Int. J. Nonlinear Anal. Appl. 14 (2023) 1, 299–307
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.27581.3654

Survey on intrusion detection system based on analysis concept drift: Status and future directions

Nora Sabah Salman*, Amer Abdulmajeed Abdulrahman

Computer Science Department, College of Science, University of Baghdad, Iraq

(Communicated by Mohammad Bagher Ghaemi)

Abstract

Nowadays, internet security is a critical concern; the One of the most difficult study issues in network security is "intrusion detection". Fight against external threats. Intrusion detection is a novel method of securing computers and data networks that are already in use. To boost the efficacy of intrusion detection systems, machine learning and deep learning are widely deployed. While work on intrusion detection systems is already underway, based on data mining and machine learning is effective, it requires to detect intrusions by training static batch classifiers regardless considering the time-varying features of a regular data stream. Real-world problems, on the other hand, rarely fit into models that have such constraints. Furthermore, various uses in the real world, Data distributions in intrusion detection systems, for example, are non-stationary, which produce concept drift over time or non-stationary learning. The word "concept drift" is used to describe the process of changing one's mind about something in an online-supervised learning scenario, the connection between the input data and the target variable changes over time. We define adaptive learning, classify existing concept drift strategies, evaluate the most typical, distinct, and widely used approaches and algorithms, describe adaptive algorithm assessment methodology, and show a collection of examples, all of this is based on the assumption that you have a basic understanding of supervised learning. The survey examines the various aspects of concept drift in a comprehensive manner in order to think about the current fragmented "state-of-the-art". As a result, which intends to give scholars, industry analysts, and practitioners a comprehensive introduction to idea drift adaptability.

Keywords: detection system, concept drift, intrusion detection, network security
2020 MSC: 68M25

1 Introduction

Because it contains such a big amount of data and information, the Internet has a variety of issues in terms of making it a secure system. Computer networks are vulnerable to a variety of threats. Security can be assured by installing firewalls and protecting software, however dynamic processes can be exploited. An intrusion detection system is one of the dynamic processes that establishes the specific objective of detecting attacks.an Intrusion Detection System (IDS) is a part of software that detects intrusions. That keeps an eye on the system for malicious activity and illegal access. Because of their simple accessibility, computer networks are prone to attack and a variety of threats from

*Corresponding author

Email addresses: nora.sabbah1201a@sc.uobaghdad.edu.iq (Nora Sabah Salman), amer.abdulrahman@sc.uobaghdad.edu.iq (Amer Abdulmajeed Abdulrahman)

Received: April 2022 *Accepted:* June 2022

attackers. An intrusion detection system analyzes a network of interconnected systems in order to prevent unusual intrusion or mayhem. The problem of intrusion detection is growing more difficult to solve as the number of computer networks grows. As computer systems become more connected, everyone has access to them, making it easier for hackers to hide their tracks and evade detection. Intrusion detection seeks to detect illegal access to, misuse of, and abuse of computer systems. Attackers' strategies and tools for hacking the network are always evolving. The intrusion detection system monitors and analyzes the network for any irregularities that could compromise computer security. Intrusion detection can be (1) Misuse and (2) Anomaly are the two methods Signature-based discovery, also known as pattern-based detection, is a method of detecting misuse. A significant benefit of these systems is that the patterns or signatures may easily develop and comprehend network activity. Anomaly detection methods rely on the creation of a normalized model of user behavior. The anomaly detection strategy for an intrusion is more successful in detecting novel threats. This is accomplished by examining network traffic using machine-learning approaches Concept drift in machine learning refers to a shift in the link between input and output data in a data stream. The term "concept drift" refers to the unspoken and concealed link between input and output variables. Any changes to the data could be made. (1) Gradual changes over time, (2) recurring or cyclical changes, and (3) abrupt or rapid changes are some of the other types of changes. Learning models must be capable of swiftly and accurately adjust to alterations. The concept drift detection approach is used to detect incoming new communications autonomously. The drift detector might be considered the most basic classifier. However, it's not like that straightforward as it is appears. On the one hand, Drift detection can be used to replace outdated models and save time, but it should not be allowed to generate too many false alarms. An algorithm that recognizes and returns information about the shifting patterns of an incoming signal is known as a concept drift detector. The model should usually be rebuilt as soon as feasible after returning the indication regarding the drift.

