مجلة دجلة • المجلد ٦ ،العدد ٢ (٢٠٢٣)، ص: ١ - ١٦
ISSN: 2222-6583

تطوير أنظمة كشف التسلل باستخدام مفهوم الانجراف في تدفق البيانات

Development of intrusion detection systems using the concept of drift in the data stream

نورا صباح سلمان / جامعة بغداد / كلية العلوم / قسم علوم الحاسوب
nora.sabbah1201a@sc.uobaghdad.edu.iq

المستخلص

الغرض من هذه الدراسة هو تقديم إستراتيجية تكيفية لاكتشاف التسلل عبر الإنترنت تستخدم التعلم الموجه نحو التدفق للتكيف مع مفهوم الانجراف في بيئة العالم الحقيقي. يتم تقييم الطريقة باستخدام مجموعة بيانات CIC-IDS 2018. الطريقة: يلغي الحل المقترح الحاجة إلى إعادة تدريب النموذج باستمرار باستخدام سلسلة من الخوارزميات لاكتشاف التغيير في تدفق البيانات والاستجابة لاكتشاف الانجراف في البيانات المتدفقة. ينتج عن هذا تكيف سريع مع التدخلات غير المتوقعة.

Abstract

One of the principal research subjects for networks that handle external threats is intrusion detection. because internet security is a crucial issue today. A fresh method for securing current computers and data networks is intrusion detection. An intrusion detection system (IDS) is a software program that monitors for illegal purposes and unauthorized system access. The work that has already been done on intrusion detection systems that use Although data mining and machine learning are useful, they necessitate the training of static batch classifiers to detect attacks independent of the time-varying features of the periodic data stream.

The purpose of this study is to offer an adaptive strategy for online intrusion detection that uses stream-oriented learning to adjust to concept drift in a real-world setting. The method is assessed using the CIC-IDS 2018 dataset. Method: The proposed solution eliminates the need to continually retrain the model by using a series of algorithms for detecting change in a data stream and responding to drift detection in the streamed data. This results in quick adaptation to unforeseen intrusions.

1.Introduction

More data has been generated over the last two years than in the entire preceding span of human history[1]. Data is expanding more quickly than ever, and in 2025, the quantity of data created annually is predicted to surpass 180 zettabytes, according to IDC. Data never sleeps;

every 60 seconds, enormous amounts of data are produced by several apps on the internet. Over 2,000,00 search queries are sent to Google per minute, and 204,166,667 emails are sent. There has been a noticeable increase in video data, about 300 hours of content uploaded to YouTube every minute [1].

These statistics show how quickly data is expanding and creating the digital cosmos. One of the key characteristics of big data is speed or velocity. The growing popularity of data streams is also a result of hardware technology improvement. Many routine daily activities, such using a credit/debit card or a smartphone, result in the automated generation of data. These activities frequently include a huge number of people, which generates enormous data streams. In a similar vein, massive volumes of image, audio, video, as well as textual streams are frequently present in telecommunications and social media. Since the data mining methods created so far are better suited for static data, Mining data streams is currently a major concern for the academic community. Learning from data streams is another significant problem because the data is changing or evolving over time and is only available for relatively brief periods of time. Data streams differ from non-streaming data in terms of its temporal characteristics (real-time data production), and as a result, new classification, clustering, pattern mining, and effective evaluation parameters are required. The issue of classification in data streams is the main topic of this research. Traditional classification algorithms cannot be applied with streaming data (one look, no random access) because to resource constraints (processing time, memory, and single scan of the data). As a result, another research issue is the development of novel learning algorithms, such as incremental or ensemble learning, to classify data streams. Because the distribution of the dataset may change with time while the model is being trained, classifying streaming data is more complex than classifying stable data. The phrase concept drift is used to indicate this occurrence. Concept drift must be taken into consideration when building any streaming model. As a result, building a training model in streaming and dynamic circumstances is frequently extremely important. Due of the exploding amount of data, research in this area has just begun to receive increasing attention [1].

