

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/378818095>

Drift Adaptive Online DDoS Attack Detection Framework for IoT System

Article in Electronics · March 2024

DOI: 10.3390/electronics13061004

CITATIONS

6

READS

173

3 authors, including:



[Henock Mulugeta](#)



Addis Ababa University

19 PUBLICATIONS 105 CITATIONS

[SEE PROFILE](#)

Article

Drift Adaptive Online DDoS Attack Detection Framework for IoT System

Yonas Kibret Beshah ¹, Surafel Lemma Abebe ²  and Henock Mulugeta Melaku ^{1,*} 

¹ School of Information Technology and Engineering, Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa 1000, Ethiopia; yonkit00@gmail.com

² School of Electrical and Computer Engineering, Addis Ababa Institute of Technology, Addis Ababa University, Addis Ababa 1000, Ethiopia; surafel.lemma@aau.edu.et

* Correspondence: henock.mulugeta@aait.edu.et

Abstract: Internet of Things (IoT) security is becoming important with the growing popularity of IoT devices and their wide applications. Recent network security reports revealed a sharp increase in the type, frequency, sophistication, and impact of distributed denial of service (DDoS) attacks on IoT systems, making DDoS one of the most challenging threats. DDoS is used to commit actual, effective, and profitable cybercrimes. The current machine learning-based IoT DDoS attack detection systems use batch learning techniques, and hence are unable to maintain their performance over time in a dynamic environment. The dynamicity of heterogeneous IoT data causes concept drift issues that result in performance degradation and automation difficulties in detecting DDoS. In this study, we propose an adaptive online DDoS attack detection framework that detects and adapts to concept drifts in streaming data using a number of features often used in DDoS attack detection. This paper also proposes a novel accuracy update weighted probability averaging ensemble (AUWPAE) approach to detect concept drift and optimize zero-day DDoS detection. We evaluated the proposed framework using IoTID20 and CICIoT2023 dataset containing benign and DDoS traffic data. The results show that the proposed adaptive online DDoS attack detection framework is able to detect DDoS attacks with an accuracy of 99.54% and 99.33% for the respective datasets.



Citation: Beshah, Y.K.; Abebe, S.L.; Melaku, H.M. Drift Adaptive Online DDoS Attack Detection Framework for IoT System. *Electronics* **2024**, *13*, 1004. <https://doi.org/10.3390/electronics13061004>

Academic Editors: Mohiuddin Ahmed and Abebe Diro

Received: 9 February 2024

Revised: 26 February 2024

Accepted: 29 February 2024

Published: 7 March 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: IoT; real-time DDoS attack detection; zero-day attack detection; concept drift; concept drift detection and adaptation

1. Introduction

The Internet of Things (IoT) has grown exponentially in the past ten years by changing ordinary objects into smart and intelligent ones that can work together to make decisions. The desire for smart apps and devices that can work autonomously without requiring human involvement has been one of the main drivers in this field. The development of efficient applications, enhanced communication protocols, and breakthroughs in embedded system design along with end user demand have all contributed to the acceleration of IoT growth. It is predicted that the number of connected IoT devices used around the world will increase by 12% on average from 27 billion in 2017 to 125 billion in 2030 [1].

Despite the wide range of applications and technological developments, IoT is susceptible to cyber attacks [2]. The absence of built-in security features has been one of the most obvious shortcomings of the IoT system. This flaw results from the fact that the majority of embedded devices lack the computational power necessary to implement sophisticated security procedures and encryption techniques.

Recent network security reports show that there is a sharp increase in the type, frequency, sophistication, and impact of distributed denial of service (DDoS) attacks on IoT systems, making DDoS one of the most challenging threats [3]. Information and network security measures such as encryption, authentication, and access control techniques are

not sufficient to defend against DDoS attacks on IoT infrastructure [4]. An effective DDoS attack detection solution is required to supplement current security measures. DDoS attack detection monitors any network traffic records that are generated within IoT networks in order to detect DDoS attacks [5]. Such systems can be developed and implemented at IoT network gateways, alert network managers, and prevent DDoS attacks.

In recent years, machine learning has grown in popularity as a technique for DDoS attack detection in IoT networks. Machine learning-based approaches use historical IoT systems normal and DDoS traffic data to train and build a model, and detect DDoS attacks. Such an approach, however, is not effective to detect DDoS attacks in IoT systems with. Its low performance is mainly attributed to IoT's unique properties, such as resource-constrained devices, enormous volumes of data, and real-time requirements. IoT systems have notable security issues because the majority of current industrial security solutions including machine learning approaches require heavy-weight computations and large memory requirements [4]. To address these problems, researchers from academia and industry have conducted several studies on machine learning-based detection techniques that attempt to deliver effective data-driven detection [6,7]. Several DDoS attack detection models are batch learning-based machine learning techniques. Batch learning techniques frequently require access to a complete dataset for model training. Learning massive IoT datasets demands a significant amount of time for retraining, computational resources, and memory due to the real-time nature of the environment. Online learning is more appropriate than are batch learning techniques for IoT DDoS attack detection given the real-time nature of many IoTs [8].

Another challenge for machine learning-based IoT DDoS attack detection approaches is concept drift, which is caused due to the dynamic nature of IoT environments. Concept drift is a situation where the statistical properties of the target variable, i.e., attack, change over time. Concept drift makes already trained models less useful in recognizing zero-day attacks. A good data analytics model must accurately detect and adapt to observed drifts in order to prevent concept drift issues and maintain high prediction accuracy. Concept drift could be categorized into sudden, incremental, gradual, recurring, and noise drifts [5]. Concept drifts are a challenge for IoT DDoS attack detection since the distribution of fault patterns varies over time. It also has data that are imbalanced because data with flaws only make up a small portion of all data.

In this work, a novel Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE) framework is proposed for DDoS attack detection using real-time data streaming. The proposed framework relies on the dynamic nature of incoming streaming data, which leads to rapid changes in data distributions, to build a model that detects concept drifts. The proposed framework reacts to different types of concept drift to perform effectively. The two commonly used drift detection methods, ADWIN [9] and DDM [10], are used in the proposed ensemble framework to detect gradual drift and sudden concept drift, respectively. ARF [11], SRPs [12], and KNN [13] are used as base learners to construct the ensemble framework. The proposed framework is evaluated on two benchmark IoT datasets: IoTID20 and CICIOT2023 datasets. The results show that the proposed adaptive online DDoS attack detection framework is able to detect DDoS attacks on IoT systems with an accuracy of 99.32% and 99.25% for the two datasets, respectively.

The contributions of this paper are as follows:

1. We proposed a novel Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE) framework for online DDoS detection that addresses gradual drift and sudden concept drift issues in a dynamic environment.
2. We evaluated the proposed framework using two well-known, publicly available IoT security datasets as a case study. We investigated and compared the proposed framework with various state-of-the-art online ensemble learning techniques.

The rest of the paper is organized as follows. Section 2 presents related works on DDoS attack detection, and concept drift detection and adaptation. Section 3 describes the proposed framework for adaptive online DDoS attack detection. Section 4 presents

the experimental environment, performance metrics, and dataset used in the experiment. Section 5 discusses the experiment results, while Section 6 presents the practical scenario in which the proposed solution could be deployed. Finally, Section 7 presents the conclusion of the paper.

2. Related Work

2.1. DDoS Attack Detection

The development of efficient and effective DDoS attack detection techniques in IoT systems has received research attention in the past decade [8]. Special focus has been given to implementing DDoS detection based on network traffic analysis. Machine learning-based DDoS detection techniques, in particular, have been hailed as promising for making inferences about DDoS attacks. Chen et al. [7] proposed a machine learning-based multi-layer IoT DDoS attack detection framework that includes IoT devices, IoT gateways, software-defined network (SDN) switches, and cloud servers. The author constructed eight smart poles equipped with different sensors on campus networks and collected sensor data as datasets over wired and wireless networks. The experimental findings demonstrated that the multi-layer DDoS detection systems have accuracy above 97% for different datasets. Additionally, the SDN controller can efficiently block malicious devices based on blacklists generated by the proposed DDoS attack detection framework.

Attota et al. [14] proposed an ensemble multi-view federated method for IoT intrusion detection. The authors addressed the limitations of centralized deployment by using edge computing paradigms for maintaining data privacy. Three artificial neural network (ANN) models were trained using the bidirectional traffic flow, unidirectional traffic flow, and packet of network traffic features. To select the optimum set of network traffic features for ANN model training, the grey wolf optimization (GWO) technique was used. This reduced the amount of memory space needed to store the training data by reducing the dimensionality of the network traffic features. ANN models' outputs are fed into a random forest (RF) model to predict attacks. The hyper parameters of the models, however, are not disclosed. An evaluation of the proposed approach was conducted using the MQTT dataset, which lacked samples of IoT DDoS attacks.