2 Network intrusion detection system data sets

In order to prevent security and network attacks, Intrusion detection systems for networks are designed to evaluate and monitor network traffic. The KDD Cup 1999 is a dataset of 41 features categorized into three groups: basic characteristics, traffic characteristics, and content features. It's most typically used to evaluate intrusion detection algorithms that need a certain collection of data to be audited, which might include a wide range of invasions.. In [28] Lee et al. used the KDD Cup 99 dataset to simulate a networked military situation. The amount of information in a KDD data set that is complicated was assessed by Tavallaee and colleagues. In [13] Mohammad Khubeb Siddiqui et al. sought to establish a link between hacker protocols and network assaults. The re-dundant and Datasets from the KDD Cup 99 have overlapping information. was suggested by Revathi and Malathi et al. [23]. The NSL-KDD dataset a suggestion by Tavallaee et al. to deal with concerns with the KDD'99 data sets. [7, 28] Dhanabal et al. In their study of NSL KDD, they assessed the utility of several classification algorithms in detecting anomalies in network traffic patterns and uncovered the link between protocols and network attacks. In the case of the NSL-KDD data set, Sarathi Partha et al. introduced Fuzzy In [24] the researchers used vectorized GA and weighted vectorized GA to detect network risks. [18] The UNSW-NB15 dataset is created by IXIA Perfect. NB15 dataset, which consists of many assault types. The data set utilized in NIDS is listed in Table 1.

Table 1: NIDS's Most Popular Datasets

NO	Dataset type	Attack type	Number of feature	labelled
1	KDD-CUP 99	Dos, probe, U2R, R2L	48	Y
2	NSL-KDD	Normal, DOS, probe, U2R, R2L	41	Y
3	KYOTO 2006+	multiple	24	N
4	ESCX 2012	Normal attack	9	Y
5	DARPA	DOS, probe, U2R, R2L	Multiple dataset	N
6	CICIDS2017	Web attacks, infiltration, botnets, brute force FTR, DOS	80	Y
7	CICIDS2018	Web attacks, infiltration, botnets, brute force FTR, DOS	80	Y
8	DEFCON	Port scan, buffer, flow attacks	None	N
9	UNSW-NB15	Fuzzess analysis, backdoors, DOS	49	Y
10	CICIDS2019	DAP, MSSQL, SNMP, SSDP, NTP, DNS, UDP	80	Y

3 Concept drift

Concept drift occurs when the statistical characteristics of a target domain change in an uncontrolled manner over time [16]. It was created by [15], who wanted to emphasize the fact that noise data can transform into non-noise data at any time ,These adjustments could be caused by changes in hidden variables that aren't directly detectable [24]. There are two types of concept drift:

1. Real concept drift is defined as changes in $p(y|X)$. Changes in p can happen with or without a change in p. (X)
2. Virtual drift happens when the entering data distribution shifts (i.e., $p(X)$ shifts) without changing $p(y|X)$. When the distribution of incoming data changes (i.e., $p(X)$ changes), virtual drift happens. Virtual drift is often referred to as temporary drift.

The primary goal of this survey is to deal with real idea drift that is not obvious in the input data distribution. In many circumstances, strategies for dealing with real It's also possible to employ notion drift to deal with drifts manifesting in the distributions of input data, but not the other way around. Techniques for tracking changing prior probability, techniques for dealing with virtual drift, and approaches for dealing with actual idea drift typically rely on input on expected performance, and techniques for dealing with novelty detection typically do not require such feedback.

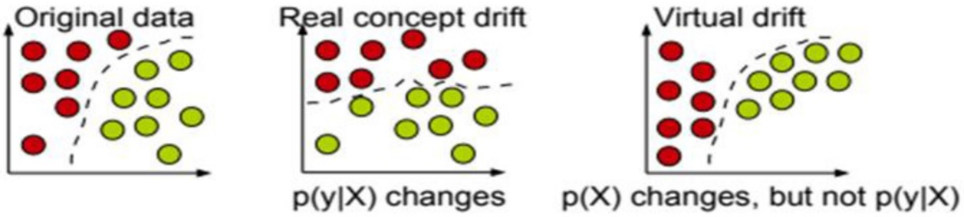

Figure 1: Types of drifts

3.1 Data changes when time passes

The distribution of data throughout time has changed can take many shapes, as seen Figure 2 shows a fictitious one-dimensional data set. Changes in the data mean occur in this data. A drift can occur quickly, gradually, incrementally, or repeatedly.