Methods: Malicious intrusions into networks have risen, making the IDS (Intrusion Detection System) architecture essential for more secure systems. In recent times, network abnormality detection has become more and more dependent on machine learning techniques. However, the currently accessible works do not examine the variation in data over time, which limits their capacity to identify novel types of incursion. Therefore, we propose an IDS with a concept drift-based incremental learning using (ADWIN,CUSUM, DDM,EDDM, Ensemble drift detection, HDDM) as detection algorithms and (active classifier, naivebyes, single classifier drift, adaptive random forest, heoffding adaptive tree) as handling algorithm to account for unanticipated changes in the status data's statistical properties over time.

The following are the paper's significant contributions:

- High-performance intrusion detection system depending on streamed data which is subject to concept drift.

- Incremental IDS system with adaptable classification model modification, reaching high
- IDS with dynamic adaptability against unknown intrusions in actual time.

The remainder of the paper is organized as follows. Part 2 examines the existing literature on stream data mining . The research approach is discussed in Section 3. Section 4 explains the findings results . Section 5 provides a summary of the paper and future work.

2- Related work

A variety of works have been done in-stream data mining to detect and classify threats. Methods described in earlier studies are covered in this section. Several datasets are used in these studies to detect drift, classify threats, and measure classification accuracy.

1. **Gama, J and et al 2013**, Describe how adaptive learning works, categorize current concept drift management software, outline the most typical, different, and popular methodologies and algorithms, discuss how adaptive algorithms be evaluated, and provide several practical applications. The survey investigates the different facets of concept drift in an integrated way so as to report on the existing dispersed state-of-the-art. As a result, it attempts to provide a full introduction to the concept of drift adaptability to researchers, industry analysts, and practitioners. [4].

   Janardan et al.2017، Show that due to single pass, in memory, and fast access times, stream data is incompatible with classification methods that perform well with stationary datasets. The numerous categorization platforms and algorithms described in this study are helpful when the datasets are changing and there is an issue with idea drift. In this study, tools that are currently in use in the big data analytics sector are also listed. Researchers studying streaming data also discovered benchmark datasets. The comparative examination of the various data streaming frameworks is covered in this essay. More study is required to adequately measure and analyze all of the available possibilities because many of these tools are still in their infancy. In contrast to distributed Nave, only VHT, CluStream, and SAMAO effectively applies adaptive model rules[1].

   Pradheep D and et al 2020 , Three classifiers are utilized to identify the intrusion after the suggested HDDM approach, based on Hoeffding inequality, is employed to detect an anomaly in the data chunks. The suggested model's transfer learning mechanism exhibits greater accuracy than incremental learning. In the second learning iteration, the transfer learning applied the knowledge from the prior learning. The intrusion detection method is demonstrated by the evaluation results using the NSL-KDD dataset. Therefore, the intrusion detection system that contains sudden or abrupt idea drift can also apply transfer learning [2].

   Jie Lu and et al 2020 , demonstrate that many hypothesis test approaches have recently emerged, despite the fact that Methods for detecting drift based on error rates and data distribution continue to dominate concept drift detection research; All drift detection techniques can answer "When" in terms of concept drift knowledge, but very few techniques can also respond to "How" and "Where"; Recent breakthroughs in idea drift

adaptation have seen a rise in the importance of adaptive models and ensemble methodologies. The development of retraining models that explicitly detect drift, however, has halted; The majority of drift detection and adaptation algorithms in use today anticipate that verification will take a long time or that the ground true label will be accessible after classification or prediction. Unsupervised or semi-supervised drift detection and adaptation have received very little research attention. There is no thorough examination of actual-world data streams from the concept drift side, such as the drift occurrence time, the severity of drift, and the drift regions. However, several computational intelligence approaches, such as fuzzy logic and competency model, have been utilized in concept drift. The significance of managing idea drift is being recognized by an increasing number of other academic fields, particularly in the big data community [3].

