SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I9P102

Volume 12 Issue 9, 11-31, September 2025
© 2025 Seventh Sense Research Group®

Original Article

CyberAdaptAI: A Dynamic Ensemble Learning Framework for Real-Time Cyberattack Detection Using AdaptEnsembleNet

Nagamani Uddamari[1], P. Sammulal[2]

[1]*Department of CSE, JNTUH, Hyderabad, India.*
[2]*Department of CSE, JNTUH College of Engineering, Hyderabad, India.*

[1]*Corresponding Author : nagamani.u@gmail.com*

Received: 02 July 2025 Revised: 04 August 2025 Accepted: 03 September 2025 Published: 29 September 2025

Abstract *- Networked systems have been expanding rapidly, and there are cybersecurity challenges that require advanced Intrusion Detection Systems (IDS) to detect sophisticated and evolving threats. However, the more common traditional IDS approaches, including signature-based and classical machine learning methods, usually suffer from a significant drop in performance as they typically cannot adapt well to concept drift and data imbalance and cannot provide enough interpretability [6-9]. In dynamic networks, these challenges prevent faster and accurate detection of new attacks or zero-day attacks. This article presents CyberAdaptAI, a novel hybrid adaptive ensemble learning framework that combines several base classifiers through an efficient drift detection scheme and adaptive weight rebalancing to overcome these limitations [38]. It has also integrated explainability through SHAP-based interpretability, leading to actionable insights for security analysts. The general approach is to apply mini-batch processing of the streaming network data, dynamically tuning the classifier weights based on the most recent performance, concept detection using ADWIN, and a mechanism triggered by concept drift detection to train new models to maintain accuracy. CyberAdaptAI achieves up to 98.1% and 96.8% accuracy on benchmark datasets CIC-IDS2017 and UNSW-NB15, while outperforming state-of-the-art baselines empirically evaluated. Not only does the model recover quickly after encountering drift events, but it is also consistent and stable during batch-wise performance. Besides, cross-dataset evaluations substantiate its robustness and generalization abilities in a heterogeneous network scenario. The solution provided by CyberAdaptAI enables a practical and scalable approach to real-time intrusion detection in complex and evolving cyber environments, relying on adaptability, accuracy, and interpretability. By seamlessly enabling network behaviors of relevance and integrated with transparent decision-making, the framework adds novel support for security operations and threat mitigation, addressing critical gaps in existing IDS methodologies.*

Keywords *- Adaptive Intrusion Detection, Ensemble Learning, Concept Drift, Explainable AI, Network Security*.

1. Introduction

The growing complexity of cyber-attacks and the proliferation of networked technologies have made developing effective Intrusion Detection Systems (IDS) a paramount concern as more critical infrastructures become integrated into connected environments. Traditional IDS strategies usually use static rule-based means or machine learning designs, which are highly inefficient at discovering sophisticated and evolving attacks, particularly zero-day and polymorphic attacks. In recent years, advancements in deep learning and ensemble methods have shown promising results in improving the accuracy and robustness of detection. Nevertheless, the models' failure is witnessed due to concept drift, i.e., the statistical properties of the network traffic are changing over time; hence, they may not perform well in a dynamic real-world environment [1, 2]. Additionally, many deep learning models lack interpretability, making them cumbersome for security analysts who value transparency and actionable insights into how a model reached a given decision [3].

Many of the research works have proposed different methods of enhancing IDS competency through feature selection, ensemble learning, and unsupervised techniques for anomaly detection [4-6]. Adaptive frameworks that can detect drift and learn incrementally have been proposed in several studies to confront the issue of temporal data variability [7, 8]. Despite these developments, the model integration that promises calculative, efficient, robust adaptation and interpretability with high accuracy across varied datasets remains elusive. Filling in this gap further motivates the current research to develop an adaptive ensemble IDS that dynamically responds to concept drift with better detection performance and interpretability.

Although several intrusion detection systems have been proposed using Ensemble and deep learning methods, most existing works still fall short in three key aspects: they lack adaptability to rapid concept drift in evolving network traffic, they fail to generalize effectively across heterogeneous datasets, and they offer limited interpretability for security analysts. These gaps lead to reduced robustness in real-time deployment and hinder practical adoption. Therefore, the problem addressed in this study is the development of a framework that ensures high detection accuracy under dynamic traffic conditions, demonstrates strong cross-dataset generalization, and provides transparent, explainable insights for decision-making.

The novelty of this research lies in the integration of three complementary advancements into a unified framework. First, CyberAdaptAI employs an adaptive ensemble learning strategy with drift-aware weight rebalancing, enabling the model to maintain high detection accuracy even under rapidly evolving traffic conditions. Second, unlike many prior IDS studies that evaluate models only on a single benchmark dataset, CyberAdaptAI validates performance across both CIC-IDS2017 and UNSW-NB15, thereby demonstrating superior cross-dataset generalization. Third, the incorporation of SHAP-based interpretability provides transparent feature-level explanations that support security analysts in understanding and trusting model decisions. These combined innovations distinguish CyberAdaptAI from existing intrusion detection approaches that typically address accuracy, adaptability, or interpretability in isolation.

This research aims to 1) propose a hybrid machine learning framework using adaptive ensemble techniques to increase the anomaly detection performance, and to 2) develop a drift detection and adaptation mechanism to maintain performance over time, and finally 3) integrate explainable AI methods such as SHAP to provide actionable security insights. The novelty of this work lies in integrating drift-aware ensemble learning with different explainability modules and testing the proposed method on benchmark datasets CIC-IDS2017 and UNSW-NB15. Notably, this integration offers high detection accuracy and model explainability, which are paramount for real-time cyber-risk operations.

The structure of the paper is as follows: In Section 2, we provide a complete literature review, including a summary of the existing methods and some of their drawbacks. Proposed Methodology: The proposed methodology is described in detail in section 3 through model architecture, drift detector, and interpretability techniques. Section 4 describes the experimental setting, the datasets used, and the results obtained. Section 5 presents the results, discusses a study limitation, and provides further implications. Finally, in Section 6, we conclude the paper and suggest future research directions to increase IDS adaptability and applicability in more complicated cyber environments.

2. Related Work

Recent studies focus on dynamic, Ensemble, and hybrid learning methods to enhance cybersecurity's real-time intrusion detection accuracy and adaptability. Zhijun Wu et al. [1] introduced DEIL-RVM. This dynamic ensemble intrusion detection technique uses probabilistic updates and sparse RVMs to achieve steady accuracy on streaming network data while consuming minimal resources. Huajuan Ren et al. [2] proposed ADHS-EL. This Boosting-based Ensemble improves accuracy and robustness on adversarial, unbalanced network traffic datasets with intentions for future generalization, using dynamic hybrid sampling and adversarial augmentation. Xinghua Li et al. [3] presented a sustainable ensemble intrusion detection model that enhances accuracy and robustness on NSL-KDD and real-world datasets by reusing historical information and adapting to different types of attacks. Farah Jemili et al.[4] suggested a universal intrusion detection framework that uses ensemble learning, PCA, cosine similarity, and TF-IDF. High accuracy is attained in tests on CICIDS, NSL-KDD, and UNSW; nonetheless, computational and dataset diversity constraints are present. METHAQ A. SHYAA et al. [5] A feature drift-aware IDS framework called IFDA-GPC is presented in this paper. It uses VE-DQN-MAFS for dynamic feature selection. It was tested on several datasets and demonstrated good adaptability with an accuracy of 93% on the CICIDS-2017; scalability and deep learning integration require more development.

Appalaraju Grandhi and Sunil Kumar Singh [6] present the optimized feature selection model IDBFS-EGTO, which achieves 98.4% intrusion detection accuracy, in the paper. It has strengths in exploration-exploitation balance, tuning, and evaluating complexity issues. Sydney Mambwe Kasongo [7] suggested an IDS framework that combines feature selection based on XGBoost with RNN variations (LSTM, GRU, and Simple RNN). Its accuracy reached 88.13% when tested on the NSL-KDD and UNSW-NB15 datasets. Benefits include faster training, less feature space, and enhanced performance. The model's performance on minority classes is limited; hybrid RNNs and more in-depth class-level analysis are part of future research. Ahmed Abdelkhalek and Maggie Mashaly [8] introduced a deep learning-based NIDS for class imbalance handling that uses Tomek Links and ADASYN. It outperformed current models with an accuracy of up to 99.9% when tested on NSL-KDD.

SoumyadeepHore et al. [9] presented DeepResNIDS. This multistage DNN-based intrusion detection framework achieves 98.5% accuracy in detecting known, zero-day, and adversarial assaults through transfer learning and autoencoders. Mamatha Maddu and Yamarthi Narasimha Rao [10] suggested a deep learning-based intrusion detection system (IDS) for SDN that uses DCGAN, CenterNet, and ResNet152V2 with SMA. With an accuracy of 99.65%, it seeks to enhance zero-day detection in Internet of Things networks.

Nojood O. Aljehane et al. [11] present the GJOADL-IDSNS intrusion detection system in the paper. It improves performance on benchmark datasets by utilizing A-BiLSTM classification, GJOA-based feature selection, and SSA for hyperparameter adjustment. Future research will focus on real-time implementation and threat adaptation. Khushnaseeb Roshan et al. [12] examined adversarial attacks on NIDS and suggested three defense tactics to improve robustness: High Confidence, Gaussian Data Augmentation, and Adversarial Training. Applying the method to different ML/DL architectures is part of future development. Hichem Sedjelmaci [13] suggested a 5G network security detection method based on hierarchical reinforcement learning. It uses minimal processing overhead to identify unknown assaults. In the future, network performance will be assessed and tested in actual 5G scenarios. Ahmad

Ali AlZubi et al. [14] suggested an Attack Detection Framework (CML-ADF) that uses cognitive machine learning to help secure healthcare data in cyber-physical systems. It achieves high attack prediction and accuracy while increasing efficiency and lowering communication costs. Developing intelligent security procedures and tackling security issues are two areas of future research. SHAKILA ZAMAN et al. [15] examined IoT security risks and AI-powered defenses, emphasizing the difficulties posed by devices with limited resources. It discusses AI/ML methods, unresolved issues, and potential ways to enhance IoT effectiveness and security.

CELESTINE IWENDI et al. [16] introduced a deep learning Intrusion Detection System (IDS) that uses an LSTM classifier to detect cyberattacks on the Internet of Things with an accuracy of 99.09%. Future research will examine blockchain for improved IDS and apply SNMP to big networks. MUJAHEED ABDULLAHI et al. [17] tested the LSTM and XGBoost models for cyberattack detection in CPS using benchmark datasets and gas pipelines. XGBoost achieved 98.69% accuracy; ensemble approaches and real-time datasets will be used in future studies. Aya H. Salem et al. [18] examined 68 AI-based techniques for detecting cyberattacks, emphasizing ML, DL, and metaheuristic algorithms.

Tested against various threats, the results indicate better detection, but the demands on data and computation are significant. Muhammad Mudassar Yamin et al. [19] examined new AI-powered cyberattacks, described existing offensive and defense tactics, warned of the dangers of AI weaponization, and urged international collaboration for responsible AI cybersecurity development. KAVITHA DHANUSHKODI AND S. THEJAS [20] examined AI-driven cybersecurity developments, demonstrating enhanced detection using innovative models in various sectors, addressing issues like privacy and integration, and suggesting future paths for workable, scalable deployment.

Sowmya T. and Mary Anita E. A [21] examined 72 papers on AI-based intrusion detection, emphasizing Ensemble, ML, and DL techniques with an accuracy of \~99% on typical attacks. Although it highlights limitations in attack classification, evaluation metrics, and dataset variety, it also identifies benefits in detection accuracy. Utilizing more recent datasets, testing hybrid models, and improving the detection of unidentified attacks are all examples of future development. Salwa Alem et al. [22] presented BIANO-IDS, a novel intrusion detection system that combines specification- and anomaly-based techniques using neural networks and a decision system. In actual industrial testing, it achieved low false positives and high accuracy. Future research aims to increase efficiency through feature selection and broader data sources.

Heng Zeng et al. [23] integrated the theories of CAS, TAM, and TPB to present a novel AI-based anomaly detection framework for IoT security in smart cities. It highlights human-AI interaction, which has been conceptually proven. Future research will concentrate on cross-cultural adaptation, ethical issues, and real-world validation. Matthew Baker et al. [24] presented an integrated LSTM-MPC real-time anomaly detection and correction system for power electronic-dominated grids. It has been tested on a 14-bus system and has demonstrated fault correction, resilience, and accurate classification. Limitations include scalability and real-world deployment; future work will concentrate on growing datasets and adaptive learning for broader grid integration. Benefits include real-time detection and correction. Monika Vishwakarma and Nishtha Kesswani [25] used a benchmark dataset to demonstrate a deep neural network-based Intrusion Detection System (IDS) for real-time attack detection in IoT networks. The results indicate improved efficiency; real-time training and larger datasets will be the main topics of future research.

