changes allow SR-SOINN to reach classification accuracy comparable to supervised methods, but at the cost of increased sensitivity to labeling noise and transient traffic fluctuations, which ultimately leads to overfitting and reduced generalization.

The stability that makes unsupervised models robust also limits their responsiveness, while excessive adaptability in semi-supervised approaches—when they lean toward SL behavior—causes them to inherit the principal weakness of SL: fragility under non-stationary conditions [14]. This limitation is not specific to any one model family; rather, it reflects a broader challenge across machine learning (ML) paradigms operating in dynamic environments. It is well established in both ML and IDS research that SL models—such as convolutional neural networks (CNNs)—struggle with concept drift due to their dependence on stationary data distributions and the need for repeated retraining to remain effective [5, 6, 15]. The quest to solve this problem led to the development of SA-SOINN –a self-adaptive neural network that combined the stability of SF-SOINN with the adaptability of SR-SOINN by dynamically tuning its SOINN engine (parameters) during training to optimize performance for the characteristics of the targeted dataset.

1.2. Contributions

We identified the following key contributions for this research:

- **Self-Adaptive Neural Networks (SANNs):** A SANN is an AI system that can automatically adjust its structure or parameters during training to improve performance, for example for function approximation or solving complex equations. Such networks can grow their architecture (e.g., by adding neurons or layers) and adapt their learning process by focusing more on challenging parts of the data, leading to more efficient and accurate results without manual intervention. Our SA-SOINN framework is situated within the broader paradigm of SANNs: it can learn incrementally, adjust its soft-forgetting approach, switch learning modes, and modify pruning rates dynamically. Although previous SANN research has focused mainly on function approximation [16] and time-series prediction [17], their application to continuous learning in non-stationary environments for IDS remains underexplored.
- **Generalized Trade-off Solutions:** The proposed framework addresses the stability–adaptability dilemma in IDS through data-driven learning-mode selection, enabling robust performance under evolving data conditions [15].
- **Variance-Aware Learning Mechanism:** We introduce a variance-aware controller that dynamically switches between unsupervised and semi-supervised learning modes within a single incremental SOINN framework. This mechanism adaptively balances stability and precision without relying on external drift detectors or hybrid model architectures.
- **Empirical Validation:** SA-SOINN achieves superior precision on complex data (99.9% on CIC-IDS-2017, 72.8% on NSL-KDD) in single-pass learning without retraining, outperforming the baselines SF-SOINN, SR-SOINN, and CNN.
- **Efficiency Implementation:** A multithreaded, hardware-aware implementation mitigates computational overhead, confirming the practicality of SA-SOINN framework for real-time deployment.
- **Open-Source Code:** We provide a publicly available, optimized Python implementation of SA-SOINN on Kaggle to foster reproducibility and further research in adaptive incremental learning paradigms.

The remainder of the paper is structured as follows: Section 2 discusses related work. Section 3 details the SA-SOINN framework. Section 4 presents the experimental evaluation, and Section 5 concludes the paper.

2. Related Work

This review traces the research trajectory from static supervised models to adaptive incremental learning, culminating in the specific stability-adaptability trade-off that our framework is designed to solve.

2.1. The Limitations of Supervised Learning in Dynamic Environments

Supervised machine learning models, including deep neural networks, long short-term memory (LSTM) networks, and CNN architectures, have long been the cornerstone of academic intrusion detection research [2]. These systems achieve high accuracy in controlled static benchmarks by learning from fixed, labeled datasets. However, their fundamental assumption of stationary data distributions is violated in real-world network environments [3].

This leads to two critical weaknesses. First, they require frequent, computationally expensive retraining to maintain accuracy against evolving attack patterns, which is operationally impractical for real-time systems [5]. Second, they are prone to overfitting transient anomalies—short-lived traffic fluctuations caused by legitimate network events—leading to false positives and a loss of generalization [6]. As Xu *et al.* [18] note, supervised models struggle to adapt to new attack types without explicit retraining on new labeled data. Although they offer precision, their inherent lack of continuous adaptability makes them unsuitable for long-term deployment in dynamic environments.

2.2. Unsupervised and Incremental Learning as a Promising Alternative

In response to these limitations, unsupervised and incremental learning paradigms have gained significant traction in IDS [19]. These methods learn from data streams in a single pass, often without labels, making them naturally suited to non-stationary environments. Techniques range from clustering and autoencoders to more sophisticated streaming algorithms.

A key advantage is their inherent adaptability. For example, Mirsky *et al.* [9] proposed Kitsune, an ensemble of autoencoders that learns normal traffic patterns incrementally. Similarly, online random forests and other drift-aware learners offer mechanisms for model updating without full retraining [20]. However, these approaches often introduce a new set of challenges: they can be sensitive to noise, require careful parameter tuning, and may prioritize stability to such a degree that they underfit genuinely novel or emerging attack patterns.

2.3. The Central Challenge: Concept Drift and Feature Dynamics

The core problem with which streaming and incremental IDS must contend is concept drift—the change in the underlying data distributions over time [3]. This is not a minor nuisance, but a fundamental characteristic of network traffic, driven by evolving software, new protocols, and changing attacker strategies.