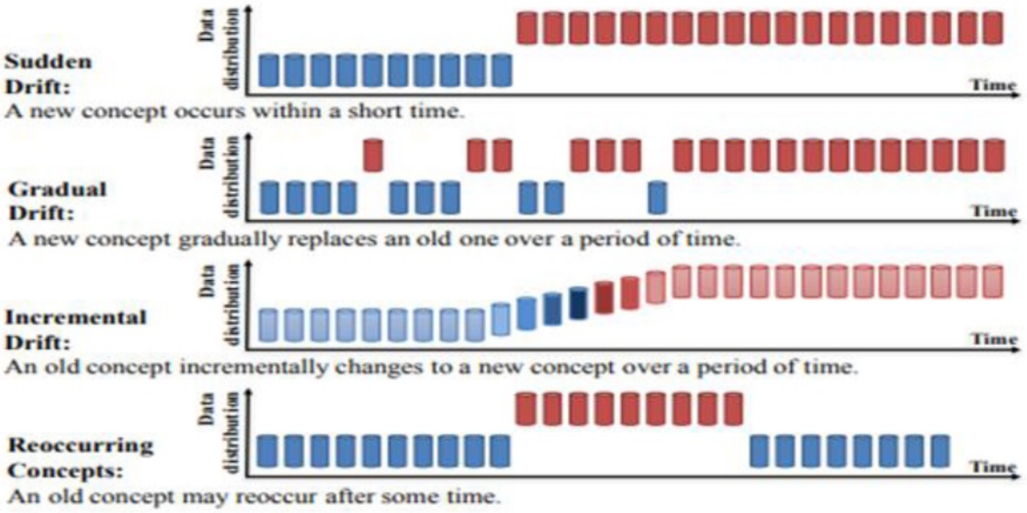
Sudden Drift:
A new concept occurs within a short time.
Gradual Drift:
A new concept gradually replaces an old one over a period of time.
Incremental Drift:
An old concept incrementally changes to a new concept over a period of time.
Reoccurring Concepts:
An old concept may reoccur after some time.
Data distribution
Time

Figure 2: Time-series patterns of change

3.2 Predictive model requirements in transforming contexts

In these cases, predictive models are applied must have procedures in place to identify and adjust to changing data in the course of time, otherwise their correctness will be jeopardized. The decision model may need to be changed over time to account for new information or completely replaced to reflect changing conditions. The following are requirements for predictive models:

1. Detect concept drift as soon as possible (and adjust if necessary)
2. Separate drift from noise and adapts to changes while being noise-resistant
3. Employ no more than a specified amount of memory for any storage and operate in less than the example arrival time.

4 Detecting concept drift

The methods and procedures for detecting and quantifying concept drift are known as drift detection. Alter the time intervals or the spots. a general summary in Figure 3 shows the four stages of the drift detection architecture.

The retrieval of data chunks is the major purpose of Stage 1 (Data Retrieval).

Stage 2 (Data Modeling) tries to abstract the collected data and extract the main features including sensitive information.

In Stage 3, the dissimilarity assessment, also known as distance estimation, takes place (Test Statistics Calculation). It determines the degree of the drift and gives data for hypothesis testing.

Stage 4 (Hypothesis Test) assesses the statistical significance of the change detected in Stage 3 using a unique hypothesis test.