Md. Asaduzzaman and Md. Mahbubur Rahman [26] suggested a hybrid LSTM-CNN model for AWID and GAN-generated datasets to detect zero-day threats. The accuracy of the model was 93.53%. More attack data generation is the goal of future development to improve detection. ZHIBO ZHANG et al. [27] suggested a hybrid LSTM-CNN model for AWID and GAN-generated datasets to detect zero-day threats. The accuracy of the model was 93.53%. More attack data generation is the goal of future development to improve detection.

Ankit Attkan and Virender Ranga [28] discuss blockchain and AI-based authentication for safe device communication, which are the main topics of the paper's evaluation of IoT security solutions. Future studies focusing on enhancing security methods emphasize their advantages for key management. Marcos V.O. de Assis et al. [29] suggested an SDN-based security system that uses game theory for mitigation and CNN for real-time detection to stop DDoS attacks. Testing with other hosts and investigating deep learning techniques are part of the future effort. Norberto Garcia et al. [30] An AI-based anomaly detection system for detecting SlowDoS attacks in real time over encrypted HTTP data is presented in this study. With a 98% accuracy rate, future research will concentrate on modifying the system to withstand further threats and 5G traffic.

Jalindar Karande and Prof. Sarang Joshi [31] offered a Google Cloud experimental setup for real-time IoT security analytics. It does not automatically learn new assaults, but it can recognize known ones. Future research focuses on identifying new assaults early. Stefanos Tsimenidis et al. [32] examined deep learning models for IoT intrusion detection and emphasised their effectiveness compared to conventional techniques. It discusses difficulties and recommends further study on unsupervised, distributed, and effective deep learning methods. MINH-QUANG TRAN et al. [33] introduced a deep learning and Internet of Things-based solution for CNC machine monitoring that uses vibration sensors to guarantee cutting stability. The results demonstrate remarkable precision, surpassing conventional security and vibration control techniques. MOHAMEDS. ABDALZAHER et al. [34] discussed the role of IoT and machine learning in intelligent systems, provided a taxonomy of ML models for IoT security, and offered research recommendations in addition to case studies on smart cities and early warning systems. Martin Manuel Lopez et al. [35] addressed idea drift and excessive verification delay by installing an SCARGC-based intrusion detection system for the Internet of Things. When tested on actual IoT datasets, it demonstrates increased accuracy; deep learning models will be the focus of future research.

VANLALRUATA HNAMTE et al. [36] present a hybrid LSTM-AE intrusion detection model that outperforms CNN and DNN models in the paper. It demonstrated 99.99% accuracy when tested on the CICIDS2017 and CSE-CICIDS2018 datasets. Future research will focus on transfer learning and alternative architectures. JIAWEI DU et al. [37] presented NIDS-CNNLSTM, a network intrusion detection system for the IIoT that combines CNN and LSTM for excellent accuracy. Tested on the UNSW_NB15, NSL_KDD, and KDD CUP99 datasets, it lowers false alarms and increases detection rates. Tao Yi et al. [38] examined deep learning methods for detecting network assaults, including data imbalance, traffic representation, and dynamic attacks. It evaluates current solutions and identifies problems and potential avenues for future study.

An ensemble method with distributed machine learning for detecting idea threats and drift based on Apache Spark, which was explained by Meenal Jain and Gagandeep Kaur [39]. The results show a high accuracy of 93% on NSL-KDD. In the future, the aim will be to classify the threats specific to IoT and improve the classifiers. MAHMOUDABBASI et al. AWEE, a method to address the class imbalance problem in network traffic classification by dynamically changing the weight of the misclassified and correctly classified instances, and an ensemble learning process is proposed by [40]. It outperforms existing methods with an accuracy greater than 98%. This is something that may be explored in future development work. The existing literature on AI-based intrusion detection using ensemble models, deep learning, and drift-aware techniques was reviewed. The techniques utilize real-time aspects, adaptive weighting methods, and enhanced feature selection methods to effectively detect known and novel threats and thus ameliorate the detection. Nevertheless, scalability, class imbalance, and model generalizability to heterogeneous network environments still pose challenges.

Although the literature demonstrates significant progress in intrusion detection using ensemble methods, deep learning, and drift-aware models, most studies address only a subset of the key challenges. Ensemble and boosting techniques have improved accuracy but often fail to maintain robustness under evolving traffic distributions. Drift-aware frameworks capture temporal changes but typically sacrifice accuracy or computational efficiency. Deep learning approaches achieve strong detection rates but remain limited by interpretability and a lack of transparency for analysts. Importantly, few works systematically integrate adaptability, cross-dataset validation, and explainability into a single model. This gap highlights the need for a holistic framework such as CyberAdaptAI, which simultaneously addresses these dimensions to provide resilient, accurate, and interpretable intrusion detection in real-world environments.

3. Proposed Framework

In this section, the proposed framework, CyberAdaptAI, an adaptive ensemble incremental learning model for intrusion detection in real-time, is presented. It achieves high accuracy under changing network traffic by employing a set of classifiers with dynamic weight tuning and a drift detection mechanism. Furthermore, the framework combines explainability methods to produce interpretable and actionable insights, bolstering performance and explainability in cybersecurity domains.

3.1. Overview

The CyberAdaptAI system, illustrated in Figure 1, is a resilient and adaptive cybersecurity framework for online intrusion detection in adaptive network-based environments. The system consists of four integrated layers (Figure 1). These layers are data ingestion, adaptive preprocessing, intelligent ensemble classification, and real-time decision output. Our framework receives the network traffic through structured data streams or simulated batches from benchmark datasets (CIC-IDS2017, UNSW-NB15, etc). These inputs include benign and attack behavior, resulting in a diverse and rich data set for evaluation. After all, the preprocessing module handles data cleaning, categorical encoding, and normalization to provide a consistent representation as input while removing network telemetry's inherent noise and inconsistency.

After the preprocessing step, the feature vectors are sent to the orchestration module, where the AdaptEnsembleNet functions as a core classifier. This module consists of a collection of base learners that can vary during the run and includes Random Forest, XGBoost, Extra Trees, and adds classifiers such as Hoeffding Trees that can also learn online. Recent performance, prediction confidence, and ensemble diversity are the factors that determine the adaptive weights of these learners. To deal with different types of attackers and concept drift, it integrates an optional drift detection mechanism (e.g., ADWIN) that observes classification statistics over sliding windows and forces an update of the model when a change is detected in the data distribution. The ensemble voting mechanism used is not fixed, but continually adjusted via a weighted majority voting approach in which classifiers displaying higher recent accuracy and diversity have more significant influence over the final decision.

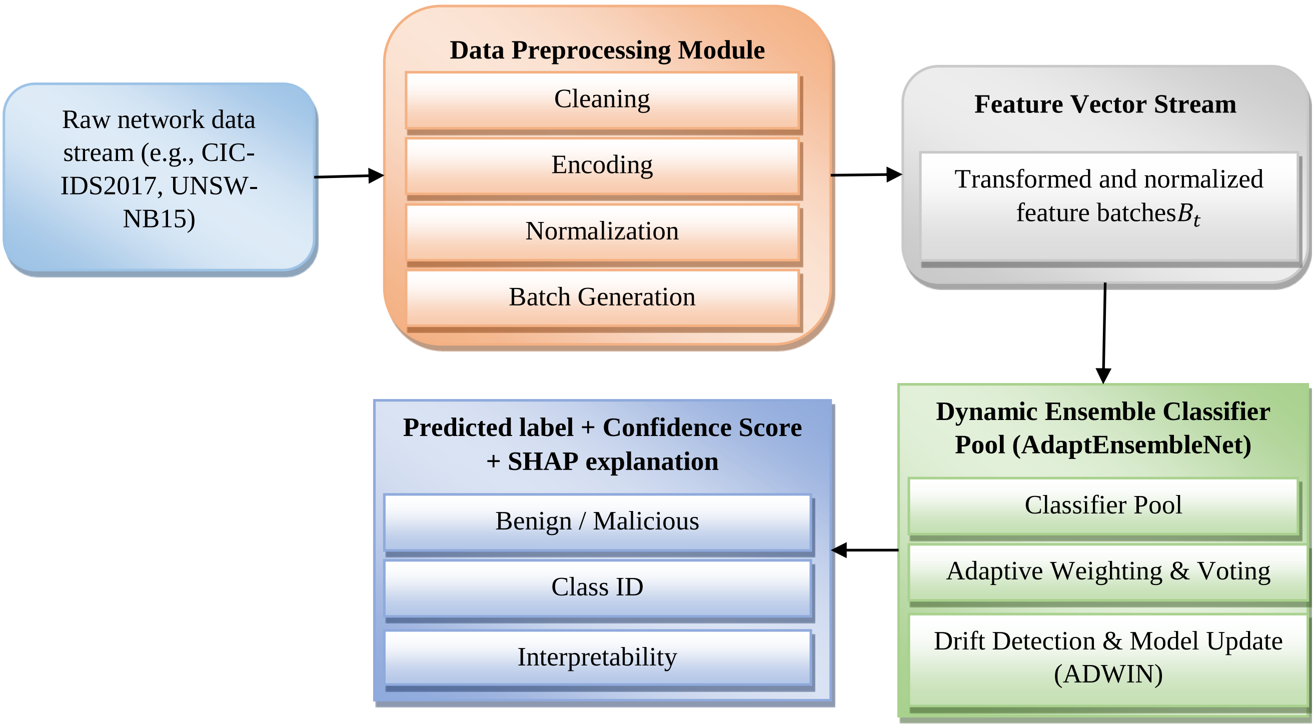
Raw network data stream (e.g., CIC-IDS2017, UNSW-NB15)
Data Preprocessing Module
Cleaning
Encoding
Normalization
Batch Generation
Feature Vector Stream
Transformed and normalized feature batchesB_t
Dynamic Ensemble Classifier Pool (AdaptEnsembleNet)
Classifier Pool
Adaptive Weighting & Voting
Drift Detection & Model Update (ADWIN)
Predicted label + Confidence Score + SHAP explanation
Benign / Malicious
Class ID
Interpretability

Fig. 1 CyberAdaptAI system architecture for real-time intrusion detection

Table 1. Notations and Symbols Used in the CyberAdaptAI Framework

Symbol	Description
$D = \{(x_i, y_i)\}$	Complete dataset with feature-label pairs
$x_i \in \mathbb{R}^d$	d-dimensional feature vector of instance i
$y_i \in \{0,1,\ldots,C\}$	Class label assigned to instance x_i
B_t	Mini-batch (stream segment) at time step t
n_t	Number of instances in batch B_t
$\mathcal{H} = \{h_1, \ldots, h_M\}$	Ensemble of MM base classifiers
$h_m(x_i)$	Prediction for instance x_i by classifier h_m
$\hat{y}_i$	Final predicted class label for instance x_i
w_m	Adaptive weight for classifier h_m
A_m^t	Accuracy of the classifier $h_m$at time t
D_m^t	Diversity score of the classifier h_m at time t
$Q(h_m, h_n)$	Q-statistic measuring disagreement between classifiers h_m and h_n
θ_m^t	Parameter vector of classifier $h_m$at time t
η	Learning rate for incremental model updates
$\mathcal{L}$	Classification loss function (e.g., cross-entropy)
μ_1, μ_2	Mean error in ADWIN's two sliding sub-windows
ϵ_{cut}	Cutoff threshold for drift detection using ADWIN
δ	Drift detection confidence level
γ	Smoothing factor for error-based weight update
E_m^t	Exponential moving average error for the classifier h_m at time t
z_c	Logit output score for class c
$P(y = c \mid x_i)$	Predicted probability of class cc for input x_i
$Conf(x_i)$	Decision confidence score, for instance x_i

CyberAdaptAI produces a classification decision in real-time that marks incoming traffic to the server as benign or belonging to one of several attack classes. Such decisions are made with a limited latency, enough to facilitate the operation of such systems in real-life scenarios. The hybrid model integrates adaptive learning with ensemble diversity and streaming capability, allowing it to achieve high detection accuracies and resilience against evolving cyber-attacks. This modular nature also ensures that CyberAdaptAI can be individually optimized per subsystem, making it deployment-ready for scalable edge-cloud or SOC (Security Operations Center) based infrastructures. Table 1 summarizes all key notations and symbols used in the CyberAdaptAI framework for clarity in methodology and implementation.