Nguyen et al. [15] demonstrated the vulnerability of federated learning-based intrusion detection systems to backdoor attack. Backdoor attack is a type of poisoning attack in which the attacker corrupts specific input to a model in order to make incorrect predictions. The anomaly detection system used the gated recurrent unit (GRU), which is a type of recurrent neural network (RNN) for detecting the anomaly behavior of IoT devices. The experiment results showed the effectiveness of backdoor attacks in circumventing state-of-the-art defenses against federated learning poisoning. The attacker performed white box poisoning on the training data by stealthily injecting malicious traffic into the benign training dataset. As a result, the model incorrectly classified malicious traffic as benign.

Ullah et al. [6] developed anomaly-based intrusion detection models for IoT networks using convolutional neural network models. The proposed convolution neural network (CNN) model was implemented using 1D, 2D, and 3D network architectures. CNN and transfer learning were employed as deep learning models for multi-class and binary classification on BoT-IoT, IoT network intrusion, MQTT-IoT-IDS2020, and IoT-23 intrusion detection datasets.

Chenget al. [16] proposed the federated transfer learning approach for intrusion detection in mobile edge computing. The federated learning approach uses transfer learning to speed up model training, lower computational costs, boost communication effectiveness, and enhance classification performance. CNN model architecture was utilized for binary classification that included three convolutional layers, two max-pooling layers, a batch normalization layer, a dropout layer, and two dense layers. The NSL-KDD dataset was used to train the model in the source domain, and the UNSW-NB15 dataset was utilized to finish the training in the target domain in order to evaluate the performance of the method.

Zainudin et al. [17] provide the CNN and LSTM hybrid mode for DDoS attack classification. The author utilized the extreme gradient boosting (XGBoost) feature selection technique in order to determine the top 10 relevant features. The proposed model was evaluated using CIC-DDoS2019 dataset, which includes DDoS attack and benign data. The experiment results show an accuracy of 99.5%.

Kumar et al. [18] developed an LSTM-based DDoS attack detection model using the CICDDoS2019 dataset. The proposed model was performed using binary classification to distinguish between DDoS attacks and benign traffic. The experimental results show that the proposed model achieved an accuracy of 98.6%.

The recent research papers on machine learning-based DDoS attack detection for IoT devices are summarized in Table 1. From the above, we can see that none of the above works considered the dynamicity of the IoT environment, which could result in concept drift. The learning methods proposed in the above works are not online attack detection methods. In this research, we proposed an Adaptive Online DDoS Attack Detection framework for IoT System.

Table 1. Summary of related works. ✕ indicates absence of drift detection while ✓ indicates presence of drift detection.

Reference	Year	Model	Drift Detection	Learning Method	Dataset
Zainudin et al. [7]	2020	FL	✕	Batch learning	ToN_IoT
Attota et al. [14]	2021	ANN and RF	✕	Batch learning	MQTT dataset
Nguyen et al. [15]	2020	GRU	✕	Batch learning	IoTDs
Ullah et al. [6]	2021	CNN	✕	Batch learning	BoT-IoT, IoT MQTT-IoT-IDS2020, and IoT-23
Cheng et al. [16]	2022	CNN	✕	Batch learning	NSL-KDD and UNSW-NB15
Zainudin et al. [17]	2022	CNN and LSTM	✕	Batch learning	CIC-DDoS2019
Kumar et al. [18]	2023	LSTM	✕	Batch learning	CIC-DDoS2019
Proposed Solution		Ensemble	✓	Online learning	IOTID20 and CICIoT2023

2.2. Concept Drift

The majority of IoT devices used in IoT systems have limited computing power and storage [4]. This limited memory capacity hinders their ability to handle and retain massive amounts of data and complex learning models. Therefore, it is crucial to develop analytics models with low computing complexity. Online learning methods that enable real-time analytics are able to fulfill IoT system time and memory requirements. Online learning based approaches can continuously update the learning model with new data samples as they arrive in a short execution time while batch learning based approaches need to frequently re-train the learning model on the entire training dataset which takes relatively longer execution time. IoT data samples are usually produced dynamically in constantly changing IoT environments. The dynamic nature of streaming traffic also makes already trained models less useful in recognizing zero-day attack.

Data analytics frequently experiences concept drift issues in real-world applications due to the change in IoT data distribution over time. Concept drift is caused by three main data distribution changes: recurring, gradual, and sudden [19]. Sudden drift happens when rapid irreversible changes occur in a short period of time. Gradual drift happens when the data distribution gradually replaces the old one over time. Recurring drift happens when the previous data distribution happens again over time.

Concept drift is formally defined as a set of samples, indicated as $S_{0,t} = \{d_0, \dots, d_t\}$ for a time interval of $[0, t]$, where $d_i = (x_i, y_i)$ represents a single observation, x_i is the feature vector, y_i is the label, and $S_{0,t}$ adheres to a specific distribution $f_{0,t}(x, y)$. If $f_{0,t}(x, y) \neq f_{t+1,\infty}(x, y)$, denoted as $\exists_t : p_t(x, y) \neq p_{t+1}(x, y)$, then concept drift happens at timestamp t_{t+1} . Concept drift at time t can also be defined as the change in the joint probability

of x and y at time t , expressed as $p_t(x, y) = p_t(x) \times p_t(y/x)$. Concept drift will happen under the following three conditions.

- $p_t(x) = p_{t+1}(x)$ while $p_t(y/x) = p_{t+1}(y/x)p_t(x)$. This type of drift is known as virtual drift because $p_t(x)$ does not affect the decision boundary.
- $p_t(y/x) \neq p_{t+1}(y/x)$ while $p_t(x) = p_{t+1}(x)$ while $p_t(x)$ remains unchanged. This is considered actual drift because it affects the decision boundary and also leads to a decline in learning accuracy.
- The combination of the first two, $p_t(x)p_{t+1}(x)$ and $p_t(y/x)p_{t+1}(y/x)$.

Concept drift poses significant issues when building machine learning models because it can cause changes in data distribution that result in machine learning model performance gradually declining over time [20]. Hence, advanced online machine learning models need to be developed in order to detect and adapt the concept drift that occurs in IoT data streams.

2.3. Drift Detection

Drift detection is a crucial element for adaptive machine learning models that can solve concept drift issues. The two primary categories of drift detection methods are performance-based and distribution-based [21].

2.3.1. Performance-Based Methods

Performance-based concept drift detection methods are based on changes in the metrics used to evaluate model performance. Common examples of indicators of concept drift are accuracy decline and increase in error rate. If the error rate of a learner gradually decreases or remains constant as more samples are learned, it often indicates constant data distribution without drift. On the other hand, if a learner's error rate drastically increases as more data is processed; it often indicates the presence of concept drift. The two well-known performance-based drift detection methods are the drift detection method (DDM) and early drift detection method (EDDM), which are able to detect concept drift by keeping track of degradations in performance. The DDM is a popular performance-based drift detection method that measures model error rate and standard deviation changes using two predefined thresholds: the warning threshold and the drift threshold [19]. DDM often performs well on data streams with sudden drift, but its reaction time is often too slow for detecting gradual drift. The early drift detection method (EDDM) is an enhanced version of the DDM that detects concept drift using the same concept drift detection mechanism [10]. The EDDM often performs well on data streams with gradual drift. Even though the EDDM frequently performs better than does the DDM, it still falls short for sudden drift. Furthermore, due to its sensitivity to noise, it may mistake noise for drift, resulting in false alarms.

2.3.2. Distribution-Based Methods

Distribution-based concept drift detection is based on changes in data distributions. Data distribution changes can be measured using statistical variables such as mean, variance, and information entropy. Adaptive windowing (ADWIN) is one such widely used method that uses adaptive sliding windows to detect concept drift based on the statistical difference between two adjacent sub-windows. The adaptive windowing method uses characteristic values such as the mean and variance, as well as variable-size sliding windows to detect concept drift. The window size is dynamically increased if there is no concept drift and reduced when concept drift is detected [9]. IoT systems with less memory often adopt distribution-based methods since they only need to retrain on the most recent samples. Windowing methods are often quick and simple to use. However, they could miss certain important historical data.

2.4. Drift Adaptation

Concept drift must be handled after it is detected by updating the current models using the appropriate drift adaptation methods. Drift adaptation is a procedure used to update a model automatically when a concept drift occurs to enhance performance of model and detect zero-day attacks. The procedures usually fully retrain or alter the learning model using new dataset. The three categories of drift adaptation methods are model retraining, incremental learning and ensemble learning.