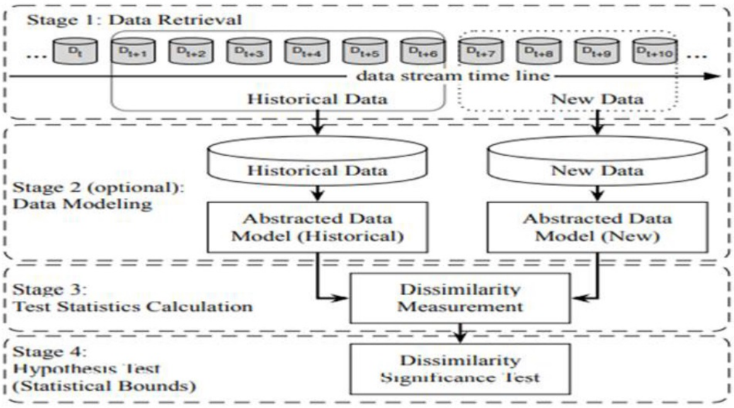
Stage 1: Data Retrieval
...
D_t
D_{t+1}
D_{t+2}
D_{t+3}
D_{t+4}
D_{t+5}
D_{t+6}
D_{t+7}
D_{t+8}
D_{t+9}
D_{t+10}
...
data stream time line
Historical Data
New Data
Stage 2 (optional): Data Modeling
Historical Data
New Data
Abstracted Data Model (Historical)
Abstracted Data Model (New)
Stage 3: Test Statistics Calculation
Dissimilarity Measurement
Stage 4: Hypothesis Test (Statistical Bounds)
Dissimilarity Significance Test

Figure 3: shows a general framework for detecting notion drift.

4.1 Algorithms for detecting concept drift

The techniques and methodologies for detecting drift are divided into three categories based on statistics from the tests used in this section.

4.1.1 Detection of drift based on error rates

The largest group of algorithms is PLearner rate of error approaches for detecting drift that are based. The goal of these algorithms is to keep track of base classifiers' online error rate has changed. If a statistically significant a rise or a fall in the error rate is discovered, an update process (drift alert) is going to start.

(The Drift Detection Method) (DDM) [30] is among the most well-known often used concept drift detection techniques. It was the first method to define the concept drift detection alert level and drift level A milestone time frame is used to implement Stage 1 in this technique, as seen in Figure 4. When a new data instance becomes available for analysis, determines whether the overall rate of online errors is high or low for the time range has grown sufficiently.

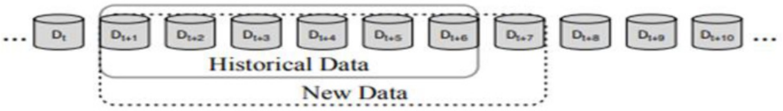

Figure 4: shows a time window for drift detection that is used as a benchmark. The window's beginning and end points are fixed, however Whenever a new data instance is received, The window is end point will be extended.

"Learning with Local Drift Detection" (LLDD) [9], "Early Drift Detection Method" (EDDM) [2], "Heoffding's inequality based Drift Detection Method" (HDDM) [14], "Fuzzy Windowing Drift Detection Method" (FW-DDM), and "Dynamic Extreme Learning Machine" (DELM) [29] all use similar algorithms.

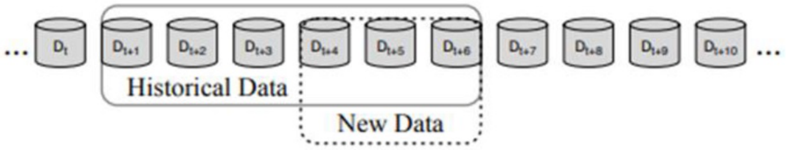

Figure 5: shows two periods of time in order to detect concept drift. The user must customize the New Data window.

In addition, (STEPD) [20] By comparing error rates, it is possible to detect a change in mistake rate the most current time window to the total time window, with In the system, In the system, there are two time windows for each timestamp Figure 6 shows how this is done. The new window's size should be determined by the user. Adaptive WINdowing (ADWIN) [3] is another common drift detection using two-time windows technique. Unlike STEPD, ADWIN requires users to declare the total size n of a "sufficiently large" window W rather than the size of the compared windows in advance. It then calculates concept nhist and nnew are sub-window sizes. based on the rate at which the two sub-windows, whist and wnew, change for all feasible cuts of W.