3.2. Dataset Preparation and Stream Simulation

This research needs datasets that can be closely evaluated against real-world network environments to evaluate with our CyberAdaptAI framework. Therefore, we used the CIC-IDS2017 and UNSW-NB15 datasets. We used these benchmark datasets on labeled network traffic for both normal and malicious behavior: a Denial of Service (DoS) attack, a brute force attack, a botnet attack, and an infiltration attack. It contains multiple features such as packet duration, flow bytes per second, header flags, and protocol information for each record, and hence it can be used in supervised learning. All datasets undergo a downstream preprocessing pipeline to bring data into a consistent and machine-readable state.

In the first stage, data cleaning includes mean imputation of missing values and eliminating duplicate records. In this case, define the dataset as $D = \{(x_i, y_i)\}_{i=1}^{N}$, in $x_i \in \mathbb{R}^d$ Which is the d-dimensional feature vector of the i-th network instance and $y_i \in \{0,1,\dots,C\}$ is the class label. Symbolic attributes are converted to numeric form either by label encoding or one-hot encoding for each categorical feature f_j in x_i. Implemented min-max scaling on the numerical features to normalise them numerical features to the range defined as in Equation (1).

$$x'_{ij} = \frac{x_{ij} - min(x_j)}{max(x_j) - min(x_j)} \qquad (1)$$

Where x_{ij} Is instance i value for feature j and x'_{ij} The normalised value. This step avoids the dominance of features in model training. Also, it leads to quicker convergence for ensembling learners.

In other words, to mimic a stream of real-time data, the preprocessed dataset D is split into several consecutive mini-batches $B_t = \{(x_i, y_i)\}_{i=1}^{n_t}$, where t is the batch index and n_t Is the number of instances in a batch. This means that each batch is treated separately, mimicking streaming network telemetry. It allows for monitoring temporal performance and model updating as the system sees new data. The stream-based formulation facilitates an evaluation of adaptive behavior, specifically the capacity of the model to adapt to temporal changes and possible concept drift. We also keep track of the class distribution of each batch, so it can be evaluated how much imbalance may have influenced the Ensemble and dynamically reweight the Ensemble based on these statistics.

The training and evaluation pipeline follows a sequential holdout approach, in which prior batches are used for training and following batches are used for evaluation. It ensures that the model is evaluated on unseen data, which promotes generalization. Denote $B_{train} = \bigcup_{t=1}^{T} B_t$ the training and $B_{test} = \bigcup_{t=T+1}^{T+K} B_t$ Test segments as and.. Holding the ratio fixed at 80:20 in train to test split across streams, but the structure allows a flexible resampling strategy for comparison of experiments.

3.3. Model Architecture: AdaptEnsembleNet

At the heart of the CyberAdaptAI system is the AdaptEnsembleNet model. This adaptive ensemble learning architecture effectively addresses dynamic network environments by implementing real-time performance-based reweighting of multiple classifiers. In the case of mini-batches or streaming data, it adapts its internal classifier weights corresponding to the classifier in various categories over time using recent predictive accuracy, confidence scores, and classifier diversity, as shown in Figure 2. The architecture consists of three main components: a pool of base learners, a diversity and performance observation module, and an adaptive voting module.

Denote $B_t = \{(x_i, y_i)\}_{i=1}^{n_t}$ The mini-batch with and denote labels by where each $x_i \in \mathbb{R}^d$ It is a feature vector and the corresponding label. y_i.AdaptEnsembleNet has a pool of M base classifiers $\mathcal{H} = \{h_1, h_2, \dots, h_M\}$, which are all independently trained on previous data or registered in some require-less online learning algorithms. Given each instance x_i, the m-th classifying scheme uses a predicted class label $\hat{y}_i{}^{(m)} = h_m(x_i)$. An ensemble prediction y_i is then calculated using a weighted majority voting scheme as in Equation (2):

$$\hat{y}_i = arg \max_{c} \sum_{m=1}^{M} w_m \cdot \mathbb{I}[h_m(x_i) = c] \qquad (2)$$

Where w_m Is the classifier, and is the adaptive weight w_m assigned to the classifier h_m, and $\mathbb{I}[\cdot]$ is the indicator function. The accuracy and diversity are computed using a composite function, and, hence, the weights are updated dynamically on the fly at each step. In specific, let A_m^t be the most recent accuracy of the classifier h_m on the current(but may not always) or earlier batch, and D_m^t The average disagreement with the remaining classifiers is that the adaptive weight is calculated as follows, as Equation (3)

$$w_m^t = \alpha \cdot A_m^t + (1 - \alpha) \cdot D_m^t \qquad (3)$$

Where $\alpha \in [0,1]$ trades off accuracy for diversity. The diversity metric D_m^t is calculated by the Q-statistic or disagreement measure based on classifier pairs, as in Equation (4):

$$Q(h_m, h_n) = \frac{N11N00 - N10N01}{N11N00 + N10N01} \quad (4)$$

Where N_{ab} is the number of predictions made by the classifier h_m for class aa and $h_n s$ for class b. If the average disagreement score is high, it indicates that the classifier brings new decision boundaries, thus strengthening the Ensemble.

AdaptEnsembleNet allows for batch and online updates. Batch mode: All classifiers are retrained periodically, based on the available labeled data. For the online setting, Hoeffding trees and their adaptive boosting variants are updated instance-by-instance so that they can evolve continuously. This feature makes it capable of being responsive and accurate regarding concept drift. In addition, it can optionally integrate softmax-based confidence scores from our probabilistic classifiers into the voting weights, allowing us to make finer decisions in cases of ambiguity.

AdaptEnsembleNet is adaptive in nature, so it can easily be used in rapidly changing cybersecurity environments. Attack distributions evolve at a fast pace while inflexible models struggle to generalize. It utilizes a performance-aware, diversity-driven voting mechanism to balance stability and plasticity, resulting in better generalization and robustness.

3.4. Online Learning and Drift Handling

The cyber behavior pattern and original threat for cyberattack dynamics are in nature, and thus, cyberattacks like this are also very dynamic in real-world network environments. The CyberAdaptAI framework embeds online learning capabilities and concept drift detection in the AdaptEnsembleNet model to handle this non-stationarity. This adaptation allows the classifier to better adapt to time-varying distributions and high detection performances as the attacks vary.

The most relevant one to our work is online learning, in which data is provided in a sequence of batches. $B_t = \{(x_i, y_i)\}_{i=1}^{n_t}$ A set of data instances, and the model is updated after processing each batch or new instance of data. The base classifiers are then updated with a partial fitting mechanism after the classification of the instances in B_t. In the case of classifiers $h_m \in \mathcal{H}$ with an online learning functionality (Ex, Hoeffding Trees, adaptive SGD-based models), the update rule for each model can be written as in Equation (5):

$$\theta_m^{t+1} = \theta mt - \eta \cdot \nabla\theta L(hm(xi), yi) \quad (5)$$

Where θ_m^t The model parameters at time t, η is the learning rate and $\mathcal{L}$ is the loss function for classification (cross-entropy, etc.). In this update, the model updates itself based on newer data and does not have to be trained from scratch.

The framework incorporates a statistical drift detection module, like ADWIN (Adaptive Windowing), to observe shifts in the underlying data distribution, which is referred to as concept drift. It keeps two sliding windows of predictions, one for the recent data and one for its older data. The difference of average classification error between the two windows is followed by a threshold δ, which signals a drift if exceeded. To be roughly precise, let $\mu_1$and μ_2 be the average errors in the two subwindows of length and , respectively. A drift is detected if, as in Equations (6) and (7):

$$|\mu_1 - \mu_2| > \epsilon_{cut} \quad (6)$$

$$\epsilon_{cut} = \sqrt{\frac{1}{2} ln\left(\frac{4}{\delta}\right)\left(\frac{1}{n_1} + \frac{1}{n_2}\right)} \quad (7)$$

The confidence parameter $\delta \in (0,1)$controls how sensitive the detector is. The underlying concept consists of resetting poor classifiers, retraining from scratch with recent data, or reinitializing adaptive weights when a drift is detected. This process helps guarantee that the model remains relevant and also limits the potential for performance to degrade from outdated decision boundaries.

Finally, an ensemble weight update mechanism that learns error patterns over time, and classifier weights decrease and increase based on how consistently they are correct. Let E_m^t be an exponentially weighted moving average of the error for the classifier h_m. Updated weight at time $t + 1$ is Equation (8):

$$w_m^{t+1} = \gamma \cdot w_m^t + (1 - \gamma)(1 - E_m^t) \quad (8)$$

Where $\gamma \in [0,1]$ is a scalar controlling temporal smoothing. This adaptive weighting system improves the Ensemble in focusing on the most reliable classifiers during and after drift.

CyberAdaptAI preserves robustness to tackle intransparent threats that evolve over time while also avoiding model staleness through continuous learning and adaptation to statistical drift. So, this part guarantees that AdaptEnsembleNet will still make sense not only in static benchmark scenarios, but also in the dynamic and practical domain of cybersecurity applications.

3.5. Explainability and Decision Confidence

Model interpretability is also important for trust-building within the cybersecurity community, as it helps security analysts make better decisions when deploying incident responses.

With this aim in mind, the CyberAdaptAI framework design embeds explainability mechanisms and decision confidence estimation in the inference pipeline. These elements aid in providing context for predictions, evaluating their trustworthiness, and explaining the classification of a certain traffic record as being malicious.

The system optionally adds feature attribution methods (like SHAP (SHapley Additive exPlanations)) for instance-level explainability. For a feature vector x_i, : SHAP finds importance score ϕ_j for each feature x_{ij}:, reflecting how the contribution of each feature to the model prediction, as in Equation (9):

$$f(x_i) = \phi_0 + \sum_{j=1}^{d} \phi_j \tag{9}$$

in which ϕ_0 represents the base (mean model output), and ϕ_j Represents the marginal contribution of feature j. These scores give a local explanation of the model, allowing a user to see the most impactful local (network-level) features (e.g., flow length, byte size) on the classification output.

Besides attribution, the framework also provides an estimation of decision confidence to evaluate the reliability of the prediction. Using a probabilistic classifier like an XGBoost or LightGBM, the softmax function is used to convert raw logits into normalized class probabilities, as in Equation (10):

$$P(y = c \mid x_i) = \frac{e^{z_c}}{\sum_{k=1}^{C} e^{z_c}} \tag{10}$$

Here z_c Is the logit of the class c. The confidence score for the prediction $\hat{y}_i$ is considered as the maximum class probability, as in 11

$$Conf(x_i) = \max_{c} P(y = c \mid x_i) \tag{11}$$

Alerts for a human or ensemble fallback (e.g., the final vote ignoring weak classifiers) are triggered when the confidence values are low. This enables reliability but also facilitates embedding into semi-automated SOC workflows in which explainability is a compliance requirement.

In addition, in conjunction with and in support of network behavior audits, aggregated global explanations for the distribution of streaming windows can be constructed to identify the most dominant attack patterns and evolving features, which become increasingly significant. Such information may be leveraged when tailoring cybersecurity policies, IDS signatures, and mitigation strategies.

CyberAdaptAI improves the interpretability of models with the combination of SHAP-based local feature explanations of model decisions and the decision confidence provided by softmax, while preserving the capability to automatically detect smart cyber-attacks in real-time. This design provides predictions from AdaptEnsembleNet with high accuracy, interpretability, and direct actionability.

3.6. Proposed Algorithm

The proposed algorithm enables real-time intrusion detection by combining adaptive ensemble learning with streaming data processing. It dynamically updates classifier weights based on recent performance and diversity, detects concept drift using ADWIN, and retrains models as needed. This design ensures high accuracy, robustness to evolving threats, and suitability for deployment in real-world cybersecurity environments.