2.4.1. Model Retraining

One of the more simple and straightforward methods to handle concept drift is model retraining. Offline models are often unable to accurately predict unseen incoming streaming data. This problem can be addressed by retraining the model using the most recent data streams. However, employing this technique can cause unnecessary model retraining or drift adaptation delays. Therefore, it is crucial to use learning models together with an appropriate drift detection method to determine when to retrain the learning model for timely and necessary updates.

There are two types of model retraining methods: full retraining and partial retraining. Retraining the learning model using the entire dataset and all available samples is known as full retraining, while partial retraining trains the model on selected parts of the dataset. The window-based method is used to partially retrain a model using the most recent data. This reduces training times but may result in the loss of historical patterns. Hence, selecting the right window size is crucial. ADWIN is a drift detection method that uses model retraining. ADWIN performs better as it uses a dynamic window to fit new data [9]. The learning model is partially retrained on only the new concept samples in order to save training time.

2.4.2. Incremental Learning Methods

Incremental learning has become commonly used in data stream analytics research. Incremental learning involves updating the learning model when each instance is processed. Incremental learning methods often partially update the learning model to fit a new data sample [22]. They do not require a sufficient amount of data prior to the training process due to the incremental learning approaches' capacity to support progressive learning. However, only a few machine learning algorithms such as MLP and multinomial NB support partial updates.

2.4.3. Ensemble Learning Methods

Ensemble learning approaches have been developed to provide powerful learners for data stream analytics to enable stronger concept drift adaptation. Ensemble learning combines multiple base learners to tackle the same problem [23]. Ensemble learning base learners can be constructed using different algorithms, and different configurations of hyper parameters configuration. Ensemble learning models are often more generalizable than single models because they combine the outputs of multiple base learners. Reusing existing models in an ensemble is significantly more effective for concept drift adaptation than training new models on data streams with recurrent concept drift [24].

Block-based ensembles and online ensembles are the two main categories of ensemble techniques used in data stream analytics [25]. Data streams are divided into fixed-size blocks by block-based ensembles, which then train a base learner on each block. The base learners will be evaluated and updated each time a new block arrives. Block-based ensembles react to gradual drifts accurately, but they frequently take longer to respond to sudden drifts. Three common block-based ensembles are accuracy-weighted ensemble (AWE) [26], accuracy-updated ensemble (AUE) [27], and the streaming ensemble algorithm (SEA) [25]. The AUE often performs better among the block-based ensembles [26]. AUE uses non-linear error functions to apply weights to base learners in order to improve performance.

To enhance the learning performance of online ensembles, different incremental learning models such as Hoeffding trees (HTs) are included. Gomes et al. [11] proposed the

adaptive random forest (ARF) approach, which makes use of HTs as base learners and ADWIN as a drift detector. The drift detection mechanism replaces underperforming base trees with new trees that better fit. Since random forest is a well-known, effective machine learning algorithm, ARF frequently outperforms many other methods. ARF also includes a powerful re-sampling method and the flexibility to accommodate various drifts types.

Gomes et al. [12] also proposed streaming random patches (SRPs) for the adaptive ensemble approach. Although its execution time is usually longer, SRPs often exhibit slightly better prediction accuracy than ARF. Leverage bagging (LB) [28] is another online ensemble that constructs base learners using bootstrap samples. Although LB is simple to construct, it usually performs worse than SRPs and ARF. Despite the fact that there are several concept drift adaptation strategies in use today, their efficacy is constrained by slow drift response and poor prediction accuracy.

Incremental learning methods often perform poorly due to their low model complexity and limited drift adaptability [29], whereas block-based ensembles face significant difficulties in determining block size and responding quickly to drift. Online ensembles, like ARF and SRPs, often outperform incremental learning and block-based ensemble methods. However, because of their randomization strategies, they give unstable learning models, adding more unpredictability to DDoS attack detection.

In line with the previous research, this research aims to detect DDoS attacks in IoT systems while addressing the concept drift issue. In particular, this paper proposes adaptive online DDoS attack detection using the ensemble learning method to address the issue of concept drift in IoT system DDoS attack detection.

3. Proposed Framework

3.1. Overview of Proposed Framework

The proposed online DDoS attack detection framework using online data stream analytics in IoT systems has three main steps: preprocessing IoT data, DDoS detection using a base model, and DDoS detection using online ensemble methods. The K-means clustering approach is used to acquire a more representative subset of the incoming IoT data streams. In order to standardize sample data distribution and scale every feature, a Z-score and min-max normalization are used. This step is discussed in Section 3.2. To handle concept drift adaptation and perform DDoS attack detection, four base learners, ARF-ADWIN, ARF-DDM, SRPs-DDM, and KNN-ADWIN, are developed. The development of the base learners is presented in Section 3.3. These base learners are made to adapt to the changing data distribution and detect DDoS attack in the data stream. Finally, the proposed AUWPAE approach, which is based on the ensemble model, is discussed in Section 3.4. AUWPAE combines the prediction of base models using their real-time mean square error rates. Figure 1 shows the proposed drift adaptive online DDoS attack detection framework for IoT systems.

3.2. Data Preprocessing

Data preprocessing is a basic step in all machine learning applications including DDoS attack detection. The performance and accuracy of the detection method can be significantly impacted by the representation, size, and quality of the incoming data. Particularly, selecting a dataset with high dimensionality and a large number of duplicate and irrelevant features will affect training. To address these issues, in the data preprocessing phase, we used data cleaning, data encoding, data normalization, and feature selection techniques.

3.2.1. Data Cleaning

The selected dataset network contains a number of different distinct dataset files that are merged into a single file. The merged single file is referred to as a combined dataset. IoT data values are generated in real-world applications as words or strings. Real-world datasets frequently have missing values due to data accessibility issues or challenges in gathering the required data. Missing values such as blank spaces, NaNs, and erroneous

data types could be represented using null values. Most machine learning models, however, cannot deal with missing values directly or are negatively impacted during learning. The goal of data cleaning is to fill the gaps left by missing data with acceptable values [28]. A number of basic cleaning techniques are used to fill missing variables. In this research, we replaced the missing values with the means of each column.

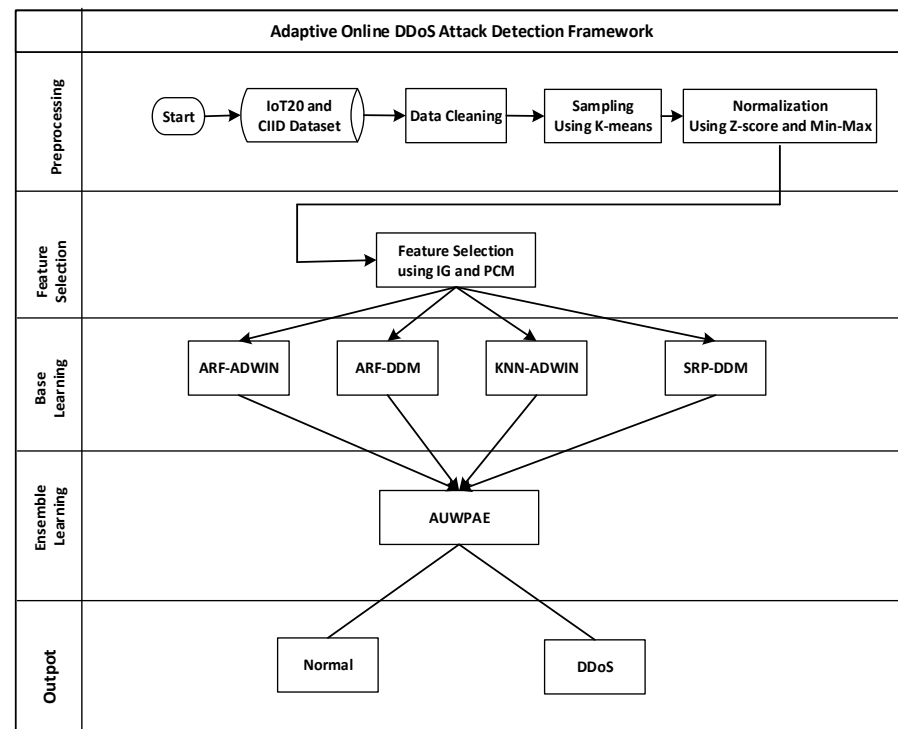


Figure 1. Drift adaptive online DDoS detection framework.

3.2.2. Data Sampling

The selection of highly representative data samples requires the use of an efficient data sampling method. The proposed framework uses the K-means based cluster sampling method to obtain a representative subset. Clustering algorithms have important roles in unsupervised models in grouping data samples based on their similarities. One of the common clustering methods is K-means, which divides an unlabeled dataset into K clusters based on the degree of similarity between data points.