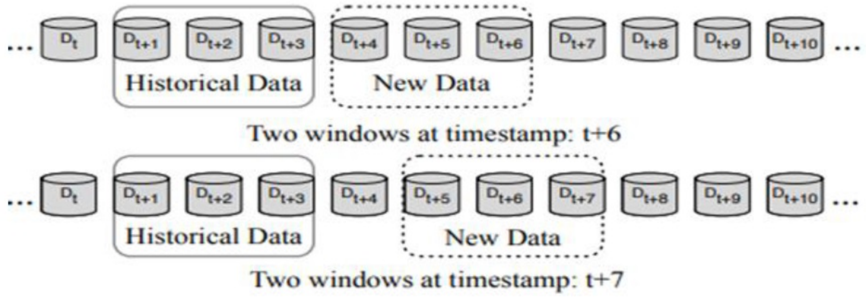

Figure 6: shows two fixed-size sliding time windows. The window for Historical Data will remain stationary; whilst the window for New Data will appear, continue to be mobile.

4.1.2 Data Distribution-based Drift Detection

The second most common form of drift detection technique is data distribution-based drift detection the difference between the distribution of historical data and the distribution of current data is significant quantified using a distance function/metric in this category of algorithms. According to the literature, [14] presented Relativeized Discrepancy is a group of distances. as the first systematic approach of change detection in data streams (RD). The Information-Theoretic Approach is another common Algorithm for detecting drift based on density (ITA). The basic idea behind this approach is to divide multidimensional historical and new data into bins using kdqTree. Multidimensional Data Statistical Change Detection (SCD) [27], Competence Model-based drift detection are two distribution-based drift detection methods/algorithms (CM), PCA-based change detection framework are two comparable distribution-based drift detection methods/algorithms (CM) (PCA-CD) [22], Equal Density Estimation (EDE) [11], Least Squares Density

Difference-based Change Detection Test (LSDD-CDT) [1], LSDD-INC (LSDD-INC) [5], and Local Drift Degree-based Density Synchronized Drift Adaptation (LDD-DSDA).

4.1.3 Detection of Drift in Multiple Hypothesis Testing

Multiple hypothesis test drift detection methods employ strategies similar to those described in the previous two categories. These are the algorithms are unique in that they employ several hypothesis tests to detect concept drift a variety of methods. There are two kinds of type of algorithms in this category: There are two types of multiple hypothesis testing:

1. Numerous hypothesis tests in parallel
2. Multiple hypothesis tests using a hierarchical structure.

The very first technique to set several drift detection hypotheses in this way was "Just-In-Time adaptive classifiers"(JIT). JIT's central idea is to increase the size of the CUSUM chart also known as "the Computational Intelligence-based CUSUM test" (CI-CUSUM), to determine the average change in the attributes that systems for learning are interested in.

Another method of detecting drift in a hierarchical structure has just been described. approach is Hierarchical Linear Four Rate (HLFR) [30]. There are two methods for detecting drift according to the request and confirm strategy are "Hierarchical Hypothesis Testing with Classification Uncertainty" (HHT-CU) and "Hierarchical Hypothesis Testing with Attribute-wise" "Goodness-of-Fit" (HHT-AG). The layer that detects in HHT-CU is a Heoffding's inequality-based hypotheses test that monitors the change in classification uncertainty measurement.

5 Adaptation to drift

The strategies are the emphasis of this section for drift adaptation or reaction, which is the process of updating current learning models based on the drift. Simple retraining, ensemble retraining, and model adjusting are the three primary kinds of drift adaptation approaches that try to remove many types of drift.

5.1 New global drift models are being developed

Retraining a new model based on the most current data to replace the old one an obsolete paradigm is possibly the most straightforward method of in response to concept drift. This method frequently uses a window strategy to save the majority of current re-training data and/or previous data about changes in distribution testing. This method is used in Paired Learners [1], which employs two learners: the steady learner and the learner who is reactive. A novel idea is discovered and the stable learner is substituted with the reactive learner if the stable learner often misclassifies cases that the learner who is reactive categorizes appropriately. This approach is straightforward to comprehend It's simple to set up and use, and it can be applied to any data stream. When deciding on an acceptable window size There must be a trade-off with a window-based method. A tiny window reflects the most recent data distribution better, whereas a big window allows more data to be collected for a new model's training. ADWIN is a common window design method that seeks to solve this difficulty. Unlike most previous it is effective and does not necessitate the user's participation. to predict a set window size in advance; rather, it analyses all possible window cuts and calculates ideal sub-window dimensions based on the rate at which the two sub-windows change. After determining the best window slashed, the old data window is eliminated, and the most recent window is used to train a new model. Rather of directly retraining the model.