Nathan Martindale and et al 2020, investigated the trade-offs in performance and run-time of different machine learning methods for network intrusion detection. All algorithms in the tests were specifically trained online using streaming traffic connection data, which limits the amount of data that can be trained on at once. Three proposed heterogeneous ensembles as well as a number of individual algorithms and homogeneous ensemble techniques were examined. Despite having a generally longer run-time, the ensembles outperformed the individual base learners overall. The best homogeneous ensemble model (the ARF) and one of the proposed heterogeneous ensembles, HAT + ARF, both demonstrated equivalent accuracy performance. Future study will be conducted to determine how the ARF's structure can be changed to provide the best performance with the lowest run-time. The best-performing ensemble consists of a single HAT with 10 HTs, while a smaller HAT by itself probably performs better with different numbers of base learners. For example, by combining five HATs with an ARF of five HTs, a greater balance or weighting between the various methods of windowing data would be achievable. It's possible that adding an autoencoder and smaller ARFs could improve performance even more, similar to how some of the linked publications proposed using an ensemble of one-class anomaly detection algorithms. [6]

Sugandh Setha and et al 2021، confirmed that Most streaming data applications want a speedy response, which necessitates re(training) an algorithm with the most latest data. The majority of current AI-based intrusion detection models are trained using static data. In contrast, data in an IDS arrives in streams, and the distribution of the data may shift over time due to changing attack patterns, results in concept drift. Furthermore, in order for the IDS to be effective, it must be able to change over time in order to identify new attack classes. Static batch learning models perform poorly in such cases since they age and must be updated over time. This work use adaptive randomization to circumvent the aforementioned issues. [17].

The major part of the proposed method in the literature is based on old datasets or static datasets. To address the aforementioned research gaps, this work It uses a set of drift detection algorithms and a set of drift adaptation algorithms to categorize threats in stream data using the most recent CIC-IDS 2018 dataset

3. Data Preprocessing

3.1.1. Dataset description

The CIC IDS 2018 dataset is used for the proposed investigation. 14 contemporary assaults are included in the sizable dataset known as CIC IDS 2018. Communications Security Establishment (CSE) and the Canadian Institute for Cybersecurity published the CIC IDS dataset (CIC). The dataset has 16 million samples and 80 features[8].

3.1.2 Data cleaning

Data cleaning is a first stage in data preprocessing procedures that is used to locate missing values. Raw data may contain incomplete records, noisy values, outliers, and inconsistent data In this research, the missing values and the infinity value in the dataset were eliminated and zeros were put in place of them, because these values affect the prediction of the model.

3.1.3 feature selection

In this step, irrelevant, tangentially relevant, or redundant features or dimensions are removed ,Three categories of feature selection techniques can be made:

- Filtering techniques: Data-related metrics like crowding or reparability are used to make the selection.
- Embedded techniques: The classifier construction includes the best features subset.
- Wrapper techniques: The learning algorithm is dependent on the selection criterion because it is a component of the fitness function

In this study, the filter strategy is used prior to the classification procedure. The filter technique is independent of the learning algorithm and is computationally quick, simple, and scalable. The filter technique allows features to be selected once and then utilized as inputs for numerous classifiers. This work employs Gain Ratio (GR) and Correlation-based Feature Selection (CFS) for feature selection [9].

A- Gain Ratio

A decision tree is a simple structure in which terminal nodes reflect decision results and non-terminal nodes represent tests on one or more attributes. The test attribute is selected using the information gain metric at each decision tree node. The information gain metric encourages the selection of attributes with a wide range of possible values. C4.5 [6, 7] enhanced the basic decision tree induction technique ID3 [5]. To overcome this prejudice, C4.5, the successor to ID3, employs a gain ratio information gain extension. J4.8 is the version of C4.5 utilized by the WEKA [8] classifier package. J4.8 [9] was used to identify the key features.

B- Correlation based Feature Selection (CFS)

The downside of univariate filters, such as information gain, is that they do not consider how distinct traits interact with one another. This is addressed with multivariate filters such as CFS. CFS considers the individual predictive power of each characteristic as well as the degree of dependency between them when calculating the value of a subset of attributes. Correlation coefficients are used to estimate the inter-correlations between features as well as the correlations between a subset of attributes and class. The relevance of a set of features grows as feature-class correlation increases and decreases as inter-correlation increases. CFS is typically used in conjunction with other search techniques such as forward selection, backward elimination, bi-directional search, best-first search, and genetic search[9].