Recent surveys, such as the one by Gunasekara et al. [14] on online streaming continual learning, systematically categorize drift types and adaptation strategies. This establishes the non-negotiable requirement for IDS models to be fundamentally adaptive.

Furthermore, the work by Lukats et al. [21] provides a comprehensive benchmark and survey of fully unsupervised concept drift detectors, evaluating their performance, and discussing evaluation challenges in streaming settings. These studies establish how drift and distribution shifts degrade model performance and justify frameworks that monitor and adapt to these changes.

Concept drift may be manifested as changes in data distributions, as well as shifts in statistical properties such as variance. Supervised models are particularly sensitive to distribution shifts due to their reliance on stationary training assumptions and fixed decision boundaries [5, 6, 15]. In contrast, incremental unsupervised models can continue updating their internal representations after a single-pass training on a balanced distribution. However, our observations indicate that their clustering performance can be significantly affected by substantial changes in data variance—especially transitions between high-variance and low-variance regimes.

For this reason, SA-SOINN adopts a variance-aware controller instead of an explicit drift-detection module. Unlike hybrid frameworks that combine supervised and unsupervised models—sometimes augmented with external drift detectors or memory buffers—SA-SOINN operates as a single self-adaptive neural network (SANN). It extends an unsupervised incremental topology-learning model with the ability to switch to semi-supervised behavior when appropriate, without introducing additional external supervised components.

Although SA-SOINN is designed to respond to evolving data conditions in a manner suitable for incremental neural networks, we do not claim that it fully resolves concept drift at this stage. The objective of the present work in the development stage is to validate the architectural behavior and the learning mechanism of SA-SOINN on benchmark datasets. Comprehensive time-ordered streaming evaluations and systematic comparisons with established drift-aware methods are left to future work during the deployment stage, where their effectiveness under real drift scenarios will be rigorously assessed.

2.4. The SOINN Family and the Stability–Adaptability Trade-off

The Self-Organizing Incremental Neural Network (SOINN) family represents a dedicated effort to build models that are intrinsically incremental and topology-aware. The original SOINN algorithm by Shen and Hasegawa [4] constructs a graph of prototypes incrementally, making it a natural fit for streaming data. Subsequent variants have focused on enhancing its robustness. SOINN+ [12] improved

noise resistance, while the enhanced three-way attribute SOINN (T-SOINN) was proposed to support unsupervised learning through attribute reduction [22].

The most direct precursor to our work is SF-SOINN (Soft-Forgetting SOINN) by Rinaldo and Foresti [10], which introduced a soft-forgetting mechanism to gracefully remove obsolete knowledge. SF-SOINN demonstrated strong performance in high-variance data, embodying the stability cherished by unsupervised learning. However, its fixed parameters limit its precision to low-variance data. Our intermediate variant, SR-SOINN (Self-Regulating SOINN) [13], introduced adaptive thresholding and semi-supervision to achieve the adaptability and precision of supervised methods, but at the cost of becoming fragile to noise and transient patterns.

This evolution within the SOINN family concretely illustrates a fundamental trade-off that extends to machine learning at large: the stability that ensures robustness inherently limits adaptability, and the adaptability that enables precision inherently introduces fragility [14].

2.5. SR-SOINN (Semi-Supervised Learning)

As a semi-supervised intermediate variant developed after SF-SOINN [10] and before our proposed SA-SOINN, the Self-Regulating SOINN (SR-SOINN) [13] was introduced to address the limitations of fixed thresholds and static learning rates on low-variance datasets. While retaining the incremental, topology-aware nature of SOINN, SR-SOINN incorporates several enhancements aimed at improving classification precision and model responsiveness:

- **Adaptive similarity thresholding:** SR-SOINN dynamically computes its similarity threshold based on local data density, in contrast to SF-SOINN, which uses a static threshold. The threshold is defined from the evolving input-space density as

$$\mathrm{thr}(t) = \mu_d(t) + k\,\sigma_d(t), \tag{1}$$

  where $\mu_d(t)$ is the rolling mean of the distances between nodes, $\sigma_d(t)$ is their standard deviation, and k is a tunable sensitivity parameter. This mechanism allows for flexible threshold tightening or relaxation in direct response to local data dispersion levels, enabling finer discrimination on structured datasets.
- **Entropy-based node pruning.** Nodes with persistently low utility or high class entropy are periodically pruned to control network size and maintain structural efficiency, which reduces redundancy and enhances model interpretability. The entropy is calculated as

$$H_n = -\sum_i p_i \log p_i, \tag{2}$$

  where p_i is the empirical label distribution associated with the node.
- **Confidence-weighted prediction.** Each node contributes a label confidence score to guide learning, inversely proportional to its normalized distance:

$$\mathrm{conf} = \exp\left(-\frac{d}{\bar{d}_{\mathrm{nbr}}}\right), \tag{3}$$

  where d is the distance to the node and $\bar{d}_{\mathrm{nbr}}$ is the mean neighbor distance.
- **Adaptive learning rates.** Learning rates are inversely proportional to node visitation frequency, preserving plasticity for new patterns while stabilizing mature clusters. They are defined as

$$\mathrm{lr}_w = \frac{\text{base\_lr}}{1 + \frac{\text{winner\_visits}}{100}}, \tag{4}$$

$$\mathrm{lr}_n = \frac{0.3\,\text{base\_lr}}{1 + \frac{\text{neighbor\_visits}}{100}}, \tag{5}$$

  for the winner node and its neighbors, respectively.