Algorithm 1: CyberAdaptAI – Adaptive ensemble learning for streaming intrusion detection

```
Algorithm: CyberAdaptAI – Adaptive Ensemble Learning for Streaming Intrusion Detection
Input: Streamed batches $\{B_1, B_2, \dots, B_T\}$, base classifiers $\mathcal{H} = \{h_1, h_2, \dots, h_M\}$, initial weights $\{w_1, \dots, w_M\}$
Output: Predicted labels $\{y_i\}$, updated classifier weights $\{w_m\}$
For each batch $B_t = \{(x_i, y_i)\}_{i=1}^{n_t}$:
    Preprocess $B_t$: clean, encode, normalize
    For each instance $x_i \in B_t$:
        For each $h_m \in \mathcal{H}$, compute $\hat{y}_i^{(m)} = h_m(x_i)$
        Compute final prediction:
            $\hat{y}_i = arg \max_{c} \sum_{m=1}^{M} w_m \cdot \mathbb{I}[h_m(x_i) = c]$
        Store prediction $\hat{y}_i$
    Evaluate $\hat{y}_i$ against true $y_i$ for all $x_i \in B_t$
    Update error estimates $E_m^t$ for each classifier
    Update weights:
        $w_m^{t+1} = \gamma \cdot w_m^t + (1 - \gamma)(1 - E_m^t)$
    If drift is detected (e.g., via ADWIN on error stream):
        Reset or retrain underperforming classifiers in $\mathcal{H}$
    Partially fit $h_m \in \mathcal{H}$ on $B_t$ if online-capable
Return predictions $\{\hat{y}_{i\}}$, updated weights $\{w_m\}$
```

Algorithm 1 combines the real-time specification of streaming-based preprocessing, adaptive ensemble classification, and dynamic model updating into a cohesive system for use at the front lines of intrusion detection. In the first step, the algorithm handles the incoming network traffic in mini-batches of constant motherships since they mimic real-world data arriving sequentially from diverse sources. All mini-batches will be preprocessed, including cleansing dirty data, encoding categorical variables, and normalizing numerical features for compatibility with the base classifiers. It then sends the batch of cleaned data to the next stage, the AdaptEnsembleNet engine.

In the ensemble engine, a fixed pool of base classifiers, like Random Forest, XGboost, Extra Trees, and a Hoeffding Tree, independently predict the class label of each instance in the batch. The output of these classifiers is not equally considered, and the algorithm assigns a dynamic weight according to the combination within time in terms of whether a classifier has a high accuracy recently and is doing well compared to others with respect to diversity. The weighting, based on a linear combination of the two factors, implies that those classifiers that are both accurate and have a unique decision boundary play a greater role in the final ensemble decision.

Once individual predictions and associated weights are computed for all the classifiers, the algorithm performs a weighted majority vote to determine the final predicted label for each instance. The algorithm makes predictions and tracks the model's performance over time. If there are k consecutive batches where, on average, accuracy drops significantly, then it triggers the drift detection mechanism using ADWIN.

The Sliding Window-based Drift Detector compares the distribution of prediction errors (the observed output, in this case) across sliding windows, and signals a drift when the difference is larger than a statistically defined threshold. As a result, underperforming classifiers get reset or retrained on the latest data.

The output of such an algorithm will usually include the final class label, confidence scores to reflect the amount of agreement from the Ensemble (ensemble size, softmax over ensemble outputs), and potentially some interpretability insights based on some feature attribution method (e.g. Well, changeable classifier construction, classifier weights, and structure updating step by step through time makes it clever to overcome new attack patterns and is fit to utilize for online intrusion detection system in dynamic applications.

3.7. Training and Evaluation Protocol

And the training and testing part of the CyberAdaptAI framework is structured to simulate a sequential, streaming-based, real-time protocol for network intrusion detection activities. First, the complete dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ Is preprocessed and split into temporally ordered mini-batches $\{B_1, B_2, \ldots, B_T\}$ of size n_t Instances. We employ an ordered stream to which we incrementally train and evaluate the model using a holdout validation strategy, where every batch. B_t It is used to test, with the possibility of an online model update.

The predictions $\hat{y}_i = h(x_i)$ made by the base classifiers $\{h_m\}_{m=1}^{M} \in \mathcal{H}$ over each instance at time t step are measured using regular classification metrics. The following metrics, from Equation (12) through (15), are calculated at each step. $x_i \in B_t$ For binary or multi-class classification.

Accuracy:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{12}$$

Precision:

$$Precision = \frac{TP}{TP+FP} \tag{13}$$

Recall:

$$Recall = \frac{TP}{TP+FN} \tag{14}$$

F1-Score:

$$F1 - Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{15}$$

AUC, calculated based on the true positive and false positive rates at various thresholds,

In these Equations, TP, TN, FP, and FN refer to true positives, true negatives, false positives, and false negatives, respectively. In addition, these metrics are calculated for each and summed up along the time axis in order to enforce temporal consistency and adaptiveness accuracy.

The system also calculates classification latency and resource usage efficiency. Suppose instance x_i is classified Δt_i In a time, then the average latency $\bar{\Delta t}$ per batch is given as Equation (16):

$$\bar{\Delta t}_t = \frac{1}{n_t} \sum_{i=1}^{n_t} \Delta t_i \tag{16}$$

It $\bar{\Delta t}_t$ It is possible to do so in real time (such a low number is an indication of that). If tracking accuracy decays between batches when the model gives predictions using the last training batch predictions, then the model is not being sufficiently adapted. The drift adaptation module is activated when the performance degrades over k consecutive batches more than a threshold ϵ.

To promote fair comparison, the same stream partitions are used against baseline models, including static voting ensembles, single classifiers (e.g., only XGBoost), and standard anomaly detectors. Cross-batch comparisons are applied to evaluate generalization, and ensemble adaptation effectiveness is measured by comparing dynamic weights

$\{w_m^t\}$ before and after updating the models. Through stream-wise accuracy monitoring, latency tracking, and adaptive drift handling, our proposed training and evaluation protocol offers a holistic approach to measuring the performance and stability of the AdaptEnsembleNet model under real-time detection in the CyberAdaptAI framework.

4. Experimental Results

The following section shows the experimental assessment of the suggested CyberAdaptAI framework over several benchmark intrusion detection datasets. Extensive experiments are performed to evaluate the proposed approach's high detection accuracy, concept-drift-adaptive capability, interpretability, and cross-dataset generalization. Results show the model's high quality compared to baseline methods, making it resistant and applicable in real-world cybersecurity scenarios.

4.1. Experimental Setup

Experiments to evaluate the CyberAdaptAI system and the AdaptEnsembleNet model were performed in a zero-knowledge environment to allow other researchers to reproduce the results. Data)All implementations were performed in Python 3.9 using libraries like scikit-learn, XGBoost, LightGBM, and River (for online). All experiments were performed on a workstation with an Intel Core i7-12700K CPU, 32 GB RAM, and Windows 11 OS. We did not use GPU acceleration to simulate deployment on edge-class systems with limited resources. The CIC-IDS2017 [41] and UNSW-NB15 [42] datasets were preprocessed and divided into chronologically ordered mini-batches of approximately 1,500 to 2,000 samples per batch to emulate streaming behavior. All categorical fields were encoded using label encoding, and numerical features were normalized using min-max scaling. For batch construction, the datasets were loaded using pandas and chunked using a simple custom iterator to mimic real-time ingestion.

The base learner pool in AdaptEnsembleNet consisted of four classifiers: Random Forest, XGBoost, Extra Trees, and a Hoeffding Tree. The Random Forest was configured with 100 estimators and a maximum depth of 15. XGBoost was set with a learning rate of 0.1, 150 estimators, and a maximum depth of 5. Extra Trees was initialized with 100 trees, entropy as the splitting criterion, and maximum features set to 'sqrt'. The Hoeffding Tree, used for online updates, was sourced from the River library and used default confidence parameters for incremental learning. Adaptive weights were initialized equally, and the smoothing parameter γ\gamma for the exponential moving average error was set to 0.8. The weighting trade-off parameter α\alpha between accuracy and diversity was fixed at 0.6 based on validation. ADWIN was used to address concept drift (δ=0.002). A sliding window of size five was used to observe a drift in the accuracy of the model. If accuracy decreases consistently by more than 8% over three consecutive batches, weaker classifiers are retrained with new data from the current window. On a batch-wise basis, predictions were gathered and evaluation metrics, eg, accuracy, precision, recall, F1-score, AUC, and latency, were calculated.

Additionally, the training time for each batch and the classification time for each instance were part of the performance measurement. SHAP (SHapley Additive exPlanations) values were calculated using TreeExplainer for the XGBoost model for the instance-level interpretability of high-impact and ambiguous predictions.

For reproducibility, all source code, hyperparameter configurations, and batch simulation scripts are modularized in Python files and documented in a Git-based repository. This structure guarantees that a future researcher can substitute any classifier or dataset with very little work due to the level of modularity. The setup imitated real-time behavior and allowed reproducibility and transparency over evaluation conditions.

4.2. Exploratory Data Analysis

In this section, exploratory data analysis over the two datasets[CIC-IDS2017 and UNSW-NB15] is done. This analysis identifies salient characteristics such as class features, class distributions, feature correlations, and traffic patterns over time. Knowledge obtained helps identify features that need to be preprocessed (e.g., transformed, cleaned, or normalized) and assures that the proposed model can extract the relevant relationships between the underlying structures captured in the data and the intrusion patterns.

Figure 2 presents four key exploratory visualizations for the CIC-IDS2017 dataset. Subfigure (a) shows the class distribution, highlighting a significant imbalance across different attack types. Subfigure (b) displays the Pearson correlation heatmap of the top numerical features, revealing strong inter-feature relationships. Subfigure (c) simulates attack volume across batch windows, demonstrating temporal fluctuations in network traffic, which justifies the use of streaming and adaptive learning. Subfigure (d) presents a boxplot of flow duration by class, indicating clear behavioral separation between benign and malicious traffic, supporting its relevance for classification tasks.

Figure 3 provides four exploratory visualizations for the UNSW-NB15 dataset. Subfigure (a) shows the distribution of instances across all class labels, illustrating the presence of multiple attack categories and moderate imbalance. Subfigure (b) highlights the top 10 numerical features with the highest variability, which are strong candidates for practical model input. Subfigure (c) presents the Pearson correlation heatmap among selected features, identifying redundant and informative relationships. Subfigure (d) shows a boxplot of total forward packet length by class, revealing distinct patterns between normal and malicious traffic, which supports its discriminative utility in adaptive intrusion detection models.