3.2. Training the Model

Concept drift is the alteration of the underlying data generating process. Concept drift in the classification context denotes fluctuation in the statistical characteristics of the target variable. The quantity the researcher wants to predict is referred to as the concept, and the target variable is the one for which the model is attempting to make a time-based prediction. The distribution that gives rise to the items in the data stream can alter with time, as was already indicated. The proposed model uses (ADWIN,CUSUM, DDM, EDDM, Ensemble drift detection, HDDM) as a detection algorithms and (active classifier, naive byes, single classifier drift, adaptive random forest, and heoffding adaptive tree) as a classifier. This addresses the issue of concept drift in intrusion detection systems.

1- ADWIN

A sliding window algorithm called ADWIN can find drifts in a data stream. To detect concept drift, this approach relies on maintaining statistics for a variable-sized window. The statistics windows are cut at various points, and the mean of various statistic measures over various windows is compared to determine the window size. If the variance between the average statistics exceeds a predetermined threshold value, a drift is recognized [5].

2- Cusum

The Sequential Probability Ratio Test serves as the foundation for CUSUM (SPRT). The key insight is that the chance of detecting a specific subsequence before and after the distribution changes at time point w differs noticeably. [10]

3- DDM

DDM's basic principle, which regulates the algorithm's online error rate, is likewise quite straightforward (the online error rate of the control algorithm). Data distribution, histogram, histogram; The model's error rate will rise when the probability distribution shifts. The online control form contained the error.

Two error rate thresholds, one for warning and one for drift, will be set by DDM. The error rate hits the alert level when the data with is inserted into the sample data, signaling the introduction of a modification to the sample probability distribution. If the mistake rate is not reduced by the subsequent input data and reaches a certain level when the dth data is entered, The probability distribution of the sample has changed, it is determined, if the drift value is

set. The model will learn from the data afterward in order to adjust to the new sample data; if the successive input data decreases the error rate, it indicates false alarm [11]

4- EDDM

The essential concept is to evaluate the distance between two error classifications rather than just the number of errors. When the learning process is learning, it will improve predictions and increase the distance between two errors. We can compute the average distance (p 0 I and standard deviation between two errors (s 0 I). When p 0 I + 2• s 0 I reaches its maximum value, we record the values of p 0 I and s 0 i. (obtaining p 0 max and s 0 max). Thus, The value of p 0 max + 2 • s 0 max corresponds to the maximum of the distribution of distances between errors. This moment is reached when the model that it is being induced best approximates the present ideas in the dataset [12]

5- Ensemble drift detection

Ensemble learning has evolved as a favored technique among academics tackling a variety of ML issues. Because each dataset comprises different drifts with varying degrees of severity and speed, an ensemble learning technique is utilized to recognize different types of drifts and classify data accordingly. Ensemble learning approaches integrate many machine learning classifiers and utilize various voting procedures to produce better results[14].

3.3 concept drift adaptation algorithm

After drift detection is performed using drift detection algorithms, the model is adapted and trained on the new data using the algorithms mentioned in this section.

1- Single classifier with data distribution monitoring

By comparing discrepancies across successive batches of data, this type of approach finds changes. A approach that compares changes between two successive batches of data is presented by Sobhani and Beigy (2011). The goal is to locate each instance's nearest neighbor in the previous batch of data, then compare their related class labels. The degree of drift (DoF) alert is activated when there is a considerable increase in the value, and the authors employ the heom distance to measure the similarity between data batches. [13]

2- active classifier

Active learning emphasizes learning a precise model with the fewest number of labels possible. Active learning is further complicated by streaming data since classifiers must adjust as the data distribution changes over time (concept drift). The most uncertain cases, which are often grouped around the decision border, are the focus of traditional active learning systems. Changes won't be seen if they don't happen close to the boundary, and classifiers won't be able to respond. This class includes four concept-drift-aware active learning algorithms for streaming data. They are founded on a combination of randomization, fixed uncertainty, dynamic labeling effort allocation across time, and search space randomization [ZBPH]. Additionally, it implements an adaptation of [CGZSelective]'s Sampling method.