5.2 An ensemble model is used for repeating drift

Preserving and reusing old models in the event of reoccurring concept drift might save time and effort when retraining a new model for repeating concepts. Using ensemble approaches to deal with the notion drift is predicated on this principle. In the last few years, the stream data mining community of researchers has given a lot of focus to ensemble approaches. Ensemble methods are made up of a group of base classifiers with varying types and parameters. Each base's output is to forecast freshly received data, a classifier is paired with voting rules. Many adaptive ensemble approaches in order to address concept drift, either by inventing specific adaptive voting rules or by enhancing standard ensemble approaches. To increase the performance of single classifiers, traditional ensemble approaches such as bagging, boosting, and Random Forests are used. All of them in order to improve to handle data in real time with a concept drift. An online form of the bagging method, which utilizes each instance just once, was first described to simulate batch mode bagging. This technique was used integrated Drift detection is made possible by the ADWIN algorithm system in a later work [4] to manage concept drift. When an idea drift has been reported, the newly proposed approach is used. Referred to as Leveraging Bagging [6] devised an adaptive boosting strategy in order to deal with concept drift by employing a hypothesis to evaluate check forecast accuracy. In a recent work, the Adaptive Random Forest (ARF) method was suggested, which combines the random forest tree technique with a concept drift detection approach, such as ADWIN, to determine when an old tree should be replaced with a new one. To deal with concept drift, many innovative ensemble approaches based on novel voting processes have been developed. In a similar study that leverages Hoeffding bound. "Dynamic Weighted Majority" (DWM) is a form of weighted majority that (DWM) to deal with notion drift, other voting procedures outside weighted voting have been used. The "Accuracy Update Ensemble" (AUE2) was developed with the goal of removing both sudden and gradual drift.

5.3 Making adjustments to existing models to account for regional drift

Developing a model that learns from changing conditions in a flexible manner input is an alternative to retraining a complete model. When the distribution of the underlying data changes, such models can partially update themselves. When drift occurs only in local areas, this strategy may be more efficient than retraining. Many solutions in this category are based on the decision tree algorithm, which has the ability to study and adapt to each sub-region separately. "The Very Fast Decision Tree classifier" (VFDT) is an online decision tree technique designed specifically for high-speed data streams. The Hoeffding bound is used to keep the number of instances necessary to separate the nodes to a minimum. CVFDT [12], an expanded version, was later developed to remove concept drift. A window that slides is kept in CVFDT to keep the most recent data. VFDTc [10] is a new effort to better VFDT with various innovations, including the capacity to remove numerical characteristics, in tree leaves naive Bayes classifiers are used, and the detection and adaptation of idea drift. VFDTc was further enhanced by majority vote, Naive Bayes, and Weighted Naive Bayes are three alternatives for selecting the best classifier in an adaptive leaf methodology. Despite VFDT's success, another decision tree method (IADEM-3) [8] attempts to remedy the same problem by using Hoeffding bounds.

6 Concept drift detection in network intrusion detection system

Because of its potential to evaluate Variance in data distribution across time, concept drift is important. Furthermore, IDS can be viewed as a classic case of concept drift the data beneath the scanner is normally in good shape, and it is dispersed evenly for a single source supplying a network with a data stream. Furthermore, in the event of an unknown breach, the present data distribution changes dynamically in comparison to past data. This drives the development of an adaptive intrusion detection approach based on incremental learning and Concept Drift that adjusts fast to novel incursion types. Furthermore, the constant introduction of new assaults and security flaws necessitates the development of an ideal classifier that can swiftly adapt to new infiltration tactics. The static batch learning technique outlined above performs poorly in this case. In other words, when a static classifier becomes outmoded, it becomes slow to respond to new incursion types, necessitating a re-training [13] that requires investments at a great cost. In contrast, with an adaptive classifier, incremental learning is possible to updates to a steady stream of data across time ensures an IDS's performance that looks promising.