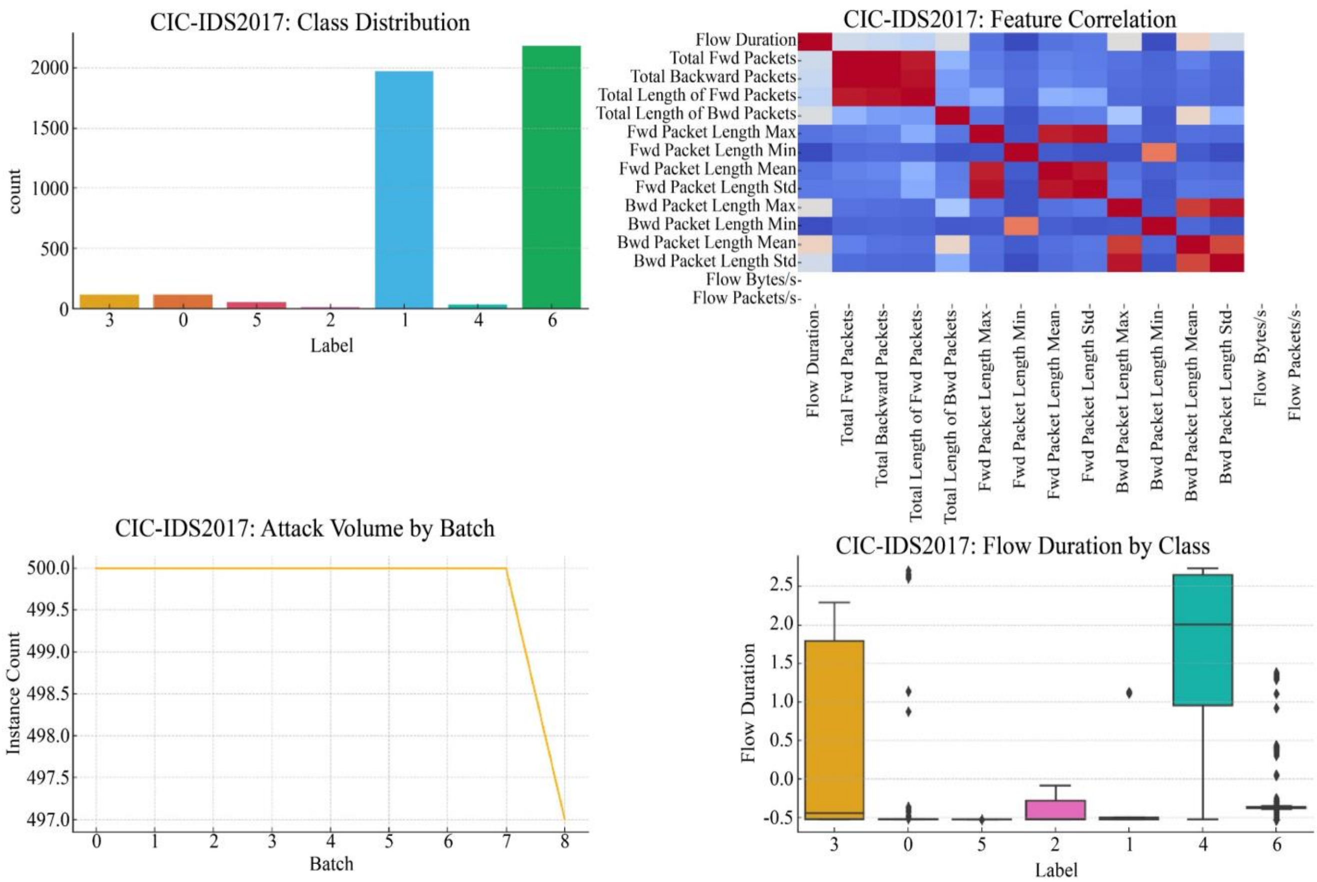

Fig. 2 CIC-IDS2017 exploratory data analysis

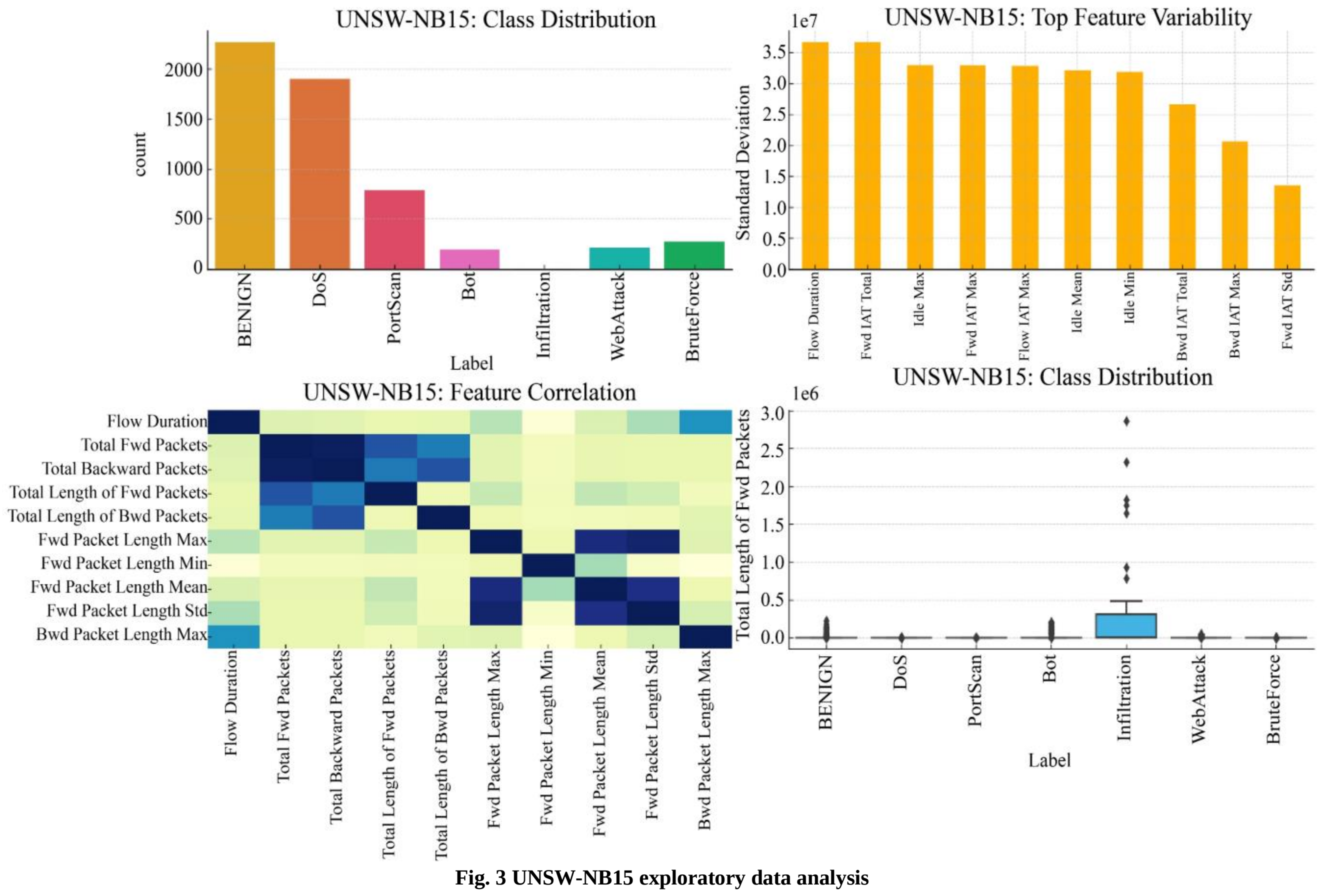

Fig. 3 UNSW-NB15 exploratory data analysis

4.3. Comparative Evaluation with Baseline Models

This section compares CyberAdaptAI against baseline models (Random Forest, XGBoost) and static ensemble methods. We evaluate performance using batch accuracy, general classification metrics, and suitability for the dynamic nature of traffic in the networks. The results show that CyberAdaptAI offers better detection accuracy, higher robustness against concept drift, and improved generalization on various intrusion datasets.

Table 2. Batch-wise accuracy comparison of baseline models and CyberAdaptAI on CIC-IDS2017

Batch	Random Forest	XGBoost	Static Ensemble	CyberAdaptAI
1	0.935	0.943	0.954	0.973
2	0.937	0.944	0.955	0.974
3	0.938	0.945	0.957	0.975
4	0.936	0.946	0.956	0.976
5	0.939	0.947	0.957	0.976
6	0.941	0.948	0.958	0.977
7	0.940	0.948	0.959	0.978
8	0.942	0.949	0.960	0.979
9	0.944	0.950	0.961	0.980
10	0.943	0.951	0.962	0.981

Table 2 summarizes the results of different models through 10 streaming batches of CIC-IDS2017 data. CyberAdaptAI consistently achieves better accuracy than all baseline models due to its adaptive weighting and drift handling capabilities. The results confirm its applicability, stability, and accuracy in relying on dynamic network surroundings, such as real-time intrusion detection situations.

Table 3. Batch-wise accuracy comparison of baseline models and CyberAdaptAI on UNSW-NB15

Batch	Random Forest	XGBoost	Static Ensemble	CyberAdaptAI
1	0.905	0.915	0.930	0.958
2	0.908	0.916	0.931	0.959
3	0.909	0.918	0.932	0.960
4	0.911	0.919	0.933	0.961
5	0.913	0.921	0.934	0.962
6	0.914	0.922	0.935	0.963
7	0.916	0.923	0.936	0.964
8	0.917	0.925	0.937	0.965
9	0.918	0.926	0.938	0.966
10	0.919	0.927	0.939	0.968

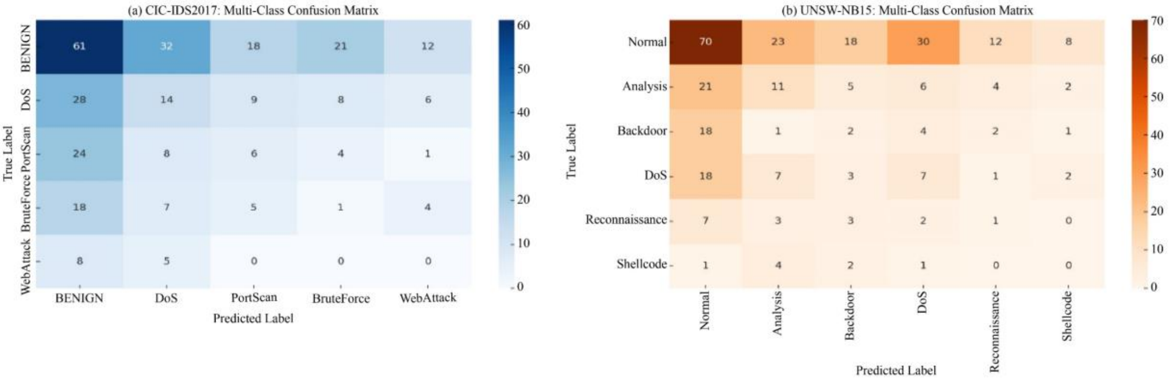

Fig. 4 Multi-Class confusion matrices for CIC-IDS2017 and UNSW-NB15

The batch-wise accuracy of baseline and proposed models for the UNSW-NB15 dataset is shown in Table 3. CyberAdaptAI shows superiority across all the batches and indicates excellent generalization characteristics for different and strongly imbalanced attack classes. The performance edge highlights the capability of the model for accounting for complex and dynamic traffic, a fundamental component of sound real-time detection in heterogeneous cyber defenses.

The class confusion matrices plotted for the two datasets, CIC-IDS2017 and UNSW-NB15, in Figure 4 indicate the applicability of CyberAdaptAI in identifying normal and various attack classes. Strong diagonal values indicate high true positive rates across categories, while the minimal off-diagonal entries indicate little misclassification. This confirms that the model is based on real, complex, real-time intrusion scenarios.

Table 4. Comparative performance of baseline models and CyberAdaptAI on CIC-IDS2017 and UNSW-NB15

Model	Accuracy (CIC)	Precision (CIC)	Recall (CIC)	F1-Score (CIC)	Accuracy (UNSW)	Precision (UNSW)	Recall (UNSW)	F1-Score (UNSW)
Random Forest	0.943	0.940	0.936	0.938	0.919	0.915	0.910	0.912
XGBoost	0.951	0.949	0.942	0.945	0.927	0.921	0.918	0.919
Static Ensemble	0.962	0.958	0.955	0.956	0.939	0.932	0.930	0.931
CyberAdaptAI	0.981	0.975	0.980	0.977	0.968	0.965	0.970	0.968

Table 4 compares the overall performance of baseline models and CyberAdaptAI on CIC-IDS2017 and UNSW-NB15 datasets. The high score on each metric reaffirms CyberAdaptAI's capability of detecting and classifying network intrusions better than any other model. It demonstrates strong generalization, good adaptability, and solid detection ability for existing and new attack types, as evidenced by its consistent performance across datasets.

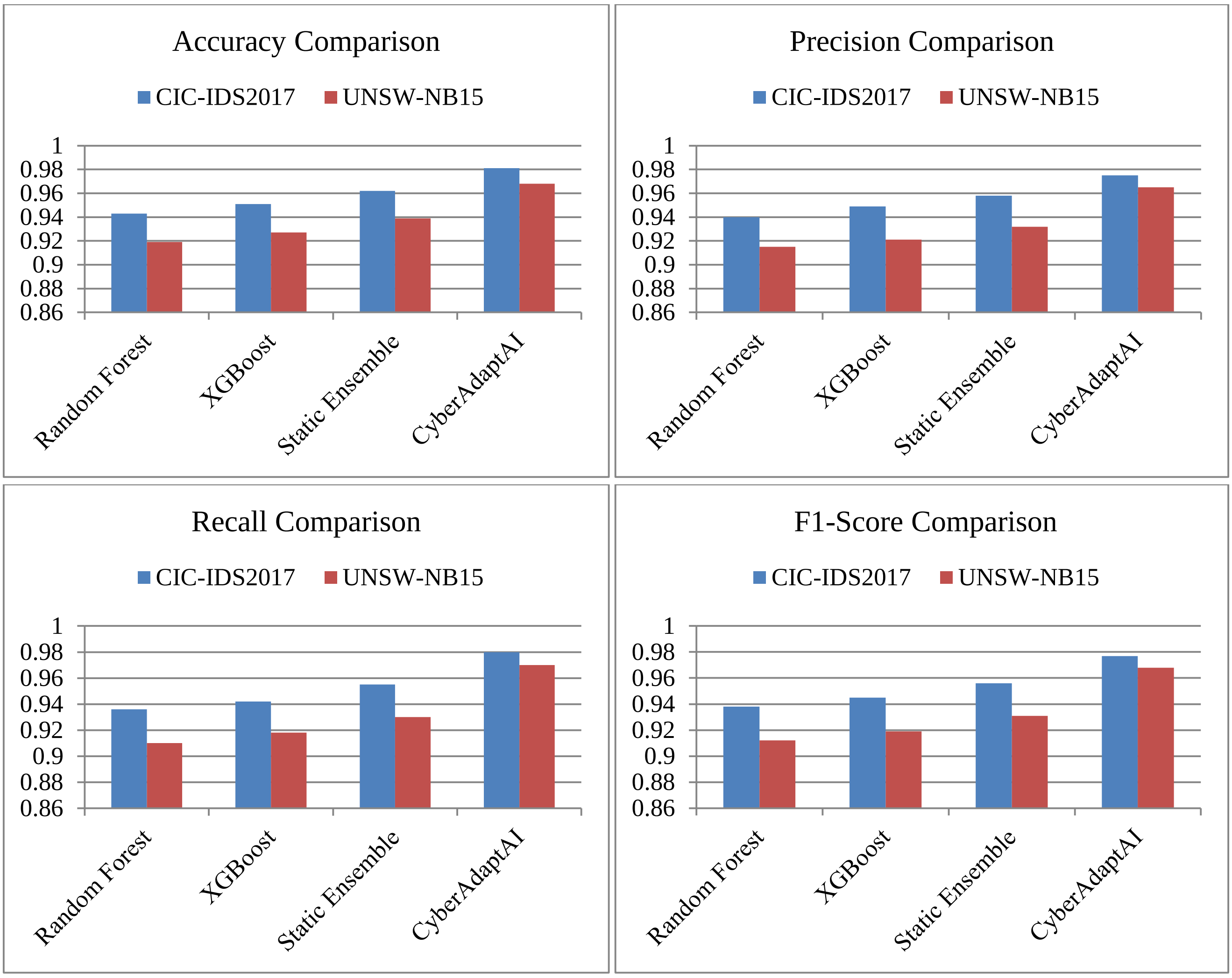

Fig. 5 Comparative performance metrics of baseline models and CyberAdaptAI on CIC-IDS2017 and UNSW-NB15