3- Hoeffding Adaptive Tree

Since we preserve the statistics data required for estimators, replacing frequency statistics counters with them does not require a window to store examples. Instead, the substitution of alternative subtrees is checked using a change detector. [13]

4- Adaptive Random Forests

With data stream learning, multiple passes over the input data are not possible. Hence, the capacity of Random Forests to handle streaming data is dependent on two factors: (1) a proper online bootstrap aggregating process, and (2) restricting each leaf split decision to a subset of features. The second condition is addressed by modifying the base tree induction process, which effectively limits the set of features taken into account for additional splits to a random subset of size m, where m M specifies the total number of features. [16]

5-naive Bayes classifiers

In statistics, the "naive Bayes classifiers" family of basic "probabilistic classifiers" is based on the application of Bayes' theorem with strong (naive) independence assumptions between the features (see Bayes classifier). These are among of the most basic Bayesian network models[1, but when combined with kernel density estimation, they can achieve high levels of accuracy. Because the number of parameters required for naive Bayes classifiers is proportional to the number of variables (features/predictors) in a learning problem, they are tremendously scalable. Maximum-likelihood training can be conducted by simply evaluating a closed-form expression, which takes linear time, rather than utilizing an expensive iterative approximation, as is the case for many other types of classifiers. [13]

4. Results and Discussions

Accuracy, Kappa, and TIME Measure are three performance indicators used to evaluate the proposed model .proposed model used a set of algorithm for drift detection (DDM,EDDM,ADWIN, Ensemble drift detection, HDDM)with set of classifier(active classifier, naive byes, single classifier drift, adaptive random forest, and heoffding adaptive tree) and calculate the result using tables to improve IDS depending on accuracy and time ,The results of using algorithms to analyze the concept of drift before making feature selection are shown in Table No. 1 and Table No. 2 and Table No. 3 show the results of using algorithms for analyzing the concept of drift using feature selection with Gain ratio filter, and Table No. 4 uses correlation attribute Eval filter with 21 feature selection , The equations below represent an evaluation of the accuracy of the model

$$\text{Accuracy} = \frac{\text{TruePositive} + \text{TrueNegative}}{\text{TruePositive} + \text{FalseNegative} + \text{TrueNegative} + \text{FalsePositive}} \quad (1)\ [5]$$

Accuracy is the percentage of samples that were correctly classified.

Kappa = (total accuracy – random accuracy) / (1- random accuracy) (2) [5]

Table(1) results Before feature selection

Detection algorithm	Retraining algorithm /active Classifier		
	Accuracy	Kappa	Time
ADWIN	4.60	4.60	0.55
CUSUM	24.91	24.91	0.64
DDM	41.42	41.42	1.79
EDDM	41.42	41.42	0.86
Ensemble drift detection	41.42	41.42	2.42
HDDM	41.42	41.42	0.78
Retraining algorithm / naïve Bayes			
ADWIN	60.50	66.50	0.42
CUSUM	86.35	86.35	0.57
DDM	88.28	88.28	1.96
EDDM	88.28	88.28	0.71
Ensemble drift detection	89.28	89.28	2.14
HDDM	88.28	88.28	0.68
Retraining algorithm / single classifier drift			
ADWIN	4.60	4.60	0.59
CUSUM	86.35	86.35	0.57
DDM	88.28	88.28	1.53
EDDM	88.28	88.28	0.91
Ensemble drift detection	89.28	89.28	2.17
HDDM	88.28	88.28	0.83
Retraining algorithm / adaptive random forest			
ADWIN	60.50	60.50	4.56
CUSUM	94.14	94.14	3.21
DDM	97.07	97.07	5.89
EDDM	97.07	97.07	3.49
Ensemble drift detection	94.14	94.14	2.17
HDDM	94.14	94.14	2.06
Retraining algorithm /heoffding adaptive tree			

ADWIN	52.68	52.68	0.55
CUSUM	94.14	94.14	3.21
DDM	94.14	94.14	1.55
EDDM	94.14	94.14	0.58
Ensemble drift detection	94.14	94.14	2.17
HDDM	94.14	94.14	0.81

From Table 1, we note that the ARF algorithm achieved good results, reaching 97% with each of the DDM and EDDM algorithms, with a relatively long execution time compared to the rest of the algorithms. The rest of the algorithms achieved lower results compared to the ARF algorithm, ranging from 4% to 94% with an execution time of up to 4 seconds

Table(2) : Concept drift algorithm with feature selection usingGR (Gain ratio filter)selected 20 attributes

