Mehran University Research Journal of Engineering and Technology

https://doi.org/10.22581/muet1982.3269

2024, 43(3) 205-218

A clustering-based method for outlier detection under concept drift

Mahjabeen Tahir [a, b *], Azizol Abdullah [a], Nur Izura Udzir [a], Khairul Azhar Kasmiran [a]

[a] *Department of Computer Science and Information Technology, University Putra Malaysia (UPM), Serdang,43400, Selangor, Malaysia*

[b] *Department of Computer Engineering, Sir Syed University of Engineering and Technology, Karachi, 75300, Pakistan*

[*] Corresponding Author: Mahjabeen Tahir, Email: engr.mahjabeen.tahir@gmail.com

Received: 14 May 2024, Accepted: 08 June 2024, Published: 01 July 2024

K E Y W O R D S

Outlier Detection

Clustering

Network Security

Streaming Data

Unsupervised Algorithms

Concept Drift

A B S T R A C T

The ongoing challenge against network security issues persists, necessitating the exploration of alternative approaches. Anomaly-based strategies, diverging from traditional signature-based methods, gain popularity for their effectiveness in detecting new attacks. However, accurately defining normal network behavior becomes increasingly challenging due to data fluctuations. This study introduces a two-step process for recognizing evolving anomalies in streaming network data. Initially, clusters are updated incrementally upon new data arrival (the updating phase). Subsequently, anomalies are identified by discerning outer and inner outliers using minimum and maximum density thresholds. A buffer concept temporarily stores incoming data to prevent misclassification of normal network samples as anomalies. Performance evaluation in Python 3 assesses the impact on detection rate, false positives, and accuracy using two popular streaming datasets (NSL-KDD and UNSWNB-15). The algorithm achieves notable results, with a detection rate of 99.12% on UNSWNB-15 and a 7.9% false positive rate on NSL-KDD, marking significant progress. The proposed approach CADSD (Cluster-based Anomaly Detection with Streaming Data), operates in real-time without pre-training. However, challenges may arise from assuming the majority of data comprises normal instances, particularly during sudden spikes in attack data, potentially diminishing algorithm effectiveness. Nonetheless, the method shows the potential to enhance network security by promptly identifying emerging anomalies in real-time streaming data. The incorporation of a buffer concept to prevent the misidentification of normal network samples as anomalies underscores the innovative nature of this approach.

1. List of Abbreviations

The following abbreviations are used in this manuscript.

MC	Micro-cluster
mD	Minimum Density
MD	Maximum Density
Pmc	Prospective microcluster
RadiusM	Radius Max
RadiusMN	Radius Min
Sb	One-Stream block length
MCcore(Imc)	Core microcluster
MCoutlier(Omc)	Outlier microcluster
B	Unique Base-cluster id
I	microclusters
EL’	Edge list microcluster

2. Introduction

This section introduces the motivation of the work, followed by the literature review and our

contributions. Finally, the organization of the manuscript is presented.

2.1 Motivation

According to a report by McAfee, cybercrime has caused global economic losses exceeding $1 trillion, which is equivalent to the global GDP of 1% [1]. The number of network attacks incrementing every year and taking on a variety of forms [2]. Many intrusion detection systems currently in use rely on signature-based techniques that can only identify known attacks by matching traffic patterns to previously defined attack signatures [3]. This means that any traffic that matches the signature is automatically flagged as an attack. However, anomaly-based intrusion detection techniques are capable of detecting unknown attacks by identifying deviations from normal network traffic patterns. Therefore, it is essential to create an accurate model of normal network traffic to ensure the effectiveness of intrusion-based detection.

The manual labelling of data demands substantial time and effort, exceeding the capacity of human experts. In contrast, unsupervised network anomaly detectors possess the ability to extract valuable insights from data without depending on pre-existing knowledge or labelled datasets.

For changing data patterns Online clustering emerges as an unsupervised method making it well-suited for immediate anomaly detection [4].

The underlying concept of these algorithms is rooted in the assumption [5] that intrusive activities constitute a minor fraction of the overall network traffic and manifest distinct patterns compared to normal traffic.

The CADSD algorithm enhances the earlier online clustering method BOCEDS [6]. The CADSD does not require any training of dataset or prior knowledge Unlike BOCEDS [6]. With the arrival of new data, the system continuously updates the normal patterns and promptly detects and removes anomalies. This reduces the computational cost and makes it suitable for real-time intrusion detection.

2.2 Related Work

Online clustering is a technique that is used to analyze evolving data streams, particularly in real time for detecting anomalies. Network traffic, which is commonly represented as data streams, can benefit from the application of online clustering algorithms to identify anomalies. Various online clustering methods have been proposed over the past two decades, such as CluStream, DenStream, C-Denstream, and SDStream [7,8]. More recently, CEDAS [9], an improved version of CODAS, has been proposed as a fully online algorithm. CEDAS employs micro-clusters to characterize data samples, and upon the arrival of new data, it either updates existing micro-clusters or generates new ones. This methodology utilizes 'energy' to handle temporal information within the micro-clusters, enabling the identification of evolving properties in the data stream and the formation of high-quality clusters. BOCEDS represents an enhanced iteration of CEDAS.Numerous recent approaches for detecting network anomalies rely on online clustering.

In 2017, Dromard et al. [5] introduced ORUNADA, a method for network anomaly detection based on clustering using a sliding window. This method detects anomalies in data streams by incrementally updating a grid with a normal pattern and identifying outliers using outlier scores. The detection accuracy was satisfactory, but the time required for processing could be improved. In 2018, researchers proposed a method for detecting network anomalies in data streams using a Cauchy density-based network [10]. This approach creates a new cluster if the distance from existing clusters surpasses a specified threshold.