- **Noise suppression and reclassification.** Isolated or short-lived nodes are automatically tagged as noise clusters and excluded from active inference, reducing false positives.

Although these mechanisms enable SR-SOINN to outperform SF-SOINN on structured datasets such as NSL-KDD, its semi-supervised behavior increases sensitivity to labeling noise and transient patterns, limiting its generalization on high-variance or noisy datasets such as CIC-IDS-2017. The algorithm's ability to self-regulate its learning parameters leads to improved precision and reduced overfitting under low variability conditions. However, this adaptability comes at the cost of stability, which motivates the development of SA-SOINN, where a variance-aware controller is introduced to dynamically balance stability and adaptability across diverse data conditions.

2.6. Positioning SA-SOINN to Bridge the Gap

Recent research has recognized the need to balance these competing demands, leading to proposals for hybrid and continuous learning systems.

Zhang et al. [23] proposed SSF (Strategic Selection and Forgetting), a continuous learning method for IDS that strategically manages memory buffers and forgets outdated knowledge to adapt to evolving threats. Another work explored combining unsupervised and supervised modules [24]. However, these approaches often require manual intervention, pre-defined drift detectors, or complex memory buffers. Doroud et al. [25] propose L-IDS, a lifelong-learning anomaly-detection system with adaptation capability in changing network environments. A very recent incremental deep-learning-based method, CTWA [26], shows promising results for a high-variance IoT dataset.

These works signal the trend toward adaptive continuous-learning systems, often combining unsupervised and supervised modules, dynamic memory or thresholding, and stream-based updating.

The concept of self-adaptive neural networks (SANNs) is well established in the machine learning literature, with significant research focused on optimizing network architectures [27] and adapting to non-stationary environments for tasks like time-series forecasting [17]. For example, Cai et al. [16] proposed a self-adaptive neural network to solve partial differential equations through dynamic architecture growth, while Stolzenburg et al. [17] demonstrated architecture learning in linear recurrent networks for time-series prediction. However, these works do not address the unique demands of intrusion detection systems, such as real-time responsiveness and concept drift handling.

Semi-supervised learning (SSL) has also gained attention in adaptive IDS due to its ability to leverage both labeled and unlabeled data. However, it suffers from notable challenges under dynamic conditions. For example, Chen et al. [7] show that pseudo-labeling in SSL can lead to confirmation bias and error accumulation, severely hurting generalization when pseudo-labels are noisy. Cascante-Bonilla et al. [8] similarly argue that pseudo-labeling mechanisms must be carefully controlled—for example, by using curriculum strategies and model restarts—to handle concept drift and out-of-distribution (OOD) unlabeled data. These findings emphasize that SSL, despite offering adaptability, can inherit the fragility of supervised models if pseudo-labeling is not carefully managed, further motivating our mode-switching strategy in SA-SOINN.

Within the specific domain of network intrusion detection, recent efforts have increasingly explored adaptive learning strategies. Qu et al. [28] proposed a self-adaptive discriminative autoencoder for intrusion detection using deep metric learning, while Hossain [29] introduced a reinforcement learning-based IDS that adapts to evolving attack patterns using Deep Q-Networks. Despite these advances, the implementation of a SANN that autonomously manages the fundamental stability–plasticity trade-off—by dynamically switching its core learning paradigm between unsupervised and semi-supervised modes based on data variance—represents a novel contribution in the field of adaptive intrusion detection.

SA-SOINN addresses this gap by introducing a variance-aware controller that enables context-aware adaptation under evolving traffic conditions. Unlike prior systems that combine fixed models in hybrid or parallel configurations, SA-SOINN provides a unified and efficient framework capable of adjusting learning behavior across diverse data regimes within a single-pass online learning paradigm.

3. Methodology

3.1. SA-SOINN Framework

The Self-Adaptive Self-Organizing Incremental Neural Network (SA-SOINN) [30] is a self-adaptive continuous learning framework that autonomously transitions between unsupervised and semi-supervised learning modes based on real-time profiling of input data variance. This capability makes it suitable for dynamic environments where traffic characteristics evolve over time [18]. The framework was designed to synthesize the stability of SF-SOINN with the adaptability of SR-SOINN, providing an integrated mechanism for context-aware learning.