Detection algorithm	Retraining algorithm /active Classifier		
	Accuracy	Kappa	Time
ADWIN	2.25	2.25	0.58
CUSUM	24.91	24.91	0.43
DDM	41.42	41.42	0.37
EDDM	41.42	41.42	0.64
Ensemble drift detection	41.42	41.42	0.67
HDDM	41.42	41.42	0.89
Retraining algorithm / naïve Bayes			
ADWIN	2.25	2.25	0.81
CUSUM	85.35	85.35	0.42
DDM	41.42	41.42	0.42
EDDM	88.28	88.28	0.37
Ensemble drift detection	91.21	91.21	0.52
HDDM	91.21	91.21	0.86
Retraining algorithm / single classifier drift			
ADWIN	59.66	59.66	0.61
CUSUM	85.35	85.35	0.54
DDM	88.28	88.28	0.42
EDDM	88.28	88.28	0.38
Ensemble drift detection	91.21	91.21	0.71

مجلة دجلة • المجلد ٦ ،العدد ٢ (٢٠٢٣)، ص: ١ - ١٦
ISSN: 2222-6583

HDDM	91.21	91.21	0.86
Retraining algorithm / adaptive random forest			
ADWIN	57.33	57.33	2.97
CUSUM	94.14	94.14	1.12
DDM	**97.07**	**97.07**	1.64
EDDM	**97.07**	**97.07**	1.64
Ensemble drift detection	94.14	94.14	1.26
HDDM	94.14	94.14	2.34
Retraining algorithm /heoffding adaptive tree			
ADWIN	52.58	52.58	0.48
CUSUM	94.14	94.14	0.45
DDM	41.42	41.42	0.43
EDDM	94.14	94.14	0.47
Ensemble drift detection	94.14	94.14	1.26
HDDM	94.14	94.14	0.83

Table No. 2, which represents the selection of 20 features using Gain ratio feature selection, achieved results close to the results in Table No. 1 in terms of accuracy, but in terms of time, we notice a significant improvement in the execution time, as the execution time decreased significantly

Table 3 : Concept drift algorithm with feature selection GR(Gain ratio filter)selected 36 attributes

Detection algorithm	Retraining algorithm /active Classifier		
	Accuracy	Kappa	Time
ADWIN	1.96	1.96	0.60
CUSUM	41.42	41.42	0.66
DDM	41.42	41.42	0.59
EDDM	41.42	41.42	0.55
Ensemble drift detection	41.42	41.42	0.96
HDDM	41.42	41.42	0.88
Retraining algorithm / naïve Bayes			
ADWIN	51.77	51.77	0.62
CUSUM	41.42	41.42	0.66
DDM	88.28	88.28	0.54
EDDM	88.28	88.28	0.48
Ensemble drift	88.28	88.28	0.85

detection			
HDDM	82.43	82.43	0.72
Retraining algorithm / single classifier drift			
ADWIN	54.67	54.67	0.99
CUSUM	83.43	83.43	0.75
DDM	88.28	88.28	0.60
EDDM	41.42	41.42	0.56
Ensemble drift detection	88.28	88.28	1.11
HDDM	91.21	91.21	0.86
Retraining algorithm / adaptive random forest			
ADWIN	64.68	64.68	5.25
CUSUM	94.14	94.14	1.84
DDM	**97.07**	**97.07**	**2.15**
EDDM	**97.07**	**97.07**	**2.35**
Ensemble drift detection	94.14	94.14	3.17
HDDM	94.14	94.14	2.53
Retraining algorithm /heoffding adaptive tree			
ADWIN	56.26	56.26	0.73
CUSUM	94.14	94.14	0.71
DDM	94.14	94.14	0.55
EDDM	94.14	94.14	2.35
Ensemble drift detection	94.14	94.14	0.94
HDDM	94.14	94.14	0.94

Table No. 3, which represents the selection of 36 features using Gain ratio feature selection, achieved results close to the results in Table No. 2 in terms of accuracy and time.