3.2.3. Data Encoding

In real world IoT data, values are generated as words or strings to make them human-readable. Data encoding involves the conversion of string information into numerical features that machine learning models are able to understand and process. Label, one-hot, and target encoding are examples of frequently used encoding methods. In this research, we used target encoding to replace categorical values with the means of the target variables.

3.2.4. Data Normalization

There are different attribute values in the combined dataset, and some features have values that are spread out over a large range while other values are spread out over a small range. This variation could affect the performance of the models. To solve this issue, we used normalization techniques to scale the feature values in the same range. Two common normalization methods for data analytics are Z-score and min-max normalization. The min-max normalization approach transforms the original data linearly to achieve a balance of comparative values between the data before and after processing. The min-max

normalization method scales a feature's values to a range between 0 and 1. The min–max scaling approach is shown in Equation (1) [30]:

$$X_{new} = \frac{X - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \quad (1)$$

where X_{new} is the result of normalization and X is the value to be normalized; Min and Max are the minimum and maximum values in each feature.

The Z-score normalization method is based on the mean and standard deviation of the data. This method is helpful if the minimum and maximum values of the data are unknown. Equation (2) shows the formula for Z-score normalization [31]:

$$X_{new} = \frac{X - \mu}{\delta} \quad (2)$$

where X_{new} is the result of normalization, X is the value to be normalized, μ = population mean, and δ = standard deviation. Min–max normalization can keep outliers in datasets; hence, it is more appropriate for DDoS attack detection models. On the other hand, Z-score normalization is robust against outliers; hence, it usually works well for data analytics problems related to non-outliers. As a result, the proposed framework chooses both the Z-score and min–max normalization to handle this issue.

3.2.5. Feature Selection

Feature selection is used to choose a subset of the original feature set in order to increase the performance and speed of machine learning models. The best-performing minimal feature set can be identified by evaluating all feature combinations. However, with a high number of features, optimization approaches can be used to examine the feature search space and determine the optimal feature set. Information gain (IG) and Pearson correlation methods are used to remove irrelevant and duplicate features, respectively.

3.3. Drift Adaptation Base Model Selection

The proposed framework consists of four base models that are used to construct the online ensemble model (see Figure 1). The framework aims to balance learning performance and efficiency since it is designed for online IoT systems. Most of the existing data stream classification algorithms specialize in only one type of drift. Some classifiers are best suited for gradual drift while others are suited for sudden drifts. Our research aims to develop a data stream classifier that reacts to both types of concept drift. In this research, the two commonly used drift detection methods, ADWIN and DDM, are used together to detect concept drift. ADWIN's sliding window can be expanded to a large-size window to detect long-term changes. This makes ADWIN an ideal match for data streams with gradual drift. DDM often works well on data streams with sudden drift, but its reaction time is often too slow for detecting gradual drift.

Online ensembles can improve learning performance using various incremental learning models like Hoeffding trees (HTs). The adaptive random forest (ARF) method uses HTs as base learners and ADWIN as a default drift detector [11]. The drift detection mechanism replaces underperforming base trees with new trees that fit the new concept. Since the random forest method is an effective technique, ARF often outperforms a variety of other methods. ARF makes the best use of re-sampling, and it is adjusted to a variety of drift types.

Streaming random patches (SRPs) is an alternative adaptive ensemble method for streaming data analytics [12]. SRPs use a combination of online bagging and random subspace algorithms for predictions. It uses a similar approach to the one above, except SRP employs a global subspace randomization mechanism and ARF's local subspace randomization. A flexible technique for increasing the diversity of base learners is global subspace randomization. Although SRPs' execution time is longer than ARF's, its prediction accuracy is frequently slightly higher.

ARF and SRPs are advanced drift adaptation approaches that have shown better performance through experimental research than other drift adaptation approaches have [11,12]. ARF is an advanced ensemble model that builds HTs for drift adaptation using local subspace randomization and that uses a drift detector for drift detection. ARF has demonstrated good performance and excellent execution time compared to those of others in solving data stream analytics problems [11]. As a result, ARF-ADWIN and ARF-DDM drift detectors are selected as base models for the proposed ensemble framework. The ensemble uses two ARF models with different drift detectors to preserve high accuracy.

The other two models are then selected among seven online learning methods (LB, SRPs, OPA, PWPAE, SRPs-ADWIN, SRPs-DDM, and KNN-ADWIN) by considering execution time and performance. LB [32], SRPs, and SRPs-ADWIN can effectively solve concept drift but they are not as effective as ARF. Although HTs' computational costs are low, this method is not selected due to performance limitations. The other two models, OPA and PWPAE [29], are not selected because of their high computational complexity. Hence, ARF-ADWIN, ARF-DDM, SRPs-DDM, and KNN-ADWIN are selected for our proposed online learning framework. Below is a summary of the justifications for selecting these models.

1. ADWIN performs better for detecting gradual drift whereas DDM performs effectively for detecting sudden drift. Combining these two makes it possible to detect sudden and gradual concept drifts.
2. Experimental results demonstrate that ARF-ADWIN and ARF-DDM perform better at handling concept drift than do other drift adaptation approaches. Since ARF-ADWIN and ARF-DDM are extensions of the adaptive random forest (ARF) method, the proposed framework inherits the ability to adapt to concept drift and improve overall system performance.
3. The selected models, ARF-ADWIN, ARF-DDM, SRPs-DDM, and KNN-ADWIN, result in better performance while maintaining a shorter execution time. Hence, they are well suited for online DDoS attack detection and concept drift adaptation.

3.4. Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE)

In this research, a novel Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE) is proposed for combining the base learners for IoT data stream analytics. AUWPAE makes use of the advantages of the previous approaches while mitigating their limitations. AUWPAE provides dynamic weights to base learners in accordance with their real-time performance. The AUWPAE approach utilizes the weighted average approach. The prediction probability of each base learner is assigned a weight and multiplied by the prediction probability of that base learner. The multiplication results are summed to determine the final prediction probability. The prediction class of the proposed model will be the final one with the highest average probability [22]. Given a data stream, $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, consisting of c different target classes, $y \in 1, \dots, c$. The predicted target category, \hat{y} , for each data input, x , can be expressed mathematically using the following Equation (3) [20]:

$$\hat{y} = \left(\arg \max_{i \in (1, \dots, c)} \right) \frac{\sum_{j=1}^k w_j p_j(y = i | L_j, x)}{b} \quad (3)$$

where L_j stands for the j th base learner model, $p_j(y = i | L_j, x)$ defines the probability of predicting a class value, i , on a data sample, x , using the j th base learner, L_j ; b is the number of base learner models, which in our case is $b = 4$; and w_j denotes the weight of each base learner model, L_j . The weight of base learner can be quantitatively assessed using the accuracy-updated ensemble method [26]. For every data stream, the weight of the base

learner model, w_j , is calculated by estimating the mean square error rate on data stream $D = (x_1, y_1), (x_2, y_2), \dots (x_n, y_n)$, as shown in Equations (4)–(6):

$$MSE_{ij} = \frac{1}{|x_i|} \sum_{\{x,y\} \in x_i} (1 - f_y^j(x))^2 \quad (4)$$

$$MSE_r = \sum_y p(y) (1 - p(y))^2 \quad (5)$$

$$w_{ij} = \frac{1}{MSE_r + MSE_{ij} + \epsilon} \quad (6)$$

where the function $f_y^j(x)$ represents the probability given by the base learner model, L_j , that x is an instance of class y . The accuracy-updated ensemble algorithm considers probabilities for all classes instead of making predictions for a single class. MSE_{ij} evaluates the prediction error of the base learner model, L_j , on datum x_i . MSE_r is the mean square error of a randomly predicted base learner model and is used as a reference point for the current class distribution. To avoid issues with division by zero, a very little positive value, ϵ , is also added to the equation. Equation (6) is used as weighting formula to combine the accuracy of the base learner model with the current class distribution. In addition, assigning ensemble members new weights for each datum, x_i , a candidate base learner model, L_j , is built from instances in the most recent data. L_j is a “perfect” base learner model, since it is trained on the most recent data. Equation (7) is used to determine its weight.

$$w_{ij} = \frac{1}{MSE_r + \epsilon} \quad (7)$$

The weight of the candidate base learner model, L_j , does not take into consideration the prediction error for base learner L_j in comparison with the function used to weight the members of the current ensemble. This approach is based on the assumption that the most recent data reflect the distribution of the present and the near-future data. L_j is considered to be the best base learner model available, because it is trained using the most recent data. The proposed Accuracy Update Weighted Probability Average Ensemble Algorithm (Algorithm 1) is shown below.