Table 2: Some concept drift algorithms using in intrusion detection system

Researcher name	**Algorithms used**	**Dataset**	**Accuracy**
Sugandh Seth and et al 2021 [26]	Adaptive Random Forest classifier with ADWIN	CIC IDS 2018	99.5
Ida Seraphim and et al 2021 [25]	The Nave Bayes classifier, the Hofding tree classifier, and the Ensemble classifier	NSL-KDD	35.72 88.7 89.23
Preeti Mishra et al 2018 [17]	Decision Tree classifier,(ANN) Naïve Bayes Classifier, support vector machine (SVM) Genetic Algorithm.	NSL-KDD	99.6
Asmah Muallem et al 2017 [19]	Hoeffding tree Restricted Hoeffding tree Accuracy Updated Ensemble	KDD Cup'99 NSL-KDD	99 93
Pradheep D et al 2020 [21]	HDDM and Hoeffding tree algorithm	NSL-KDD 99	0.79% 0.88%

7 Conclusion

In this study, we discussed the notion of drift in general and how it might arise in situations where it is impossible to exist in stationary data, such as nonstationary (streaming data), where a change in data properties occurs, resulting in concept drift. Then we went through the concept of data drift and the different sorts of drift, such as actual and virtual drift, as well as the way of data changes over time and the different types of changes that can occur. The forms of intrusion that can occur, which are separated into two types: MISUSE and ANOMALLY, and how to recognize and deal with them, as well as the types of data sets utilized in intrusion detection systems, their features, and the types of attacks in these groups, were all covered. Then we discussed Methods for detecting drift and algorithms that are divided into three groups based on Millennium's test results. On the basis of error rate-based drift detection, one was divided. The initial one is category is founded on Drift Detection Algorithms, the second is founded on Data Distribution-based Algorithms. Detection of Drift Its techniques were discussed, and the third category is based on Multiple Hypothesis Test Drift Detection and its algorithms. Then, drift adaptation and the techniques used to address different types of drift were examined, which were separated into three groups: training new models for global drift, model ensemble for recurring drift, and updating existing models for regional drift. It was also discussed how to cope with concept drift NIDS, as well as a number of studies in this subject.

References

[1] S.H. Bach and M. Maloof, *Paired learners for concept drift*, Proc. 8th Int. Conf. Data Mining, 2008, pp. 23–32.

[2] M. Baena-Garcıa, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavalda and R. Morales-Bueno, *Early drift detection method*, Proc. 4th Int. Workshop Knowledge Discov. from Data Streams **6** (2006), 77–86.

[3] A. Bifet and R. Gavalda, *Learning from time-changing data with adaptive windowing*, Proc. 2007 SIAM Int. Conf. Data Mining, Soc. Ind. Appl. Math., 2007, pp. 443–448.

[4] A. Bifet, G. Holmes and B. Pfahringer, *Leveraging bagging for evolving data streams*, Proc. Joint Eur. Conf. Machine Learn. Knowledge Discovery in Databases, Springer, 2010, pp. 135–150.

[5] L. Bu, D. Zhao and C. Alippi, *An incremental change detection test based on density difference estimation*, IEEE Trans. Syst. Man, Cybernet. Syst. **47** (2017), no. 10, 2714–2726.

[6] F. Chu and C. Zaniolo, *Fast and light boosting for adaptive mining of data streams*, Proc. 8th Pacific-Asia Conf. Knowledge Discovery Data Mining, H. Dai, R. Srikant and C. Zhang (Eds.), Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 282–292.

[7] L. Dhanabal and S.P. Shantharajah, *A study on NSL-KDD dataset for intrusion detection system based on classification algorithms*, Int. J. Adv. Res. Comput. Commun. Eng. **4** (2015), no. 6, 446–452.

[8] I. Frias-Blanco, J. del Campo-Avila, G. Ramos-Jimenez, A.C. Carvalho, A. Ortiz-Diaz and R. Morales-Bueno, *Online adaptive decision trees based on concentration inequalities*, Knowledge-Based Syst. **104** (2016), 179–194.

[9] J. Gama and G. Castillo, *Learning with local drift detection*, Proc. 2nd Int. Conf. Adv. Data Min. Appl., Springer, 2006, pp. 42–55.