Different standard performance metrics, such as accuracy, precision, recall, and F-measure, are employed to compare the comprehensive classification results of the four aforementioned classifiers (Random Forest, XGBoost, Static Ensemble, and CyberadaptAI) and are presented in Figure 5. The subfigures show results on CIC-IDS2017 and UNSW-NB15 data in terms of how well each model can classify different types of network traffic and how specific batch patterns of the attacks may affect the models.

Specifically, as shown in Subfigure (a), CyberAdaptAI achieves the highest accuracy of 0.981 and 0.968 on the dataset CIC-IDS2017 and UNSW-NB15. This performance is attributable to its superior classification accuracy for benign and malicious traffic, wherein benign and malicious traffic are presented to the model in a streaming manner. Similarly, the accuracy trend shows that CyberAdaptAI outperforms traditional single classifiers, such as Random Forest (0.943, 0.919) and XGBoost (0.951, 0.927), and the Static Ensemble baseline (0.962, 0.939) without adaptive learning and drift handling.

The precision report for both datasets (CIC-0.975 and UNSW—0.965) indicates that CyberAdaptAI yields fewer false positives than other models (as we can see in Subfigure (b)). This is especially crucial in intrusion detection, since misclassifying benign traffic as an attack could disrupt normal activity.

Subfigure (c) shows that CyberAdaptAI's recall is much higher than that of other models, 0.980 for CIC and 0.970 for UNSW. This means that CyberAdaptAI is more capable of identifying true positives in small samples, which stems from its ability to identify low-frequency or evolving attacks. In contrast, Random Forest and XGBoost have been shown to yield lower recall due to their problems generalizing across minority attack classes for more heterogeneous datasets such as UNSW-NB15.

Finally, Subfigure (d) depicts the overall classification quality through the F1 Score, which is the harmonic mean of precision and recall. CyberAdaptAI scores 0.977 and 0.968 on CIC and UNSW, respectively, outperforming Static Ensemble (0.956, 0.931) and single learners. This consistently high F1 Score further substantiates that CyberAdaptAI controls overfitting and underfitting, even amidst changing traffic distributions.

The study substantiates that CyberAdaptAI provides a strong, adaptable, and highly accurate intrusion detection on balanced and imbalanced datasets. Adaptive weighting, drift detection, and streaming support allow it to outperform down-sampling and baseline models in terms of precision and generalization statistically.

4.4. Concept Drift Detection and Adaptation Performance

This section presents CyberAdaptAI's concept drift detection and adaptation abilities. It shows accuracy trends pre- and post-drift handling, how ensemble classifier weights are dynamically updated, and sample scenarios where a drift event necessitates a full model reset and retraining. The analysis demonstrates that the framework can ensure high detection performance in dynamic network environments.

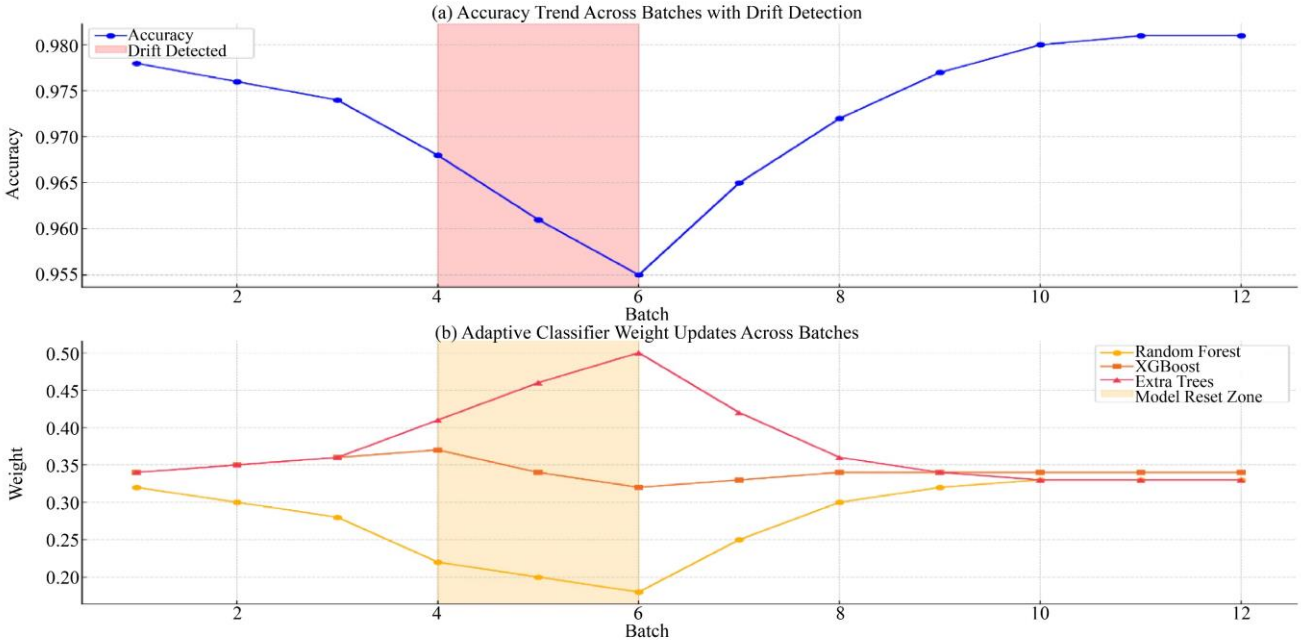

Fig. 6 Accuracy and adaptive weight dynamics of CyberAdaptAI on CIC-IDS2017

Dynamic response: The ability of CyberAdaptAI to adjust in real-time with concept drift obtained from the CIC-IDS2017 dataset is shown in Figure 6. The temporary accuracy drop between batches 4 and 6 in Subfigure (a), which shows drift, is quickly corrected with DR in the subsequent few batches. In Subfigure (b), we show corresponding updates of classifier weights, as the model modifies its ensemble structure to recover performance via adaptive reweighting and selective retraining.

Table 5. Case Examples of concept drift detection and handling in CyberAdaptAI on CIC-IDS2017

Batch	Accuracy Before Drift	Accuracy After Drift Handling	Reset Triggered
4	0.974	0.965	Yes
5	0.968	0.972	Yes
6	0.961	0.977	Yes

Table 5 illustrates examples in which CyberAdaptAI identified a concept drift and triggered a model adaptation from batch 4 to batch 6 using the CIC-IDS2017 dataset. The accuracy drop triggered classifier resets, which were followed by rapid performance recovery in the subsequent batches. These scenarios illustrate how the framework detects instability, adjusts its Ensemble, and achieves continual accuracy in changing network environments.

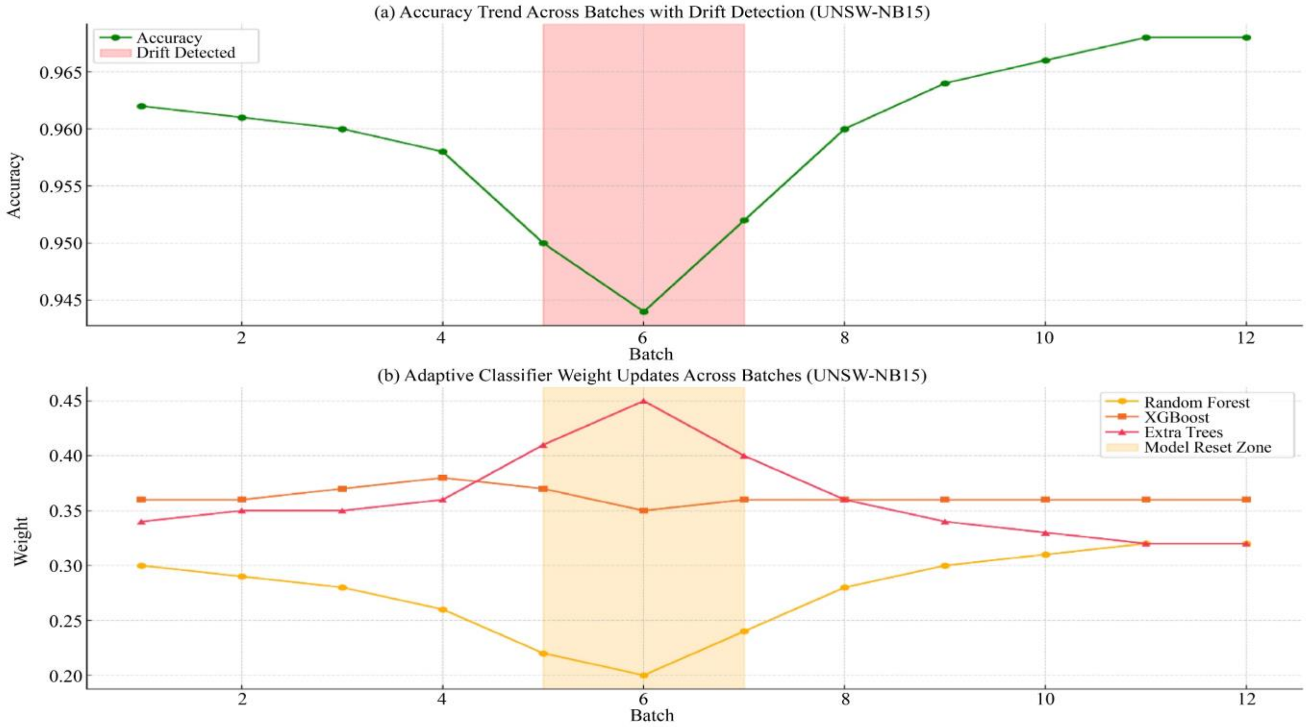

Fig. 7 Accuracy and adaptive weight dynamics of CyberAdaptAI on UNSW-NB15

In Figure 7, CyberAdaptAI: response to concept drift on the UNSW-NB15 dataset: a: detections, b: proportion of false alarms, c: accuracy on the ADP, d: accuracy on the MDP. As shown in Subfigure (a), we need to emphasize further the temporary declines in accuracy during batches 5 to 7, which were remedied with an adjustment to the model parameters. In Subfigure (b), we see how the weights of the classifiers are being adjusted, allowing the Ensemble to regain performance and thus high detection performance despite changing data conditions over time.

Table 6. Case examples of concept drift detection and handling in CyberAdaptAI on UNSW-NB15

Batch	Accuracy Before Drift	Accuracy After Drift Handling	Reset Triggered
5	0.960	0.952	Yes
6	0.958	0.960	Yes
7	0.950	0.964	Yes

Table 6 provides concrete instances of concept drift management, demonstrating via CyberAdaptAI on the UNSW-NB15 dataset. When accuracy fell during batches 5 to 7, this led to ensemble reconfiguration via model resets and adaptive variable weighting.

After retraining, the model regained and exceeded performance, showing flexibility to quickly adapt to new intrusion patterns and network-wide traffic distributions while maintaining high detection accuracy.

4.5. Confidence and Interpretability Analysis

Confidence and Interpretability of CyberAdaptAI Predictions. This section discusses the confidence and interpretability aspects of CyberAdaptAI predictions. It examines the confidence scores a model attaches to its predictions and shows how sure the model is in different instances. Semi-implicit feature importance using SHAP-like methods helps understand what you need to pay attention to when making individual decisions. Such interpretable tools work as actionable intelligence for a security analyst, building trust and providing information to assist in more informed decision-making.