The computational complexity of the AUWPAE model is primarily determined by the complexity of the selected based models; the AUWPAE method itself has a linear computational complexity of $O(pwb)$, where p is probability of predicting the base learner, b is the number of base learner models, which in our case is $b = 4$, and, w denotes the weight of each base learner model. Most of the performance comparisons are conducted with the base learners used in the proposed framework. AUWPAE adds a weighting algorithm that is linear in complexity on top of the base learners. Hence, the proposed framework should have nearly the same complexity as the base learner, which can be further improved by replacing lower complexity base learners.

The proposed Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE) makes use the advantages of the previous approaches while mitigating their limitations. We developed AUWPAE to address both sudden and gradual drifts, and enhance the detection and adaptability capabilities of current methods. Further, this study aims to propose a novel ensemble framework that successfully achieves a trade-off between execution time and predictive accuracy.

Algorithm 1: Accuracy Update Weighted Probability Averaging Ensemble (AUWPAAE).

```

Input: X_train, y_train, X_test, y_test
Output: t, m (time steps and accuracy metrics)
1  function AUWPAAE (X_train, y_train, X_test, y_test)
2      Initialize a, t, e, eps=0.0001
3      Initialize base learners j = 4
4      for each instance (xi, yi) in X_train, y_train do. // learn base learners using training set
5          Lj ← (xi, yi)
6      end for
7      for each instance (xi, yi) in X_test, in y_test do. // predict using test set
8          ypredj ← bl(xi) //predict target
9          Yprobj ← blProbPred(xi) //target prediction probability
10         bl.update(xi, yi) //update base learner.
11     end for
12     for j = 1 to 4 do
13         ei ← MSEi(yi, ypredj) //calculate real-time MSE for each base learner
14         W ← 1/(ej + eps) //calculate weight
15         yprobj0 ← yprobj //probability for class 0
16         yprobj1 ← yprobj //probability for class 1
17         yprob0 ← yprobj ∑j=14 Wj*yprobj0 //average weighted probability for class 0
18         yprob1 ← yprobj ∑j=14 Wj*yprobj1 //average weighted probability for class 0
19     end for
20     If yprob0 > yprob1
21         ypred ← 0
22     else yprob0 < yprob1
23         ypred ← 1
23     calculate accuracy, precision, recall, and F1-score;
25     return t, a
26 end function

```

4. Experiment

4.1. Dataset

To evaluate the proposed online DDoS detection framework, we used two security datasets: CICIoT2023 and IoTID20. In 2023, the Canadian Institute for Cyber security developed CICIoT2023 dataset [33]. To generate the attack traffic, 33 attacks are executed on 105 IoT devices as targets. The traffic includes normal and attack traffic from DDoS, DoS, Recon, and Web-based DDoS attack, brute force, spoofing, and Mirai. The dataset contains the most common and current attacks as of this time. However, for this experiment, we use DDoS and normal traffic. CICIoT2023 is a new realistic IoT dataset that was created by generating real IoT device traffic data from both legitimate and malicious IoT devices that include different DDoS attacks.

The IoTID20 dataset has been used for developing DDoS attack detection in several studies [34,35]. The authors of the IoTID20 dataset [36] performed binary and multiclass classification, and reported the accuracy scored for different classifier methods. DDoS attack types included in the IoTID20 dataset are: Mirai-ACK flooding, Mirai brute force, Mirai-HTTP flooding, and Mirai-UDP flooding. The IoTID20 dataset is a relatively new dataset that considers IoT devices while containing DDoS attacks, and several recent works have used it to develop IDS [37]. IoTID20 focuses on IoT security and provides a wide range of attack and normal samples from various IoT devices.

4.2. Performance Metrics

The proposed framework is analyzed from a variety of perspectives to provide a comprehensive view of the experiment. The four-performance metrics, accuracy, precision, recall, and F1-score, are the primary metrics we used to evaluate the performance of the

proposed framework. Latency and throughput are also measured to evaluate the learning and detection speed.

In this case, latency refers to the delay in the response time to a specific input. It represents the typical processing time needed by our model to analyze and classify each sample. On the contrary, throughput is a measure of how many data the system can process in a given amount of time. In our context, it corresponds to the number of samples our model can analyze and classify in a given unit of time. Low latency and high throughput are the two essential performance criteria for machine learning and data analytics models. A balance between prediction accuracy and latency should be maintained by efficient learning models in order to achieve real-time analytics.

4.3. Experiment Environment

The proposed online DDoS detection framework aims to detect DDoS attacks on IoT systems. To observe the performance of the proposed approach, we developed a prototype using Python 3.10 programming language in the Jupyter Notebook environment. The River library [35] was used for data stream analytics and addressing concept drift through machine learning. The experiment was conducted on a machine running Intel(R) Xeon(R) CPU @2.20 GHz and with 16 GB of RAM.

5. Experimental Results and Discussion

The datasets used during the experiments are IOTID20 and CICIOT2023. The attack patterns on both datasets have changed overtime, resulting in three and seven concept drifts on the IOTID20 and CICIOT2023 datasets, respectively. The changes and in particular the occurrence of the concept drifts in the datasets show the dynamic nature of IoT streaming traffic. The concept drifts are shown in Figures 2 and 3 using black arrows. Except for the third and the last drifts shown for the CICIOT2023 dataset in Figure 3, all other concept drifts shown for both the IOTID20 dataset in Figure 2 and CICIOT2023 dataset in Figure 3 are sudden drifts. The third and the last drifts shown for the CICIOT2023 dataset in Figure 3 are gradual drifts. The performance of the proposed model, which is AUWPAE, is compared with that of other state-of-the-art online adaptive learning methods, such as ARF-ADWIN, ARF-DDM, SRPs-ADWIN, SRPs-DDM, KNN-ADWIN, HTs, LB, and PWPAE, using two datasets: IOTID20 and CICIOT2023. The experimental results are presented in Figures 2 and 3, and Tables 2 and 3.

Figure 2 and Table 2 show performance comparisons of the proposed model, which is AUWPAE, with other previously proposed models, i.e., ARF-ADWIN, ARF-DDM, SRPs-ADWIN, SRPs-DDM, HTs, LB, KNN-ADWIN, and PWPAE, using the IOTID20 dataset. The performance metrics used for comparison were accuracy, precision, recall, F1-score, latency, and throughput. From the experimental results, in terms of accuracy, the proposed model achieved an average accuracy of 99.54%. This result shows that AUWPAE outperforms the other online adaptive learning methods. The reason why the proposed model performed better than the others is mainly attributed to the weighting algorithm used for ensemble. The two selected base models, SRPs-DDM and KNN-ADWIN, achieved better accuracy. The accuracies of SRPs-DDM and KNN-ADWIN are 98.84% and 99.13%, respectively, while using the IOTID20 dataset. This is because the proposed model is an extension of the AUE, and the weighting of classifiers is performed using non-linear error function, which contributes to performance enhancement. In terms of precision, recall, and the F1-score, the experimental results show that the proposed model achieved 99.51%, 99.99%, and 99.76%, respectively. These results show that the proposed approach has relatively accurate and precise DDoS attack detection capability than the other solutions and is robust to concept drifts.