[10] J. Gama, R. Rocha and P. Medas, *Accurate decision trees for mining high-speed data streams*, Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, ACM, 2003, pp. 523–528.

[11] F. Gu, G. Zhang, J. Lu and C.-T. Lin, *Concept drift detection based on equal density estimation*, Proc. 2016 Int. Joint Conf. Neural Networks. IEEE, 2016, pp. 24–30.

[12] G. Hulten, L. Spencer and P. Domingos, *Mining time-changing data streams*, Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, San Francisco, California, ACM, 2001, pp. 97–106.

[13] A. Kashyap and A. Nayak, *Different machine learning models to predict dropouts in MOOCs*, IEEE Int. Conf. Adv. Comput. Commun. Inf. (ICACCI), 2018, pp. 80–85.

[14] D. Kifer, S. Ben-David and J. Gehrke, *Detecting change in data streams*, Proc. 30th Int. Conf. Very Large Databases, VLDB Endowment **30** (2004), 180–191.

[15] A. Liu, Y. Song, G. Zhang and J. Lu, *Regional concept drift detection and density synchronized drift adaptation*, Proc. 26th Int. Joint Conf. Artif. Intell., 2017.

[16] N. Lu, G. Zhang and J. Lu, *Concept drift detection via competence models*, Artif. Intell. **209** (2014), 11–28.

[17] P. Mishra, V. Varadharajan, U. Tupakula and E.S. Pilli, *A detailed investigation and analysis of using machine learning techniques for intrusion detection*, IEEE Commun. Surveys Tutorials **21** (2018), no. 1, 686–728.

[18] N. Moustafa and J. Slay, *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*, Military Commun. Inf. Syst. Conf. MilCIS 2015 - Proc., 2015, pp.1–6.

[19] A. Muallem, S. Shetty, J.W. Pan, J. Zhao and B. Biswal, *Hoeffding tree algorithms for anomaly detection in streaming datasets: a survey*, J. Inf. Secur. **8** (2017), no. 4.

[20] K. Nishida and K. Yamauchi, *Detecting concept drift using statistical testing*, Proc. 10th Int. Conf. Discovery Science, V. Corruble, M. Takeda and E. Suzuki (Eds.), Berlin, Heidelberg: Springer, 2007, pp. –269.

[21] D. Pradheep, R. Gokul, V. Naveen and J. Vijayarani, *Anomaly intrusion detection based on concept drift*, Global J. Comput. Sci. Technol. 20 (2020), no. 2, 14–22.

[22] A.A. Qahtan, B. Alharbi, S. Wang and X. Zhang, *A PCA-based change detection framework for multidimensional data streams*, Proc. 21th Int. Conf. Knowledge Discovery Data Mining, ACM, 2015, pp. 935–944.

[23] D.A.M.S. Revathi, *A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection*, Int. J. Eng. Res. Technol. **2** (2013), no. 12, 1848–1853.

[24] J.C. Schlimmer and R.H. Granger Jr, *Incremental learning from noisy data*, Machine Learn. **1** (1986), no. 3, 317–354.

[25] B.I. Seraphim and E. Poovammal, *Analysis on intrusion detection system using machine learning techniques*, Computer Networks, Big Data and IoT, Springer, Singapore, 2021, pp. 423–441.

[26] S. Setha, G. Singha and K.K. Chahala, *Drift-based approach for evolving data stream classification in intrusion detection system*, Workshop Comput. Networks Commun., 2021, pp. 23–30.

[27] X. Song, M. Wu, C. Jermaine and S. Ranka, *Statistical change detection for multi-dimensional data*, Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining, San Jose, California, USA, ACM, 2007, pp. 667–676.

[28] M. Tavallaee, E. Bagheri, W. Lu and A.A. Ghorbani, *A detailed analysis of the KDD CUP 99 data set*, IEEE Symp. Comput. Intell. Secur. Defense Appl. CISDA, 2009, pp. 1–6.

[29] S. Xu and J. Wang, *Dynamic extreme learning machine for data stream classification*, Neurocomput. **238** (2017), 433–449.

[30] S. Yu and Z. Abraham, *Concept drift detection with hierarchical hypothesis testing*, Proc. 2017 SIAM Int. Conf. Data Mining. SIAM, 2017, pp. 768–776.