The high-level architecture of the SA-SOINN framework is illustrated in Figure 2 and summarized in Algorithm 1. During operation, SA-SOINN automatically profiles incoming data streams to estimate statistical variance and structural characteristics. Based on these properties, it dynamically switches between stable and adaptive operational modes without manual intervention. This enables the model to maintain reliable behavior when traffic patterns are consistent while rapidly adapting when shifting behaviors emerge.

The variance-aware controller selects the learning mode and adjusts threshold scaling and pruning sensitivity, reinforcing stable behavior on low-variance environments and increasing flexible behavior in high-variance environments. This mechanism allows SA-SOINN to preserve previously learned patterns while adapting to evolving traffic characteristics during continuous operation.

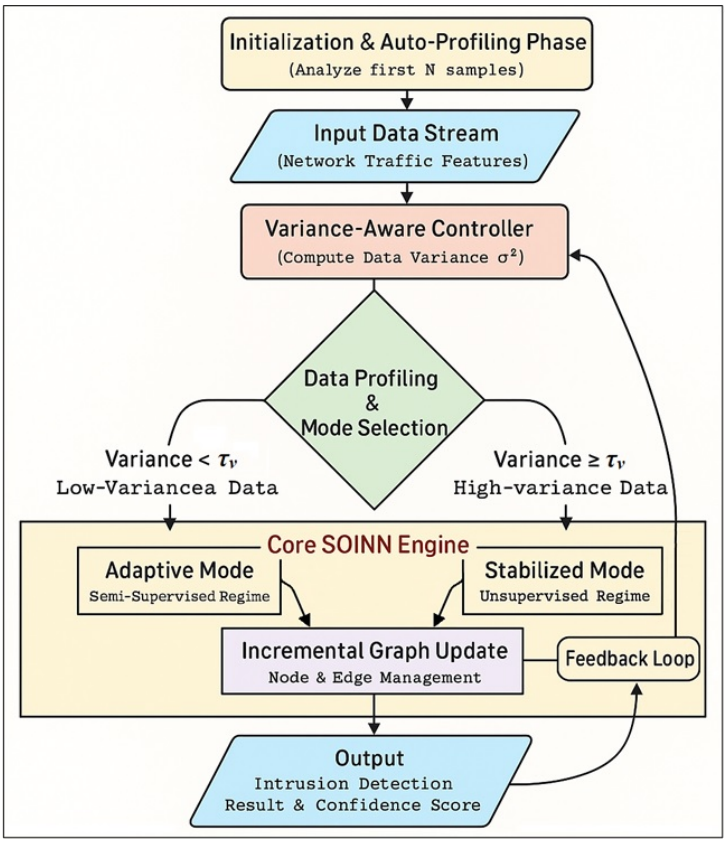

Figure 2: High-Level Architecture of the SA-SOINN Framework

The Variance is computed as the aggregate variance of the distances over a rolling window of size N. The window size N is fixed and empirically chosen to balance the stability and responsiveness of the profiling; but a detailed sensitivity analysis of N is left for future work.

During training, SA-SOINN learns the node prototypes, their class-label distributions, edge ages and visitation frequencies, as well as the mode-dependent decision thresholds $\mathrm{thr}(t)$ derived from rolling statistics $\mu_d(t)$ and $\sigma_d(t)$. In contrast, window size N, variance threshold τ_v, base learning rates,

Algorithm 1 : SA-SOINN

1. **Initialization (auto-profiling).**
   - Initialize a rolling buffer B of size N to store nearest-prototype distances.
   - For the first N samples, compute the nearest-prototype distance d_t and append it to B.
   - Compute rolling statistics: mean $\mu_d(t)$, standard deviation $\sigma_d(t)$, median distance, and (if semi-supervised) the label distribution.
   - Based on these statistics, set the initial mode:
     - if $\sigma_d(t) < \tau_v$, use **Adaptive Mode** (low-variance data);
     - else, use **Stabilized Mode** (high-variance data).
2. **Streaming update.**
   - For each incoming sample, compute its nearest-prototype distance d_t.
   - Update buffer B and recompute $\mu_d(t)$ and $\sigma_d(t)$.
3. **Variance-based mode adaptation.**
   - *Adaptive Mode (low-variance data):*
     - set dynamic threshold $\mathrm{thr}(t) = \mu_d(t) + 1.5\,\sigma_d(t)$;
     - use aggressive learning rates, e.g., winner_lr $= 0.25$, neighbor_lr $= 0.05$;
     - apply entropy-based label-confidence reinforcement (for semi-supervised updates);
     - perform frequent node pruning to reduce overfitting.
   - *Stabilized Mode (high-variance data):*
     - set a static threshold proportional to the median distance in B, e.g., $\mathrm{thr}(t) \propto \mathrm{median}(B)$;
     - use smaller learning rates, e.g., winner_lr $= 0.1$, neighbor_lr $= 0.02$;
     - perform slow pruning with longer edge aging (e.g., max_edge_age ≈ 100);
     - focus on preserving cluster centers (more "memorizing").
4. **Topology update.**
   - Update the winning node and its neighbors using the current learning rates.
   - If $d_t > \mathrm{thr}(t)$, create a new node.
   - Update edge connections and visitation frequencies.
5. **Pruning.**
   - Remove aged edges exceeding max_edge_age.
   - Remove isolated or weakly activated nodes.

and aging limits are treated as fixed hyperparameters that control the aggressiveness of adaptation and pruning. In this work, we follow the recommended ranges from prior SOINN-family studies [10, 11, 12, 13] and empirically confirm that SA-SOINN maintains stable performance under moderate variations of these hyperparameters, leaving a full sensitivity analysis for future work.