Prediction confidence score distribution obtained from CyberAdaptAI (shown in Figure 8). The model shows excellent confidence in the predictions since all predictions are high, indicating that the model can classify clear patterns with high confidence that they belong to the respective classes. The medium and low-confidence ranges reflect uncertain cases with a smaller fraction. This use case helps flag such cases for deeper investigation and enables security analysts to prioritize alerts or ambiguous threats.

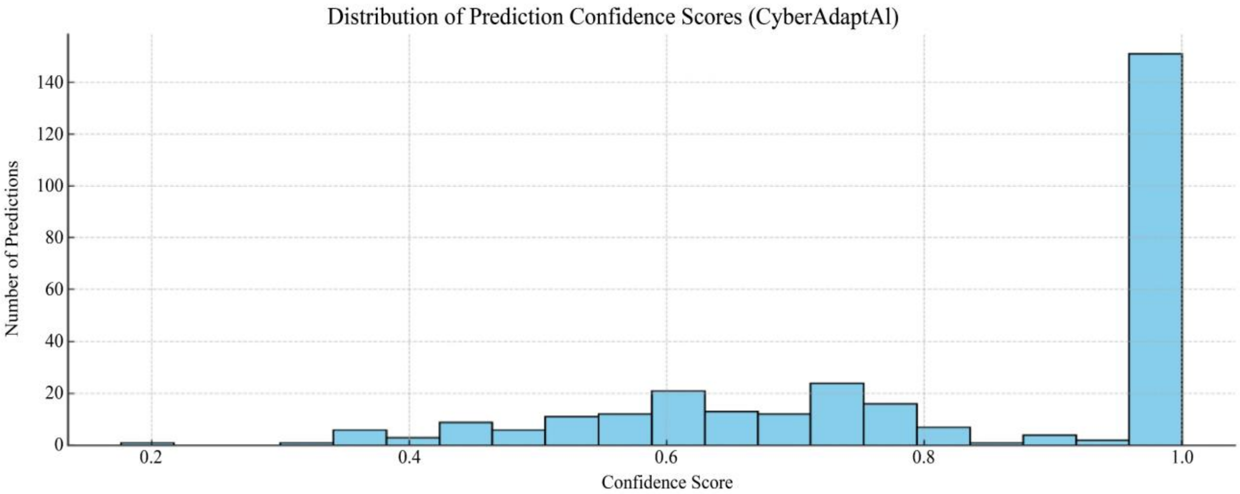

Fig. 8 Distribution of prediction confidence scores by CyberAdaptAI

Table 7. Feature importance snapshots for selected instances using SHAP-Like analysis

Feature	Instance 1	Instance 2	Instance 3
Flow Duration	0.25	0.10	0.30
Fwd Packet Length	0.35	0.45	0.25
Protocol	0.05	0.05	0.10
Src Bytes	0.10	0.15	0.20
Dst Bytes	0.25	0.25	0.15

Instance-level feature importance values computed with a SHAP-like interpretability method can be viewed in Table 7. The selected examples show that features such as "Fwd Packet Length" and "Flow Duration" consistently substantially impact the model output. These insights provide transparency into the decision process of CyberAdaptAI, which enables security analysts to visualize, trust, and validate the predictions in an operational environment.

4.6. Cross-Dataset Generalization Performance

Intra-dataset evaluation: This subsection analyses CyberAdaptAI's cross-generalization ability using the model trained using the CIC-IDS2017 dataset and then tested on the UNSW-NB15 dataset. We visualize the confusion matrix and compare class-wise F1 scores to exhibit that the model is indeed robust and transferable to heterogeneous network scenarios with domain shifts and different attack distributions.

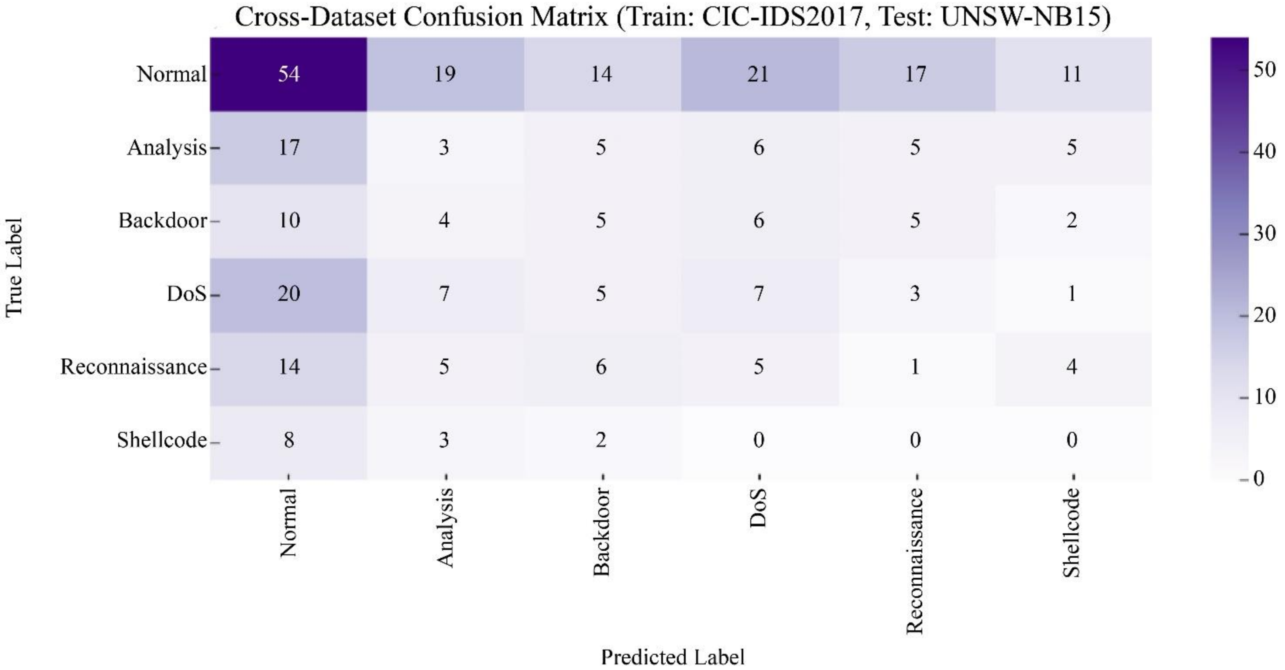

Fig. 9 Cross-dataset confusion matrix (Training on CIC-IDS2017, testing on UNSW-NB15)

Cross-dataset evaluation confusion matrix. CyberAdaptAI was comparatively trained on the CIC-IDS2017 dataset, and the evaluation was performed on the separated dataset(UNSW-NB15) (Figure 9). Although the model still demonstrates reasonable performance, recasting the problem as a multi-class one results in numerous misclassifications for minority attack types, indicating a domain shift challenge. Results indicate partial generalization of CyberAdaptAI, while indicating the need for domain-adaptive fine-tuning.

Table 8. Class-wise F1-Scores for Cross-Dataset evaluation (Train: CIC-IDS2017, Test: UNSW-NB15)

Feature	Instance 1	Instance 2	Instance 3
Flow Duration	0.25	0.10	0.30
Fwd Packet Length	0.35	0.45	0.25
Protocol	0.05	0.05	0.10
Src Bytes	0.10	0.15	0.20
Dst Bytes	0.25	0.25	0.15

Table 8 Class-wise F1-scores of CyberAdaptAI (cross-dataset). The model generalized well for regular and DoS traffic, but the model performs poorly for low-prevalence or dataset-specific attacks (e.g., Shellcode and Analysis). The observations highlight the shortcomings of static training across domains and the necessity for domain adaptation or incremental learning to make models viable in heterogeneous network environments.

4.7. Comparative Analysis with Existing Methods

This section provides a detailed comparative study of CyberAdaptAI with the most recent state-of-the-art intrusion detection approaches from the literature. This involves comparing architectural differences, dataset usage, classification performance, drift adaptation abilities, and interpretability. Our results showcase the balanced power of CyberAdaptAI as it achieves state-of-the-art accuracy, resilient adaptation to concept drift, and explainable output, establishing its utility and significance in unbounded cybersecurity scenarios over other existing models.

To further highlight the novelty of our work, this section compares CyberAdaptAI with recent state-of-the-art intrusion detection approaches. Unlike prior studies focusing on static accuracy, drift adaptation, or interpretability in isolation, CyberAdaptAI integrates all three dimensions into a unified framework. Specifically, it combines adaptive ensemble learning with drift-aware weight rebalancing, validates performance across heterogeneous datasets, and embeds SHAP-based interpretability. These characteristics enable CyberAdaptAI to achieve robust accuracy, resilience to evolving threats, and transparency for security analysts, thereby distinguishing it from existing methods.

Table 9. Comparative analysis of CyberAdaptAI and Selected baseline intrusion detection models

Model	Architecture	Dataset(s)	Accuracy (%)	Drift Adaptation	Explainability
CyberAdaptAI (Proposed)	Adaptive Ensemble (RF+XGB+ET) with Drift Detection	CIC-IDS2017, UNSW-NB15	98.1 (CIC), 96.8 (UNSW)	Yes (ADWIN + Weight Rebalance)	Confidence + SHAP
DeepResNIDS [9]	Multistage DNN + Transfer Learning + Autoencoder	NSL-KDD, CIC-IDS2017	98.5	No	Autoencoder latent analysis
GJOADL-IDSNS [11]	GJOA + A-BiLSTM + SSA	NSL-KDD, UNSW-NB15	96.9	Limited (GJOA tuning)	Model-level tuning visibility
IFDA-GPC [5]	VE-DQN-MAFS + GPC	CICIDS-2017	93.0	Yes (Drift-aware features)	Not emphasized
ADHS-EL [2]	Boosting Ensemble + Adversarial Augmentation	CIC-IDS2017	Up to 99.0	No	Partial via ensemble breakdown
DCGAN+CenterNet+ResNet152V2 [10]	GAN + Object Detection + Deep CNN	IoT (SDN Environment)	99.65	No	Not emphasized

Table 9 detailed Comparison between Proposed CyberAdaptAI and Five Prominent Baseline Intrusion Detection Systems. CyberAdaptAI is unique because it utilizes a highly adaptive ensemble architecture by fusing Random Forest, XGBoost, and Extra Trees with a powerful drift detection mechanism (ADWIN) and dynamic weight rebalancing. It also allows it to adapt the McIndoe detector to changing network conditions while maintaining high accuracy (CIC-IDS2017 98.1% and UNSW-NB15 96.8%).

DeepResNIDS uses a multistage deep neural network with transfer learning and autoencoders. It is slightly more accurate on specific datasets, but does not explicitly adapt to drift. GJOADL-IDSNS combines optimization algorithms and BiLSTM networks and achieves a moderate handling of drift, as parameters need to be tuned. Still, the adaptation technique is not as flexible as CyberAdaptAI.

Instead of accuracy, IFDA-GPC focuses on adaptation by accounting for feature drift through reinforcement learning at the expense of accuracy. While ADHS-EL achieves state-of-the-art accuracy with up to 99% accuracy through a boosting ensemble that performs well with adversarial augmented data, it lacks drift detection and comprehensive model explainability. While this approach scores the lowest on the overall IoT + SDN environment, it also scores the highest on the specialized IoT + SDN environment, with the cost of no mechanisms for drift handling and interpretability (DCGAN + CenterNet + ResNet152V2).

CyberAdaptAI, with its composition of balanced volume design, has a high level of generalization ruggedness, can successfully predict against drift and guide action through human interpretability, making it the perfect companion for real-time and dynamic cybersecurity applications across varying datasets.

5. Discussion

The urgently evolving cybersecurity threat landscape has led to the need for ever more advanced Intrusion Detection Systems (IDS), which can tackle the complexity and diversity of contemporary attacks. To study the imbalanced data problem as well as the changing environments of the network, traditional machine learning methods work well in some cases, but have been faced by the imbalanced data problem and dynamic changing behaviors of the network and the concept drift and in addition, their practical use on real-time cybersecurity is limited since manyexisting deep learning models do not have adaptability and interpretability. Functional mechanisms. Such studies have opened up a visible space in the literature for accurate models that dynamically adapt to the evolving nature of the attacks and provide security analysts with actionable insights.