Table 4 : Concept drift algorithm with feature selection using CSF (correlation attribute Eval) select 21 attributes

Detection algorithm	Retraining algorithm /active Classifier		
	Accuracy	Kappa	Time
ADWIN	6.38	6.38	0.70
CUSUM	41.42	41.42	0.80
DDM	41.42	41.42	0.65
EDDM	41.42	41.42	0.56
Ensemble drift detection	41.42	41.42	0.89

HDDM	41.42	41.42	0.76
Retraining algorithm / naïve Bayes			
ADWIN	62.61	62.61	0.82
CUSUM	82.43	82.43	0.75
DDM	85.35	85.35	0.57
EDDM	85.35	85.35	0.56
Ensemble drift detection	85.35	85.35	0.80
HDDM	85.35	85.35	0.71
Retraining algorithm / single classifier drift			
ADWIN	62.61	62.61	0.95
CUSUM	82.43	82.43	0.73
DDM	85.35	85.35	0.57
EDDM	85.35	85.35	0.54
Ensemble drift detection	85.35	85.35	0.80
HDDM	85.35	85.35	0.76
Retraining algorithm / adaptive random forest			
ADWIN	62.93	62.93	6.72
CUSUM	94.14	94.14	1.93
DDM	**97.07**	**97.07**	**2.25**
EDDM	**97.07**	**97.07**	**2.01**
Ensemble drift detection	94.14	94.14	2.34
HDDM	94.14	94.14	2.14
Retraining algorithm /heoffding adaptive tree			
ADWIN	59.13	59.13	0.19
CUSUM	94.14	94.14	0.68
DDM	94.14	94.14	0.61
EDDM	94.14	94.14	0.54
Ensemble drift detection	94.14	94.14	0.78
HDDM	94.14	94.14	0.73

Table No. 4, which represents the selection of 20 features using (correlation attribute eval), achieved results close to the results in Table No. 2,3 in terms of accuracy and a similar implementation time

4.1 result analysis

By comparing the set of algorithms used to detect concept drift and model retraining algorithms prior to feature selection, we discover that the active classifier algorithm used in drift detection with the model retraining algorithms did not achieve the desired results in terms of accuracy and kappa in classification. In terms of accuracy, kappa, and duration, naive Bayes and single classifier drift with model retraining algorithms got nearly identical results and outperformed the active classifier method. With the EDDM and DDM retraining algorithms, the adaptive random forest produced good results, But, the issue is the time delay. The heoffding adaptive tree method performed well with the CUSUM retraining algorithms DDM, EDDM, and EDDM. All the same, it is less accurate and has a lower kappa than the adaptive random forest method, but it is faster. When the Gain ratio filter was used to select features, 36 attributes were selected using the same previous algorithms in terms of drift detection and retraining the model did not affect the results in terms of accuracy and kappa, but there was an improvement in terms of time as the delay, 20 attributes were selected. In terms of accuracy and kappa, we got the same findings as before, but there was an improvement in terms of precision.

To improve the findings, a second strategy for choosing features was applied. Using correlation attribute Eval, 20 attributes were chosen. We saw an improvement in the outcomes when utilizing the same previous approaches for drift detection and model retraining.

We received the same accuracy and kappa as before, but we also got a latency improvement.

As a consequence, we discovered that the adaptive random forest detection algorithm outperformed the DDM and EDDM retraining algorithms in terms of accuracy and kappa, achieving 97% accuracy and kappa. The latency before making a feature selection was reduced from 5.89 to 1.64 with DDM and 3.49 with EDDM when 20 features were picked using the Gain ratio filter.

5- Conclusion and future work

Most streaming data applications require a quick response, which necessitates re(training) an algorithm with the most recent data. The majority of existing AI-based intrusion detection models are trained on static data. In contrast, data comes in streams in an IDS, and the data distribution may change over time as attack patterns evolve, leading in idea drift. Moreover, for the IDS to perform successfully it needs to evolve itself to be able to detect new threat classes throughout the time. In such cases the static data models performs badly since static batch learning models grow old and has to be updated with time. To overcome the above difficulties, In this paper, a set of algorithms were used to detect drift and a set of algorithms to eliminate drift and make comparisons between the set of algorithms. It was noted that the EDDM algorithm with the ARF with (DDM,EDDM) algorithm achieved the highest

percentage in terms of ACCURACY and in terms of time , accuracy rate 97.07 % ,time 1.64 ,94% with ensemble drift detection and HDDM , HAT algorithm achieved 94% with (CUSUM,DDM,EDDM, Ensemble drift detection ,HDDM) as accuracy

Future work

Doing more tests on new datasets for (intrusion detection systems) such as (CICIDS2019) and Utilizing datasets from various disciplines with varying features to test and enhance the performance of the approaches In this investigation, we assume that cases come sequentially. The framework could be expanded in the future to situations where data is supplied in packages of instances.