For output decisions, SA-SOINN assigns each input to the label of its nearest winning node, producing a predicted class label (binary or multiclass, depending on the datasets). During semi-supervised operation, node-level confidence scores influence label updates but do not alter the deterministic prediction rule at inference time. The final SA-SOINN architecture incorporates optimized adaptive thresholds and pruning heuristics, improving classification performance while retaining its continuous-learning capability.

4. Results and Discussion

4.1. Experimental Setup

The experiments were conducted on the Kaggle computational platform using Python 3.11, running on multi-core CPUs (4 virtual cores, 16 GB RAM). This environment also facilitated the evaluation of multithreaded implementation features integrated into SA-SOINN. Each model was evaluated using the

same runner module to ensure fairness and identical training configurations.

We utilized two widely adopted benchmarks—NSL-KDD and CIC-IDS-2017—which exhibit complementary characteristics and are summarized in Table 1.

- **NSL-KDD** (`KDDTrain.txt`) is a smaller low-variance dataset with a roughly balanced label distribution and moderate class diversity: Denial of Service (DoS), Probing Attack (Probe), Remote to Local Attack (R2L), User to Root Attack (U2R), and Normal traffic. It consists of 125,973 instances and 41 features, and is a multiclass dataset with approximately a 47/53 split between attack and normal samples [31].
- **CIC-IDS-2017** (`Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv`) exhibits high class imbalance and contains dense Distributed Denial-of-Service (DDoS) activity generated using well-known attack tools such as LOIC (Low Orbit Ion Cannon) and HOIC (High OrbitIon Cannon). In our experiments, this subset is used as a binary classification task (DDoS vs. normal traffic). It consists of 225,746 instances and 78 features, with approximately 75% of the traffic labeled as attack [32].

As a non-incremental reference model, we implement a lightweight 1D CNN that operates on standardized feature vectors. The network consists of two 1D convolutional layers with ReLU activations, followed by adaptive max pooling, flattening, and a final linear layer that outputs class scores. The data was standardized and divided into stratified training and test sets, and the model is trained with cross-entropy loss using the Adam optimizer. For the single-pass variant, the CNN is trained for a single epoch to approximate an online learning constraint, whereas the multi-pass variant is trained for multiple epochs in batch mode to obtain a stronger offline reference. Full details of the implementation and additional experiments are reported in our technical report [33].

All experiments followed the benchmarking protocol for data preparation and normalization described in our previous work [11], using a stratified 80/20 train–test split. This protocol evaluates single-pass learning without retraining during the development stage; however, it does not involve a time-ordered streaming evaluation with controlled drift points. Explicit drift-stream benchmarks are reserved for future work during the deployment and testing stages. To prevent data leakage, standardization was applied only to training data and then reused for testing. Feature vectors were normalized using `StandardScaler` (excluding the mean), and categorical fields were one-hot encoded. The model was seeded using three representative samples from distinct classes. Training was conducted in single-pass mode with `learning=True`, while the evaluation of the test set was carried out with `learning=False`. For the full implementation and runner details, we refer the reader to [11]. Performance was evaluated using standard classification metrics: accuracy, precision, recall, F1-score, and running time (T).

Table 1
Summary of the NSL-KDD and CIC-IDS-2017 datasets

Dataset	Year	Instances	Features	Types	Attack	Normal	Label type	Imbalance
NSL-KDD (KDDTrain+)	2009	125,973	41	DoS, Probe, R2L, U2R, Normal	∼47%	∼53%	Multiclass	Moderate
CIC-IDS-2017 (Friday DDoS)	2017	225,746	78	DDoS/DoS vs Normal	∼75%	∼25%	Binary	High

4.2. Comparative Performance: SA-SOINN versus Baselines

Table 2 summarizes the comparative performance of SA-SOINN, SR-SOINN, SF-SOINN, and the CNN baselines under single-pass learning constraints [13, 30, 33].

The results in Table 2 clearly illustrate the stability–adaptability trade-off between the SOINN variants. In the structured NSL-KDD dataset, SR-SOINN's adaptive thresholding achieved higher precision (73.1% vs. 61.0%) and a stronger F1-score (67.7% vs. 62.5%) than SF-SOINN, indicating improved false-positive

reduction and more balanced precision–recall on low-variance data. Although SF-SOINN achieved slightly higher accuracy (66.2% vs. 64.0%), this is mainly due to SR-SOINN's broader adaptive scaling, which can be fine-tuned further. Recall remained similar for both models, suggesting a comparable ability to detect true anomalies.