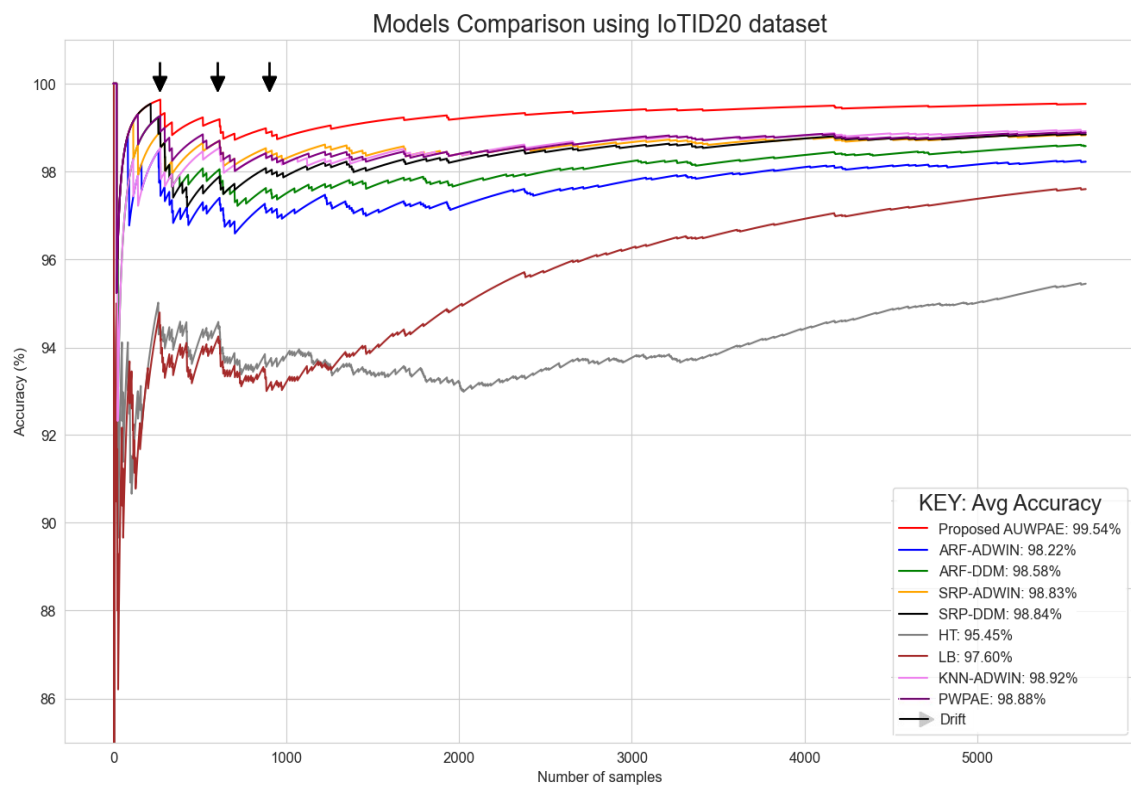


Figure 2. Model comparison using IoTID20 dataset. Arrows indicate concept drifts.

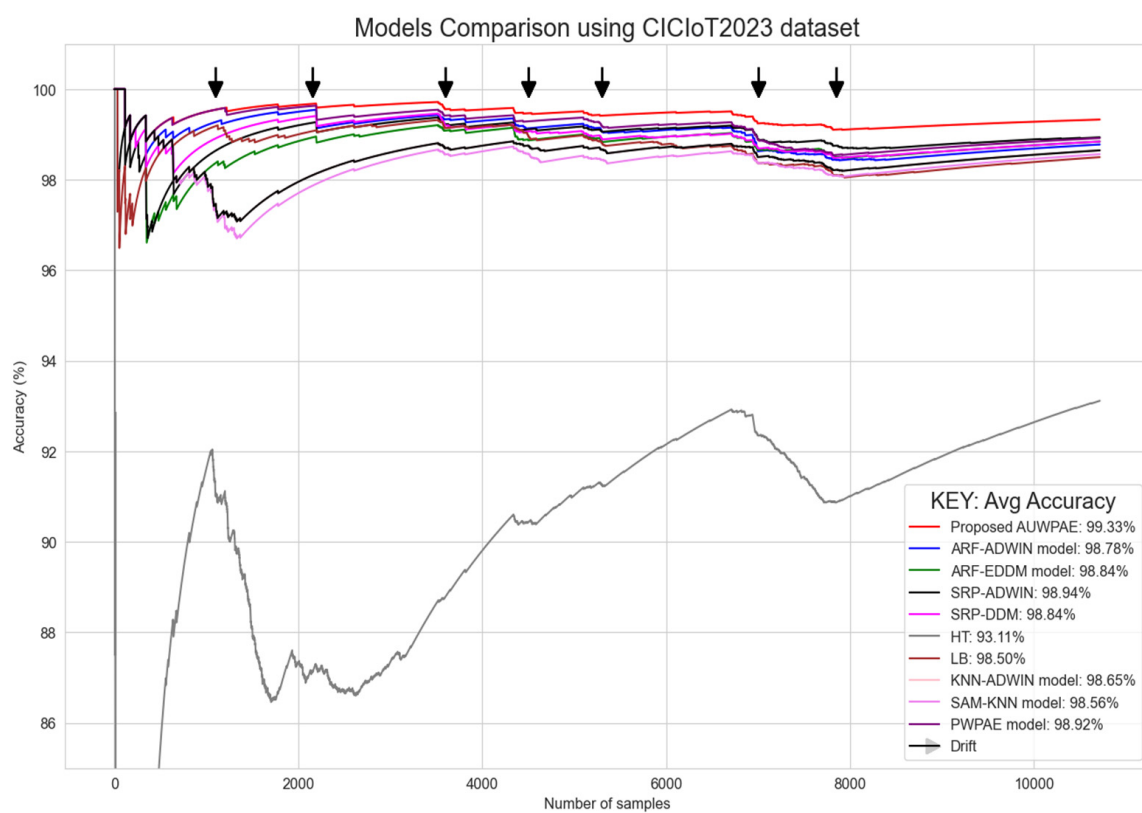


Figure 3. Models comparison using CICIDS2017 dataset. Arrows indicate concept drifts.

Table 2. Model comparison using IoTID20 dataset.

Models Comparison Using IoTID20 Dataset						
	Accuracy	Precision	Recall	F1-Score	Latency (ms)	Throughput (Sample per Second)
ARF-ADWIN	98.22	98.60	99.53	99.06	0.24	4166
ARF-DDM	98.58	98.80	99.7	99.25	0.26	3846
SRPs-ADWIN	98.83	98.94	99.83	99.38	2.79	358
SRPs-DDM	98.84	98.94	99.85	99.39	2.90	345
HTs	95.45	95.91	99.42	97.63	0.11	9090
LB	97.60	98.22	99.25	98.72	1.56	641
KNN-ADWIN	99.13	99.10	99.75	99.42	1.15	869
PWP AE	98.88	98.86	99.96	99.41	2.79	358
AUWP AE	99.54	99.51	99.99	99.76	2.73	365

Table 3. Model comparison using CICIOT2023 dataset.

Models Comparison Using CICIOT2023 Dataset						
	Accuracy	Precision	Recall	F1-Score	Latency (ms)	Throughput (Sample per Second)
ARF-ADWIN	98.78	97.37	94.58	95.95	0.26	3723
ARF-EDDM	98.83	98.16	94.21	96.15	1.29	773
SRPs-ADWIN	98.94	97.75	95.25	96.48	2.32	430
SRPs-DDM	98.83	97.03	95.37	96.19	2.41	413
HTs	93.11	75.89	80.69	78.22	0.25	3997
LB	98.50	95.44	95.81	95.62	0.12	7692
KNN-ADWIN	98.65	94.53	96.77	95.64	0.13	7182
PWP AE	98.92	97.75	95.13	96.14	1.38	721
AUWP AE	99.33	99.88	96.53	97.78	1.29	773

In terms of latency, the proposed model uses an average of 2.73 ms to classify a given sample as attack and normal while using IoTID20 dataset (see Figure 2 and Table 2). The other online adaptive models HTs, ARF-ADWIN, ARF-DDM, KNN-ADWIN, and LB performed better than the proposed approach, i.e., AUWP AE, in terms of processing time. AUWP AE is better than only PWP AE, SRPs-ADWIN and SRPs-DDM, respectively. The reason why the proposed model took more time to process and give results is mainly attributed to the ensemble algorithm. However, on the proposed ensemble model, ARF-ADWIN and ARP-ADWIN have positive contribution on latency. Since, the two models ARF-ADWIN and ARF-DDM achieved excellent latency compared to the other models which are 0.24 ms and 0.26 ms using the IoTID20 dataset, respectively. ARF-ADWIN is the second fastest approaches in the experiment while KNN-ADWIN is the third.

For this experimental setup, the last performance metrics used were throughput. The experimental result shows that the average throughput achieved by the proposed model (AUWP AE) is 365 samples/second. This result would mean that AUWP AE is better than PWP AE, SRPs-DDM, and SRPs-ADWIN. ARF-DDM, LB, HTs, and KNN-ADWIN performed better than the proposed model. The relatively high throughput values inversely correlate with the latency of the methods. Their accuracy and F-score values, however, is low when compared to the proposed approach. This shows that even though the proposed

system is slower to classify data samples in microseconds, it is able to detect DDoS attacks better than the other methods in the presence of concept drift.

Figure 3 and Table 3 show the experimental results of the proposed model and the other methods considered in this research while using CIIoT2023 dataset. In terms of accuracy, the proposed model, i.e., AUWPAE, has achieved 99.33% accuracy, which is the highest when compared to the other methods in this experiment. Moreover, in terms of precision, recall, and F1-score performance metrics, AUWPAE has achieved 98.88%, 96.53%, and 97.98%, respectively, which are better than the other methods. These results indicate that the proposed approach is able to detect DDoS attacks more accurately than the other methods in the presence of concept drifts.

The latency of AUWPAE is 1.29 ms. AUWPAE is relatively slow when it is compared with HTs, ARF-ADWIN, KNN-ADWIN, and LB, and fast when compared with PWPAE, SRPs-DDM, and SRPs-ADWIN. LB is the fastest method but has low accuracy and a low F1-score when compared with AUWPAE. The higher latency value of AUWPAE is mainly attributed to the ensemble algorithm. However, this has given it the advantage to detect DDoS attacks accurately. This result is also consistent with the IOTID20 dataset.