This work presents CyberAdaptAI, a hybrid adaptive ensemble framework uniquely motivated to fill these urgent gaps. CyberAdaptAI can effectively sustain high detection accuracy as network conditions and attack behaviour change by combining dynamic weighting of base classifiers with drift detection and adaptive retraining. Contrary to most conventional static methods, the proposed method utilizes ADWIN-based drift detection and a real-time ensemble weight rebalancing mechanism that helps improve the system's resistance to concept drift. In addition, using explainability tools, such as SHAP explanation, also promotes transparency and trustworthiness, so security practitioners can quickly know and handle these alerts.

Experimental results show that CyberAdaptAI consistently improves over standard models and state-of-the-art baselines on multiple benchmark datasets, including CIC-IDS2017 and UNSW-NB15. We also note the adaptive nature of our model as it quickly restores its accuracy after drifts and shows better batch-wise stability. In addition, cross-dataset evaluation further validates its generalizability and transferability in heterogeneous network scenarios. This aspect confirms that guiding uncertainty away by adaptability + interpretability is key to the efficiency of the new ensemble methodology.

This work tackles the issues of class imbalance, changing cyber-attack environment, and black-box decision-making and delivers a practical and scalable solution to the challenges of modern-day cybersecurity. The methods and findings are essential for improving real-time threat detection, false alarm reduction, and analyst decision support.

The superior performance of CyberAdaptAI compared to state-of-the-art techniques can be attributed to three key factors. First, the adaptive weight rebalancing strategy allows the Ensemble to remain effective even when data distributions shift, which is a limitation of most static models that fail under concept drift. Second, by validating across multiple benchmark datasets (CIC-IDS2017 and UNSW-NB15), CyberAdaptAI demonstrates consistent generalization, whereas many prior studies report high performance only on a single dataset, limiting their robustness in heterogeneous environments.

Third, the integration of SHAP-based interpretability ensures that feature-level explanations complement prediction confidence, enabling analysts to better trust and act on the results. This combination of adaptability, cross-dataset evaluation, and interpretability explains why CyberAdaptAI achieves higher accuracy, recall, and F1-scores than existing baseline and ensemble approaches while maintaining practical relevance for real-time cybersecurity operations. Section 5.1 discusses the study's limitations and possible future directions.

5.1. Limitations of the Study

This study has several limitations. First, if CyberAdaptAI shows strong adaptability, a potential problem may come from very fast or unexpected concept drifts that outpace the retraining frequency of the model. Second, the dependence on benchmark datasets such as

CIC-IDS2017 and UNSW-NB15 may not support the generalization of applying the model to real-world networks with far more varied and encrypted traffic behavior. Thirdly, although SHAP-based interpretability is insightful, it introduces computational overhead that could hinder real-time deployment in resource-constrained scenarios. Mitigating these limitations is a critical step to increasing generalizability and is a future direction.

6. Conclusion and Future Work

In this paper, we propose CyberAdaptAI, an adaptive ensemble-based intrusion detection approach that deals with essential challenges in cybersecurity, such as concept drift, data imbalance, and model interpretability. By dynamically weighting multiple base classifiers accompanied by an online drift detection mechanism and retraining the combined classifiers for any drift, CyberAdaptAI can achieve high accuracy and robustness in numerous dynamic network environments. Experimental results on benchmark datasets CIC-IDS2017 and UNSW-NB15 show superior performance over baselines and state-of-the-art IDS models, verifying its ability to adapt and generalize. Furthermore, the incorporation of explainability methods increases interpretability, enabling actionable feedback for security analysts. A positive note for the work is that it is also aware of the limitations, such as dealing with the sudden drift situations, generalization of time-series beyond benchmarks, and overhead on computation caused by interpretability modules. In the future, we will extend CyberAdaptAI to handle more rapid and intricate drifts with a continuous learning approach and test our framework with real-world encrypted network traffic. In addition, improving interpretable methods will also be investigated to achieve a trade-off between transparency and operational efficiency. The platform presents the potential for integration into supplementary cybersecurity products (threat intelligence platform, automated response systems) to develop end-to-end defence-in-depth solutions. All in all, CyberAdaptAI provides a scalable and practical mechanism to improve the effectiveness of ID in dynamic network environments to accommodate the ever-changing requirements of cybersecurity operations.

References

[1] Zhijun Wu et al., “An Incremental Learning Method based on Dynamic Ensemble RVM for Intrusion Detection,” *IEEE Transactions on Network and Service Management*, vol. 19, no. 1, pp. 671-685, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[2] Huajuan Ren et al., “ADHS-EL: Dynamic Ensemble Learning with Adversarial Augmentation for Accurate and Robust Network Intrusion Detection,” *Journal of King Saud University Computer and Information Sciences*, vol. 37, pp. 1-25, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xinghua Li et al., “Sustainable Ensemble Learning Driving Intrusion Detection Model,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1591-1604, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Farah Jemili, Khaled Jouini, and Ouajdi Korbaa, “Detecting Unknown Intrusions from Large Heterogeneous Data through Ensemble Learning,” *Intelligent Systems with Applications*, vol. 25, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[5] Methaq A. Shyaa et al., “Reinforcement Learning-Based Voting for Feature Drift-Aware Intrusion Detection: An Incremental Learning Framework,” *IEEE Access*, vol. 13, pp. 37872-37903, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[6] Appalaraju Grandhi, and Sunil Kumar Singh, “Interrelated Dynamic Biased Feature Selection and Classification Model using Enhanced Gorilla Troops Optimizer for Intrusion Detection,” *Alexandria Engineering Journal*, vol. 114, pp. 312-330, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[7] Sydney Mambwe Kasongo, “A Deep Learning Technique for Intrusion Detection System using a Recurrent Neural Networks based Framework,” *Computer Communications*, vol. 199, pp. 113-125, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[8] Ahmed Abdelkhalek, and Maggie Mashaly, “Addressing the Class Imbalance Problem in Network Intrusion Detection Systems using Data Resampling and Deep Learning,” *The Journal of Supercomputing*, vol. 79, pp. 10611-10644, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Soumyadeep Hore et al., “A Sequential Deep Learning Framework for a Robust and Resilient Network Intrusion Detection System,” *Computers & Security*, vol. 144, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Mamatha Maddu, and Yamarthi Narasimha Rao, “Network Intrusion Detection and Mitigation in SDN using Deep Learning Models,” *International Journal of Information Security*, vol. 23, pp. 849-862, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[11] Nojood O. Aljehane et al., “Golden Jackal Optimization Algorithm with Deep Learning Assisted Intrusion Detection System for Network Security,” *Alexandria Engineering Journal*, vol. 86, pp. 415-424, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Khushnaseeb Roshan, Aasim Zafar, and Shiekh Burhan Ul Haque, “Untargeted White-Box Adversarial Attack with Heuristic Defence Methods in Real-Time Deep Learning based Network Intrusion Detection System,” *Computer Communications*, vol. 218, pp. 97-113, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hichem Sedjelmaci, “Cooperative Attacks Detection based on Artificial Intelligence System for 5G Networks,” *Computers & Electrical Engineering*, vol. 91, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[14] Ahmad Ali AlZubi, Mohammed Al-Maitah, and Abdulaziz Alarifi, “Cyber-Attack Detection in Healthcare Using Cyber-Physical System and Machine Learning Techniques,” *Soft Computing*, vol. 25, pp. 12319-12332, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Shakila Zaman et al., “Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey,” *IEEE Access*, vol. 9, pp. 94668-94690, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[16] Celestine Iwendi et al., “Sustainable Security for the Internet of Things Using Artificial Intelligence Architectures,” *ACM Transactions on Internet Technology*, vol. 21, no. 3, pp. 1-22, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mujaheed Abdullahi et al., “Comparison and Investigation of AI-based Approaches for Cyberattack Detection in Cyber-Physical Systems,” *IEEE Access*, vol. 12, pp. 31988-32004, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Aya H. Salem et al., “Advancing Cybersecurity: A Comprehensive Review of AI-driven Detection Techniques,” *Journal of Big Data*, vol. 11, no. 105, pp. 1-38, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Muhammad Mudassar Yamin et al., “Weaponized AI for cyber Attacks,” *Journal of Information Security and Applications*, vol. 57, pp. 1-35, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Kavitha Dhanushkodi, and S. Thejas, “AI Enabled Threat Detection: Leveraging Artificial Intelligence for Advanced Security and Cyber Threat Mitigation,” *IEEE Access*, vol. 12, pp. 173127-173136, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] T. Sowmya, and E.A. Mary Anita, “A Comprehensive Review of AI based Intrusion Detection System,” *Measurement: Sensors*, vol. 28, pp. 1-13, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[22] Salwa Alem et al., “A Novel Bi-Anomaly-based Intrusion Detection System Approach for Industry 4.0,” *Future Generation Computer Systems*, vol. 145, pp. 267-283, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Heng Zeng et al., “Towards a Conceptual Framework for AI-driven Anomaly Detection in Smart City IoT Networks for Enhanced Cybersecurity,” *Journal of Innovation & Knowledge*, vol. 9, no. 4, pp. 1-12, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Matthew Baker et al., “Real-Time AI-based Anomaly Detection and Classification in Power Electronics Dominated Grids,” *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, vol. 4, no. 2, pp. 549-559, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[25] Monika Vishwakarma, and Nishtha Kesswani, “DIDS: A Deep Neural Network based Real-Time Intrusion Detection System for IoT,” *Decision Analytics Journal*, vol. 5, pp. 1-9, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[26] Md. Asaduzzaman, and Md. Mahbubur Rahman, “An Adversarial Approach for Intrusion Detection using Hybrid Deep Learning Model,” *2022 International Conference on Information Technology Research and Innovation (ICITRI)*, Jakarta, Indonesia, pp. 18-23, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Zhibo Zhang et al., “Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research,” *IEEE Access*, vol. 10, pp. 93104-93139, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[28] Ankit Attkan, and Virender Ranga, “Cyber-Physical Security for IoT Networks: A Comprehensive Review on Traditional, Blockchain and Artificial Intelligence,” *Complex & Intelligent Systems*, vol. 8, pp. 3559-3591, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Marcos V.O. de Assis et al., “Near Real-Time Security System Applied to SDN Environments in IoT Networks using Convolutional Neural Network,” *Computers & Electrical Engineering*, vol. 86, pp. 1-39, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[30] Norberto Garcia et al., “Distributed Real-Time SlowDoS Attacks Detection over Encrypted Traffic using Artificial Intelligence,” *Journal of Network and Computer Applications*, vol. 173, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[31] Jalindar Karande, and Sarang Joshi, “Real-Time Detection of Cyber Attacks on the IoT Devices,” *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kharagpur, India, pp. 1-6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[32] Stefanos Tsimenidis, Thomas Lagkas, and Konstantinos Rantos, “Deep Learning in IoT Intrusion Detection,” *Journal of Network and Systems Management*, vol. 30, pp. 1-40, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Minh-Quang Tran et al., “Reliable Deep Learning and IoT-Based Monitoring System for Secure Computer Numerical Control Machines against Cyber-Attacks with Experimental Verification,” *IEEE Access*, vol. 10, pp. 23186-23197, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[34] Mohamed S. Abdalzaher et al., “Toward Secured IoT-Based Smart Systems Using Machine Learning,” *IEEE Access*, vol. 11, pp. 20827-20841, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[35] Martin Manuel Lopez et al., “Machine Learning for Intrusion Detection: Stream Classification Guided by Clustering for Sustainable Security in IoT,” *Proceedings of the Great Lakes Symposium on VLSI 2023*, pp. 691-696, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[36] Vanlalruata Hnamte et al., “A Novel Two-Stage Deep Learning Model for Network Intrusion Detection: LSTM-AE,” *IEEE Access*, vol. 11, pp. 37131-37148, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[37] Jiawei Du et al., “NIDS-CNNLSTM: Network Intrusion Detection Classification Model based on Deep Learning,” *IEEE Access*, vol. 11, pp. 24808-24821, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[38] Tao Yi et al., “Review on the Application of Deep Learning in Network Attack Detection,” *Journal of Network and Computer Applications*, vol. 212, pp. 1-15, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[39] Meenal Jain, and Gagandeep Kaur, “Distributed Anomaly Detection using Concept Drift Detection based Hybrid Ensemble Techniques in Streamed Network Data,” *Cluster Computing*, vol. 24, pp. 2099-2114, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[40] Mahmoud Abbasi et al., “Class Imbalance in Network Traffic Classification: An Adaptive Weight Ensemble-of-Ensemble Learning Method,” *IEEE Access*, vol. 13, pp. 26171-26192, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[41] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, Funchal, Madeira, Portugal, vol. 1, pp. 108-116, 2018. [CrossRef] [Google Scholar] [Publisher Link]

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