In contrast, on the complex CIC-IDS-2017 dataset, SF-SOINN performed strongly in all metrics (99.7%). Its fixed thresholds favor the memorization of dense and homogeneous DDoS traffic. However, SR-SOINN suffered a sharp performance drop (56.7%), as its adaptive mechanisms are more sensitive to noise and high variance. SR-SOINN also recorded the longest runtime, reflecting the overhead of repeated recalculations during adaptation. These limitations directly motivated the development of SA-SOINN, which integrates multithreading and optimized code paths to mitigate the computational cost of adaptivity.

We evaluated SA-SOINN against all baselines under the same single-pass constraint to simulate a continuous learning scenario. The CNN serves as an important reference point: it has high capacity, but is inherently non-incremental and requires retraining to maintain accuracy. The results are presented in Table 2.

Table 2
Comparative results of SA-SOINN, SOINN variants, and CNN under single-pass constraints

Dataset	Model	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Time	CPU threads
NSL-KDD	SA-SOINN	**72.8**	72.2	**72.8**	**72.1**	10m 16s	2
NSL-KDD	SA-SOINN	**72.8**	72.2	**72.8**	**72.1**	8m 16s	4
NSL-KDD	SR-SOINN	64.0	**73.1**	64.0	67.7	20m 42s	–
NSL-KDD	SF-SOINN	66.2	61.0	66.2	62.5	14m 21s	–
NSL-KDD	CNN (1 epoch)	59.3	50.6	59.3	53.2	**13s**	–
NSL-KDD	CNN (20 epochs)	71.3	69.4	71.3	69.3	4m 23s	–
CIC-IDS-2017	SA-SOINN	95.1	**99.9**	95.1	97.4	30m 12s	2
CIC-IDS-2017	SA-SOINN	95.1	**99.9**	95.1	97.4	26m 55s	4
CIC-IDS-2017	SR-SOINN	56.7	56.7	56.7	56.7	52m 49s	–
CIC-IDS-2017	SF-SOINN	**99.7**	99.7	**99.7**	**99.7**	13m 04s	–
CIC-IDS-2017	CNN (1 epoch)	99.2	99.2	99.2	99.2	**16s**	–

In NSL-KDD, SA-SOINN achieved the highest accuracy (72.8%) and the most balanced F1-score (72.1%), outperforming all other methods in a single pass. Remarkably, it even outperformed CNN, which required 20 full epochs to reach a lower F1-score of 69.3%. This demonstrates SA-SOINN's capacity to achieve strong performance immediately, without retraining overhead.

In CIC-IDS-2017, SA-SOINN achieved a robust performance (95.1% accuracy, 97.4% F1) with a near-perfect precision of 99.9%. This highlights its ability to avoid false-positive alarms in noisy, high-variance traffic, making it suitable for anomaly detection in such environments. Although CNN also performed well on CIC-IDS-2017, its poor results on NSL-KDD in a single pass (59.3% accuracy, 53.2% F1) highlight its reduced adaptability under single-pass constraints across datasets with differing statistical characteristics.

In terms of computational efficiency, SA-SOINN recorded 10 min 16 s using 2 CPU threads and 8 min 16 s using 4 threads on NSL-KDD—faster than both SR-SOINN and SF-SOINN (14 min 21 s). This shows that the multithreaded implementation scales effectively with the available CPU resources (configurable through the `num_cores = 1, 2, 3, . . . , n` parameter in `SA_SOINN.py`). This scalable execution for speed confirms SA-SOINN's practicality for real-world deployment.

Finally, a closer examination of the baselines reveals important nuances. Although SF-SOINN achieved a near-perfect 99.7% across all metrics in CIC-IDS-2017, this strongly suggests overfitting to the dominant DDoS class rather than true adaptive learning. In contrast, SA-SOINN's combination of dynamic thresholds and adaptive mechanisms produced both high generalizable accuracy and near-perfect precision under single-pass conditions. Its ability to remain stable across datasets with very different statistical properties represents a significant advancement for continuous-learning IDS frameworks.

5. Conclusions and Future Work

This paper introduced SA-SOINN, a self-adaptive neural network (SANN) designed for continuous intrusion detection in dynamic network environments. The framework extends the original SOINN architecture by integrating a variance-aware controller that profiles data distributions in real time and dynamically switches between unsupervised and semi-supervised learning modes. This mechanism addresses the fundamental stability–plasticity trade-off that has traditionally limited machine learning approaches for intrusion detection in evolving environments.

Our evaluation shows that SA-SOINN effectively balances this trade-off, demonstrating strong generalization across diverse data regimes in a single-pass, online learning setting. This is particularly important for real-world IDS scenarios, where retraining is often computationally expensive and operationally impractical. Under a single-pass constraint, SA-SOINN outperforms both its SOINN-family variants and CNN-based baselines on structured datasets such as NSL-KDD, while maintaining high precision, recall, and F1-score on complex datasets like CIC-IDS-2017.