The throughput of the proposed model, the AUWPAE, achieved 773 samples/second, making it better than ARF-EDDM, SRPs-ADWIN, SRPs-DDM, and PWPAE. However, it performed worse than ARF-ADWIN, HTs, and KNN-ADWIN did.

The AUWPAE showed better accuracy and F1-score values for both datasets while taking relatively more time (in ms) to process the data. This indicates that AUWPAE is able to correctly classify normal and DDoS attacks accurately while there are three or more concept drifts in a short period of time. In real-world contexts, however, concept drifts would happen less frequently. Hence, the AUWPAE is expected to perform better in a real-world scenario. Even though AUWPAE is relatively slow, we believe that the accuracy of the classification plays an important role in making a decision to protect an IoT system. Moreover, the latency of AUWPAE is less than the latency requirement of IoT production applications [38].

6. Proposed Solution Deployment Location

This section examines the potential deployment and expected performance of the proposed adaptive online DDoS attack detection framework. IoT endpoints are commonly used for IoT data stream collection, IoT edge servers are used for preliminary for DDoS attack detection analytics, and IoT cloud servers are used for comprehensive DDoS attack detection. The proposed AUWPAE model could be deployed in the IoT cloud, edge servers for IoT DDoS attack detection, and the control server located at the edge layer. The first option is deploying it in IoT edge devices, which provide quick data processing to reduce the size for long-distance data transfer but typically have limited computational capacity. Edge computing allows the detection of local DDoS attacks on edge servers or control servers at the edge layer. Control servers can perform preliminary and fundamental IoT DDoS attack detection jobs locally, including data pre-processing and feature selection. Deploying high-performance computing equipment at the IoT edge layer will significantly minimize latency. Hence, the proposed deployment location considers the DDoS attack detection solution at THE IoT edge using high-performance computing equipment.

The other option could be deploying it in IoT cloud servers for DDoS attack detection. IoT cloud servers often include several cloud machines with high computational power and resources, allowing them to use cloud computing to carry out complex DDoS attack detection activities. However, deploying the proposed framework on the cloud server poses a high risk to the privacy of the data. Additionally, there is a delay in the classification of IoT data due to requests and responses to the central cloud server. Figures 4 and 5 show the proposed framework's deployment location at edge servers and in the IoT cloud, respectively.

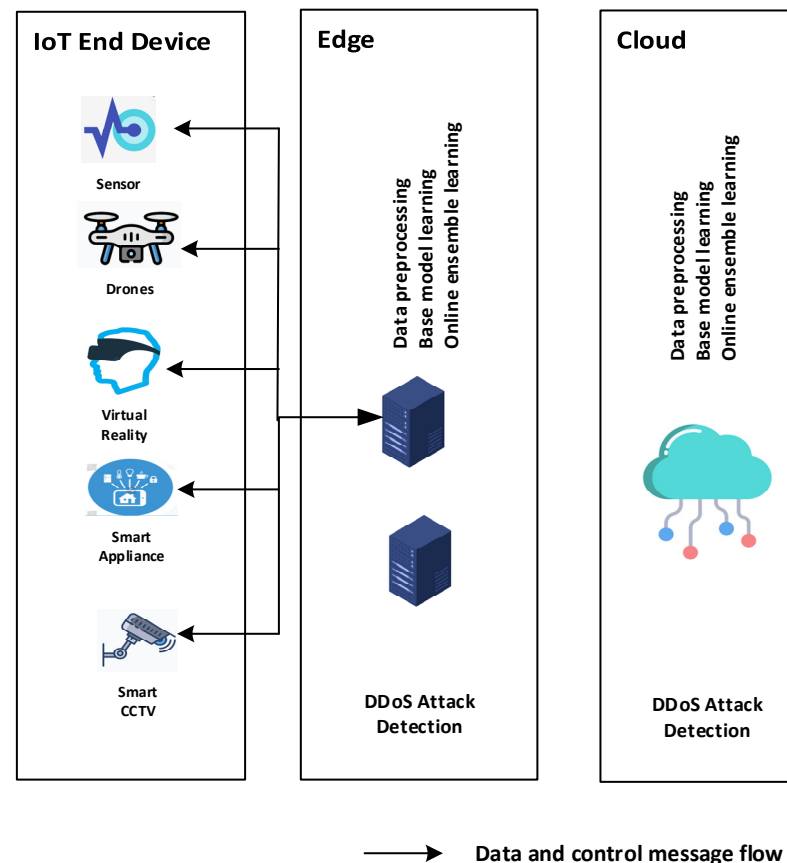


Figure 4. Proposed framework deployment location at IoT Edge Server.

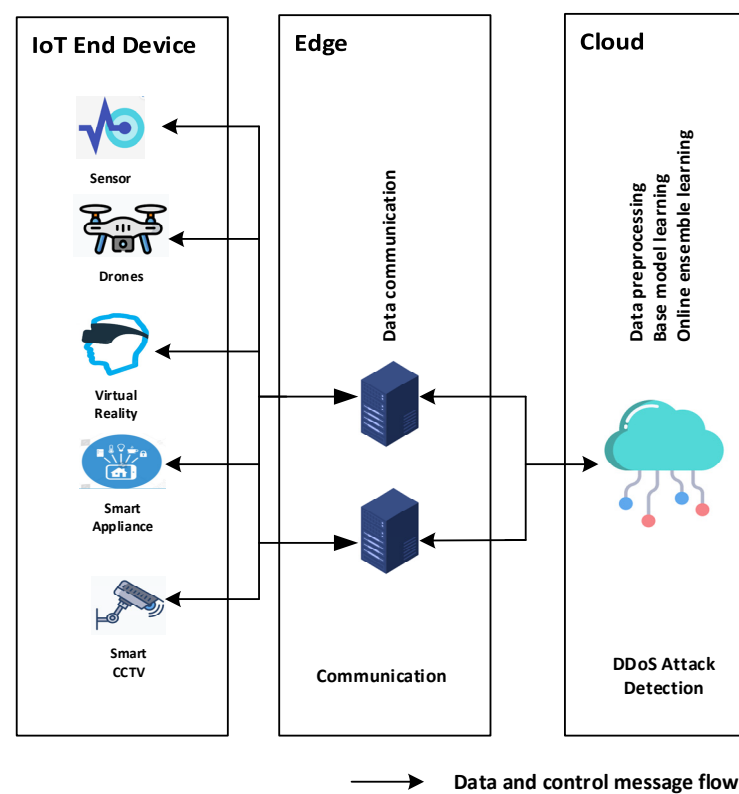


Figure 5. Proposed framework's deployment location in IoT cloud.

7. Conclusions

IoT DDoS attack detection solutions are usually developed to protect IoT systems from DDoS attacks using IoT data stream analytics. However, IoT data are usually dynamic and could have concept drifts. This paper provides a novel Accuracy Update Weighted Probability Averaging Ensemble (AUWPAE) approach to detect concept drift and perform zero-day DDoS detection. We evaluated the proposed model using the IoTID20 and CICIoT2023 datasets with benign and DDoS traffic data that had concept drifts. The results show that AUWPAE achieved better accuracies of 99.54% and 99.33% for the respective datasets when compared with those of the other eight models. This result indicates that the proposed adaptive online DDoS attack detection framework, which uses AUWPAE is able to detect DDoS attacks in the presence of concept drifts. In this paper, we also presented the IoT DDoS attack detection solution deployment framework for IoT systems.

As part of future work, we plan to implement the two deployment scenarios of the proposed approach described in Section 6 in a real-world setting. We also plan to further investigate the use of other algorithms as base learners.