Reference

5-Bateman, T., & Snell, S. (2019). Management: Leading & Collaborating in Competitive World, 13e

1-Janardan, Shikha Mehta*(2017):Concept drift in Streaming Data Classification: Algorithms, Platforms and Issues , Information Technology and Quantitative Management

2- y Pradheep D, Gokul R, Naveen V & Vijayarani J (2020) :Anomaly Intrusion Detection based on Concept Drift

3- Jie Lu, Fellow, IEEE, Anjin Liu, Member, IEEE, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang (2020) : Learning under Concept Drift: A Review ,

4-JOAO GAMA, INDRE˙ ZLIOBAIT ˇ E, ALBERT BIFET, MYKOLA PECHENIZKIY, ABDELHAMID BOUCHACHIA(2013): A Survey on Concept Drift Adaptation

5- Sugandh Setha , Gurwinder Singha , Kuljit Kaur Chahala (2021) : Drift-based approach for evolving data stream classification in Intrusion detection system

5- Nathan Martindale , Muhammad Ismail , and Douglas A. Talbert (2020) : Ensemble-Based Online Machine Learning Algorithms for Network Intrusion Detection Systems Using Streaming Data

6- suad A.ALasadi and Wesam S.Bhaya (2017):review of data preprocessing techniques in data mining

7- Amer Abdulmajeed Abdulrahma , Mahmood Khalel Ibrahem (2020) : Intrusion Detection System Using Data Stream Classification

9-Asha Gowda Karegowda, A. S. Manjunath & M.A.Jayara(2010): COMPARATIVE STUDY OF ATTRIBUTE SELECTION USING GAIN RATIO AND CORRELATION BASED FEATURE SELECTION

10- Namitha K, G Santhosh Kumar(2020): CUSUM Based Concept Drift Detector for Data Stream Clustering

11- Jo˜ao Gama1,2 , Pedro Medas1 , Gladys Castillo1,3 , and Pedro Rodrigues1(2013): Learning with Drift Detection

12- Manuel Baena-Garc´ıa1 , Jos´e del Campo-Avila ´ 1 , Ra´ul Fidalgo1 , Albert Bifet2 , Ricard Gavald`a2 , and Rafael Morales-Bueno (2018) : Early Drift Detection Method

13- Imen Khamassi1 , Moamar Sayed-Mouchaweh2 , Moez Hammami1 and Khaled Ghédira1(2013): Ensemble classifiers for drift detection and monitoring in dynamical environments

14- AHMAD ABBASI1 , ABDUL REHMAN JAVED 2 , CHINMAY CHAKRABORTY 3 , JAMEL NEBHEN 4 , WISHA ZEHRA1 , AND ZUNERA JALIL 2(2021): ElStream: An Ensemble Learning Approach for Concept Drift Detection in Dynamic Social Big Data Stream Learning, 2021

15- Indr˙e Žliobait˙e, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes (2020): Active Learning With Drifting Streaming Data

16- Heitor M. Gomes1 · Albert Bifet2 · Jesse Read2,3 · Jean Paul Barddal1 · Fabrício Enembreck1 · Bernhard Pfharinger4 · Geoff Holmes4 · Talel Abdessalem(2017): Adaptive random forests for evolving data stream classification

17- Sugandh Setha , Gurwinder Singha , Kuljit Kaur Chahala (2021) : Drift-based approach for evolving data stream classification in Intrusion detection system,

18- Subiksha Srinivasa Gopalan1 , Dharshini Ravikumar1 , Dino Linekar1 , Ali Raza1 , Maheen Hasib2: Balancing Approaches towards ML for IDS(2021): A Survey for the CSE-CIC IDS Dataset