Although SA-SOINN provides a promising solution to intrusion detection in dynamic environments, further evaluation is needed in real-time streaming contexts and concept drift scenarios to fully validate its robustness and adaptability to evolving traffic patterns.

5.1. Future Work

Future efforts will focus on improving the operational versatility and deployment readiness of SA-SOINN in dynamic real-world environments. One key direction involves integration into online cloud ecosystems and containerized platforms (e.g., Docker) to enable scalable, modular, and platform-independent execution within distributed infrastructures. This will improve the algorithm's adaptability to both cloud-based and edge-based deployment scenarios.

We also aim to strengthen the resilience of the self-adaptive framework by contributing the SA-SOINN codebase as an open-source project. This will foster community-driven development and enable further experimentation with its auto-tuning parameters.

Although the present study, at the development stage, focuses on validating SA-SOINN's architectural behavior and learning mechanism through offline evaluation, future research will prioritize real-time deployment and systematic evaluation under evolving data conditions. Specifically, we plan to test SA-SOINN in live streaming environments where traffic characteristics change over time and to compare its performance against established drift-aware methods.

Moreover, SA-SOINN's structural flexibility and hardware awareness make it a promising candidate for cross-domain applications. Future work will explore its deployment in heterogeneous environments, such as IoT networks and cloud platforms, to validate its suitability for edge and distributed IDS scenarios. This will also enable further experimentation with auto-tuning strategies and AI-compatible enhancements to improve performance across diverse operational contexts.

Acknowledgments

This work was supported by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan (PNRR) funded by the European Union – Next Generation EU, Mission 4, CUP G23C24000790006 (2024–25).

Declaration on Generative AI

During the preparation of this work, the author(s) used QuillBot for grammar and spelling checks. Furthermore, the author(s) used Microsoft Copilot for citation management. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. De Nardin, S. Zottin, C. Piciarelli, G. L. Foresti, Deep learning-based intrusion detection systems for phishing email detection: A short survey, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, Honolulu, HI, USA, 2025, pp. 7567–7575. URL: https://openaccess.thecvf.com/content/ICCV2025W/VisionDocs/html/De_Nardin_Deep_Learning-Based_Intrusion_Detection_Systems_for_Phishing_Email_Detection_A_ICCVW_2025_paper.html.

[2] A. Khraisat, I. Gondal, P. Vamplew, J. Kamruzzaman, Survey of intrusion detection systems: Techniques, datasets and challenges, Cybersecurity 2 (2019) 20. doi:10.1186/s42400-019-0038-7.

[3] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, A. Bouchachia, A survey on concept drift adaptation, ACM Computing Surveys 46 (2014) 44:1–44:37. doi:10.1145/2523813.

[4] F. Shen, O. Hasegawa, Self-organizing incremental neural network and its application, in: K. Diamantaras, W. Duch, L. Iliadis (Eds.), Artificial Neural Networks – ICANN 2010, volume 6354 of *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2010, pp. 535–540. doi:10.1007/978-3-642-15825-4_74.

[5] P. Zhou, A survey of streaming data anomaly detection in network security, PeerJ Computer Science 11 (2025) e3066. doi:10.7717/peerj-cs.3066.

[6] F. Cerasuolo, G. Bovenzi, D. Ciuonzo, A. Pescapè, Adaptable, incremental, and explainable network intrusion detection systems for internet of things, Engineering Applications of Artificial Intelligence 144 (2025) 110143. doi:10.1016/j.engappai.2025.110143.

[7] B. Chen, J. Jiang, X. Wang, P. Wan, J. Wang, M. Long, Debiased self-training for semi-supervised learning, in: Advances in Neural Information Processing Systems 36 (NeurIPS 2022), New Orleans, LA, USA, 2022. doi:10.5555/3600270.3602619, neural Information Processing Systems Foundation.

[8] P. Cascante-Bonilla, F. Tan, Y. Qi, V. Ordonez, Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, 2021, pp. 6912–6920. doi:10.1609/aaai.v35i8.16852.

[9] Y. Mirsky, T. Doitshman, Y. Elovici, A. Shabtai, Kitsune: An ensemble of autoencoders for online network intrusion detection, in: Proceedings of the Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 2018. doi:10.14722/ndss.2018.23204.

[10] M. R. Martina, G. L. Foresti, A continuous learning approach for real-time network intrusion detection, International Journal of Neural Systems 31 (2021) 2150060. doi:10.1142/s012906572150060x.

[11] D. G. Ejeh, G. L. Foresti, M. Miculan, A. D. Nardin, Real-time anomaly detection in docker containers: A continuous learning approach using sf-soinn, in: CEUR Workshop Proceedings, Bologna, Italy, 2025. URL: https://ceur-ws.org/Vol-3962/paper45.pdf.

[12] C. Wiwatcharakoses, D. Berrar, Soinn+, a self-organizing incremental neural network for unsupervised learning from noisy data streams, Expert Systems With Applications 143 (2019) 113069. doi:10.1016/j.eswa.2019.113069.