Author Contributions: Conceptualization, Y.K.B., S.L.A. and H.M.M.; methodology, Y.K.B.; validation, Y.K.B.; formal analysis, Y.K.B.; investigation, Y.K.B. and S.L.A.; writing—original draft, Y.K.B.; writing—review and editing, Y.K.B., S.L.A. and H.M.M.; supervision, S.L.A. and H.M.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Otoun, Y.; Liu, D.; Nayak, A. DL-IDS: A deep learning-based intrusion detection framework for securing IoT. *Trans. Emerg. Telecommun. Technol.* **2022**, *33*, e3803. [\[CrossRef\]](#)
- Singh, J.; Pasquier, T.; Bacon, J.; Ko, H.; Eysers, D. Twenty Security Considerations for Cloud Supported Internet of Things. *IEEE Internet Things J.* **2016**, *3*, 269–284. [\[CrossRef\]](#)
- Hussain, F.; Abbas, S.G.; Husnain, M.; Fayyaz, U.U.; Shahzad, F.; Shah, G.A. IoT DoS and DDoS Attack Detection using ResNet. In Proceedings of the 23rd IEEE International Multi-Topic Conference, INMIC2020, Bahawalpur, Pakistan, 5–7 November 2020; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020. [\[CrossRef\]](#)
- Raza, S.; Wallgren, L.; Voigt, T. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Netw.* **2013**, *11*, 2661–2674. [\[CrossRef\]](#)
- Pajila, P.J.B.; Julie, E.G. Detection of DDoS Attack Using SDN in IoT: A Survey. In *Lecture Notes on Data Engineering and Communications Technologies*; Springer: Cham, Switzerland, 2020; Volume 33, pp. 438–452. [\[CrossRef\]](#)
- Ullah, I.; Mahmoud, Q.H. Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks. *IEEE Access* **2021**, *9*, 103906–103926. [\[CrossRef\]](#)
- Chen, Y.-W.; Sheu, J.-P.; Kuo, Y.-C.; Van Cuong, N. Design and Implementation of IoT DDoS Attacks Detection System based on Machine Learning. In Proceedings of the 2020 European Conference on Networks and Communications (EuCNC), Dubrovnik, Croatia, 15–18 June 2020.
- Lai, T.T.; Tran, T.P.; Cho, J.; Yoo, M. DoS attack detection using online learning techniques in wireless sensor networks. *Alex. Eng. J.* **2023**, *85*, 307–319. [\[CrossRef\]](#)
- Bifet, A.; Gavalda, R. Learning from time changing data with adaptive windowing. In Proceedings of the 7th SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics Publications, Minneapolis, MN, USA, 26–28 April 2007; pp. 443–448. [\[CrossRef\]](#)
- Baena-García, M.; del Campo-Ávila, J.; Fidalgo, R.; Bifet, A.; Gavalda, R.; Morales-Bueno, R. Early Drift Detection Method. 2006. Available online: <https://www.researchgate.net/publication/245999704> (accessed on 10 January 2024).
- Gomes, H.M.; Bifet, A.; Read, J.; Barddal, J.P.; Enembreck, F.; Pfahringer, B.; Holmes, G.; Abdessalem, T. Adaptive random forests for evolving data stream classification. *Mach. Learn.* **2017**, *106*, 1469–1495. [\[CrossRef\]](#)
- Gomes, H.M.; Read, J.; Bifet, A. Streaming random patches for evolving data stream classification. In Proceedings of the IEEE International Conference on Data Mining, ICDM, Beijing, China, 8–11 November 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 240–249. [\[CrossRef\]](#)
- Losing, V.; Hammer, B.; Wersing, H. KNN classifier with self adjusting memory for heterogeneous concept drift. In Proceedings of the IEEE International Conference on Data Mining, ICDM, Barcelona, Spain, 12–15 December 2016; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2017; pp. 291–300.

14. Attota, D.C.; Mothukuri, V.; Parizi, R.M.; Pouriyeh, S. An Ensemble Multi-View Federated Learning Intrusion Detection for IoT. *IEEE Access* **2021**, *9*, 117734–117745. [\[CrossRef\]](#)
15. Nguyen, T.D.; Rieger, P.; Miettinen, M.; Sadeghi, A.-R. Poisoning Attacks on Federated Learning-based IoT Intrusion Detection System. In Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS) 2020, San Diego, CA, USA, 23–26 February 2020. [\[CrossRef\]](#)
16. Cheng, Y.; Lu, J.; Niyato, D.; Lyu, B.; Kang, J.; Zhu, S. Federated transfer learning with client selection for intrusion detection in mobile edge computing. *IEEE Commun. Lett.* **2022**, *26*, 552–556. [\[CrossRef\]](#)
17. Zainudin, A.; Ahakonye, L.A.C.; Akter, R.; Kim, D.-S.; Lee, J.-M. An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IoT Networks. *IEEE Internet Things J.* **2023**, *10*, 8491–8504. [\[CrossRef\]](#)
18. Kumar, D.; Pateriya, R.K.; Gupta, R.K.; Dehalwar, V.; Sharma, A. DDoS Detection using Deep Learning. *Procedia Comput. Sci.* **2023**, *218*, 2420–2429. [\[CrossRef\]](#)
19. Gama, J.; Medas, P.; Castillo, G.; Rodrigues, P. LNAI3171—Learning with Drift Detection. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2004.
20. Bayram, F.; Ahmed, B.S.; Kassler, A. From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowl.-Based Syst.* **2022**, *245*, 108632. [\[CrossRef\]](#)
21. Wang, P.; Jin, N.; Davies, D.; Woo, W.L. Model-centric transfer learning framework for concept drift detection. *Knowl.-Based Syst.* **2023**, *275*, 110705. [\[CrossRef\]](#)
22. He, J.; Mao, R.; Shao, Z.; Zhu, F. Incremental Learning in Online Scenario. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
23. Kumar, S.; Singh, R.; Khan, M.Z.; Noorwali, A. Design of adaptive ensemble classifier for online sentiment analysis and opinion mining. *Peer J. Comput. Sci.* **2021**, *7*, e660. [\[CrossRef\]](#) [\[PubMed\]](#)
24. Lu, J.; Liu, A.; Dong, F.; Gu, F.; Gama, J.; Zhang, G. Learning under Concept Drift: A Review. *IEEE Trans. Knowl. Data Eng.* **2020**, *31*, 2346–2363. [\[CrossRef\]](#)
25. Brzezinski, D. Block-Based and Online Ensembles for Concept-Drifting Data Streams. Ph.D. Thesis, Poznan University of Technology, Poznan, Poland, 2015.
26. Sun, Y.; Shao, H.; Zhang, B. Ensemble based on Accuracy Diversity Weighting for Evolving Data Streams. *Int. Arab. J. Inf. Technol.* **2022**, *19*, 90–96. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Brzezi, D.B.; Stefanowski, J. Accuracy Updated Ensemble for Data Streams with Concept Drift. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2011.
28. Bifet, A.; Holmes, G.; Pfahringer, B. Leveraging Bagging for Evolving Data Streams. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 2010.
29. Yang, L.; Manias, D.M.; Shami, A. PWPAE: An Ensemble Framework for Concept Drift Adaptation in IoT Data Streams. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021. [\[CrossRef\]](#)
30. Saheed, Y.K.; Abiodun, A.I.; Misra, S.; Holone, M.K.; Colomo-Palacios, R. A machine learning-based intrusion detection for detecting internet of things network attacks. *Alex. Eng. J.* **2022**, *61*, 9395–9409. [\[CrossRef\]](#)
31. Wang, H.; Wei, Q.; Xie, Y. A Novel Method for Network Intrusion Detection. *Sci. Program.* **2022**, *2022*, 1357182. [\[CrossRef\]](#)
32. Liu, W.; Zhu, C.; Ding, Z.; Zhang, H.; Liu, Q. Multi-class imbalanced and concept drift network traffic classification framework based on online active learning. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105607. [\[CrossRef\]](#)
33. Canadian Institute for Cybersecurity. CICIOT Dataset 2023. 2023. Available online: <https://www.unb.ca/cic/datasets/index.html> (accessed on 29 August 2023).
34. Maniriho, P.; Niyigaba, E.; Bizimana, Z.; Twiringiyimana, V.; Mahoro, L.J.; Ahmad, T. Anomaly-based Intrusion Detection Approach for IoT Networks Using Machine Learning. In Proceedings of the 2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 17–18 November 2020; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; pp. 303–308. [\[CrossRef\]](#)
35. River. August 2023. Available online: <https://riverml.xyz/0.21.0> (accessed on 15 August 2023).
36. Ullah, I.; Mahmoud, Q.H. A Scheme for Generating a Dataset for Anomalous Activity Detection in IoT Networks. In *Lecture Notes in Computer Science*; Including Sub series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics; Springer: Cham, Switzerland, 2020; pp. 508–520. [\[CrossRef\]](#)
37. Rustam, F.; Jucut, A.D. Malicious traffic detection in multi-environment networks using novel S-DATE and PSO-D-SEM approaches. *Comput. Secur.* **2024**, *136*, 103564. [\[CrossRef\]](#)
38. Schulz, P.; Matthe, M.; Klessig, H.; Simsek, M.; Fettweis, G.; Ansari, J.; Ashraf, S.A.; Almeroth, B.; Voigt, J.; Riedel, I.; et al. Latency Critical IoT Applications in 5G: Perspective on the Design of Radio Interface and Network Architecture. *IEEE Commun. Mag.* **2017**, *55*, 70–78. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.