[13] D. G. Ejeh, G. L. Foresti, M. Miculan, A. D. Nardin, Sr-soinn: A semi-supervised extension for structured data learning, Technical Report, Kaggle, 2025. URL: https://www.kaggle.com/code/dennisejeh/sr-soinn.

[14] N. Gunasekara, B. Pfahringer, H. M. Gomes, A. Bifet, A survey on online streaming continual learning, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI-23), Survey Track, 2023, pp. 6628–6637. doi:10.24963/ijcai.2023/743.

[15] 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, Engineering Applications of Artificial Intelligence (2024). doi:10.1016/j.engappai.2024.109143, early access.

[16] Z. Cai, J. Chen, M. Liu, Self-adaptive deep neural network: Numerical approximation to functions and pdes, Journal of Computational Physics 455 (2022) 111021. doi:10.1016/j.jcp.2022.

111021.

[17] F. Stolzenburg, S. Litz, O. Michael, O. Obst, Efficient time-series approximation with linear recurrent neural networks: Architecture learning and predictive power, Neural Computing and Applications 37 (2025) 27027–27055. doi:10.1007/s00521-025-11655-y.

[18] C. Xu, Y. Zhan, G. Chen, Z. Wang, S. Liu, W. Hu, Elevated few-shot network intrusion detection via self-attention mechanisms and iterative refinement, PLOS ONE (2025). doi:10.1371/journal.pone.0317713.

[19] A. Miguel-Diez, A. Campazas-Vega, C. Álvarez-Aparicio, G. Esteban-Costales, A. M. Guerrero-Higueras, A systematic literature review of unsupervised learning algorithms for anomalous traffic detection based on flows, Logic Journal of the IGPL (2025). doi:10.1093/jigpal/jzaf020.

[20] F. Jemili, K. Jouini, O. Korbaa, Intrusion detection based on concept drift detection and online incremental learning, International Journal of Pervasive Computing and Communications 21 (2025) 81–115. doi:10.1108/IJPCC-12-2023-0358.

[21] D. Lukats, O. Zielinski, A. Hahn, F. Stahl, A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams, International Journal of Data Science and Analytics 19 (2025) 1–31. doi:10.1007/s41060-024-00620-y.

[22] J. Ren, L. Liu, H. Huang, J. Ma, C. Zhang, L. Wang, B. Liu, Zhao, Soinn intrusion detection model based on threeway attribute reduction, Electronics 12 (2023) 5023. doi:10.3390/electronics12245023.

[23] X. Zhang, R. Zhao, Z. Jiang, H. Chen, Y. Ding, E. Ngai, S.-H. Yang, Continual learning with strategic selection and forgetting for network intrusion detection, in: IEEE INFOCOM 2025 – IEEE Conference on Computer Communications, London, United Kingdom, 2025. doi:10.1109/INFOCOM55648.2025.11044615.

[24] T. Zoppi, A. Ceccarelli, Prepare for trouble and make it double! supervised–unsupervised stacking for anomaly-based intrusion detection, Journal of Network and Computer Applications 189 (2021) 103106. doi:10.1016/j.jnca.2021.103106.

[25] H. Doroud, O. Alkhateeb, E. A. Jarchlo, F. Dressler, Lids: A lifelong learning approach for intrusion detection, in: Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC), 2023, pp. 482–487. doi:10.1109/IWCMC58020.2023.10182443.

[26] H. Wang, Y. Yang, P. Tan, Ctwa: A novel incremental deep learning-based intrusion detection method for the internet of things, Artificial Intelligence Review 58 (2025). doi:10.1007/s10462-025-113589.

[27] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, X. Wang, A comprehensive survey of neural architecture search: Challenges and solutions, ACM Computing Surveys 54 (2021) 1–37. doi:10.1145/3447582.

[28] Y. Qu, Q. Zhang, M. Zheng, L. Yang, Network intrusion detection by adaptive deep metric learning, in: CloudComp 2024, volume 617 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Springer, 2025, pp. 105–117. doi:10.1007/978-3-031-92517-7_8.

[29] M. A. Hossain, Deep q-learning intrusion detection system (dq-ids): A novel reinforcement learning approach for adaptive and self-learning cybersecurity, ICT Express 11 (2025) 875–880. doi:10.1016/j.icte.2025.05.007.

[30] D. G. Ejeh, G. L. Foresti, M. Miculan, A. D. Nardin, Technical report: Introducing sa-soinn – a self-adaptive ids for dynamic environments, Technical Report, Kaggle, 2025. URL: https://www.kaggle.com/code/dennisejeh/sasoinn-ver2.

[31] M. Tavallaee, E. Bagheri, W. Lu, A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 53–58. doi:10.1109/CISDA.2009.5356528.

[32] R. Panigrahi, Cicids2017, IEEE Dataport, 2025. doi:10.21227/akxq-9v09.

[33] D. G. Ejeh, G. L. Foresti, M. Miculan, A. D. Nardin, Technical report: Introducing sa-soinn – cnn test, Technical Report, Kaggle, 2025. URL: https://www.kaggle.com/code/dennisejeh/cnn-test.