In 2018, Bigdeli et al. [11] introduced an incremental anomaly detection approach that utilizes Gaussian mixture models (GMMs) for cluster representation. This two-layer structured model updated the Gaussian mixture models (GMMs) with new upcoming data samples and successfully filtered out noise.

During the creation of a new cluster this method required clusters to be labeled as either regular or assault. In 2019,[3] introduced a model for anomaly detection that relied on clustering of data that arrives continuously. The clusters were continuously refreshed as new data samples arrived, making efficient use of available memory resources. Outliers were identified as the distance from the current clusters increases exceeded a pre-defined threshold. To detect network anomalies based on behavior analysis Chonghua et.al., 2022 [12] Unlike other state-of-the-art algorithms that only consider the difference between normal and abnormal, CCAD tracks the problem of collective anomaly scattered among multiple clusters when applying clustering-based algorithms in stream network traffic. [13], Meryem et al. in 2023 demonstrated the utilization of autoencoder-based models in the Fed-ANIDS algorithm, which has been proven to outperform other generative adversarial network-based models regarding detection precision and reduced false positives. An overview of outlier detection techniques from 2022 to 2023 is presented in Table 1.

Table 1

Techniques of network anomaly detection published between 2022 and 2023

Authors & year	Method	Supervised/ Unsupervised	Offline / Online	Dataset	Performance Metrics
Shajoun & Xin Wang 2023 [20]	Multitask learning	Super vised	online	UNSW-NB15	Accuracy, Recall, F1 score, training time
Xueyuan et.al., 2023 [21]	MSRC	Super vised	offline	KDD99, NSLKDD, UNSW-NB15, CICIDS 2018	Accuracy, Recall, F1 score.
Igor et.al., 2023 [22]	N/A	Super vised	offline	UNSW-NB15	Accuracy, F2 score, AUC
Maya et.al., 2023 [23]	GBDT	Super vised	offline	NSL-KDD, UNSW-NB15, and HIKARI-2021	Accuracy, Precision, Recall, F1, MCC
Chonghua et.al., 2022 [49]	CCAD	Unsuper vised	online	UNSW-NB15, CICIDS2017	ROC, AUC, Runtime
Meryem et.al., 2023 [50]	Fed-ANIDS	Unsuper vised	online	USTC-TFC2016, CIC-IDS2017, CSE-CIC-IDS2018	Accuracy, F1 score, FDR
Aiguo et.al., 2022 [41]	NBAD	Unsupervised	offline	KDD99, CICIDS 2017	Accuracy ROC, Recall, Training and testing time

2.3 Contributions

The CADSD (Clustering-Based Anomaly Detection for Network Streaming Data) method suggests the anomaly detection phase involves partitioning the analyzed data stream into two blocks, the latest data represented with block two. Consequently, the outlier micro-clusters Omc in this subsequent block might not be able to classify a significant number of data samples due to less time to transition in Imc which are core clusters. To avoid misclassification, a shield is employed to save the outlier clusters.

The buffer plays a crucial role in storing outlier micro-clusters Omc and contributes to the precise differentiation between regular and outlier micro-clusters Omc during the clustering process in CADSD. In the suggested framework, the outlier clusters kept in the storage are called prospective micro-clusters Pmc. With a new stream of data coming in and the process progressing, some prospective clusters may develop into core clusters Imc and ultimately transition into regular clusters. This approach differs from BOCEDS as the buffer stores outdated micro-clusters temporarily. The methods differ in their strategies to manage and adapt to changing data streams.

The CADSD structure is designed to save concise information about micro-clusters. which can be used to provide a summary of network traffic samples. This enables the detection of low-density samples in local areas through micro-clustering, and low-density clusters in global areas through base-clustering. Normal patterns, referred to as normal base clusters, are updated incrementally over time, allowing for real-time detection of outliers. In general, the contributions of CADSD can be outlined as follows:

- A novel method has been created to identify anomalies in real-time data streams. This approach consists of clustering data streams through a two-phase process. In the initial stage, clusters are dynamically updated with incoming data. Next, in the outlier finding stage potential clusters Pmc are stored in a storage and any anomalies are identified and removed during the outlier's removal stage.
- Anomalies are redefined by setting thresholds for minimum density (LD) and maximum density (MD). This redefinition encompasses not only outlier clusters Omc having a density under the minimum limit as well as base clusters possessing a particular density level lower than the maximum requirement of density.

- A new algorithm with a buffer is proposed to prevent prospective clusters Pmc from being erroneously identified as genuine anomalies during the anomaly detection phase leading to a significant enhancement in detection performance. The most recent dataset, received after the last anomaly detection phase, is split into two distinct streams. The second stream, comprising the latest data, retains any outliers identified as prospective micro-clusters.
- The recently introduced unsupervised anomaly detection algorithm surpasses current solutions in various aspects, including detection capability, false positive rate, accuracy, and computational complexity. Notably, it excels in the real-time detection of network anomalies.

2.4 Organization

The article follows a structured format outlined as follows: Section 2 provides an in-depth examination of the proposed CADSD algorithm, accompanied by its pseudo-code. Moving forward to Section 3, we showcase the results obtained from experiments conducted on two extensively employed network streaming datasets, substantiating the effectiveness of CADSD. Additionally, we conduct a comparative analysis of CADSD with other algorithms using evaluation metrics. Section 4 provides the limitations of the proposed algorithm, and finally, Section 5 serves as the conclusion, summarizing the study and outlining potential future research directions.

3. Methodology

3.1 CADSD: Proposed Algorithm

The CADSD algorithm aims to establish a standard pattern that can be continually updated as characteristics evolve. Additionally, the algorithm must efficiently detect and eliminate abnormal data points with high precision. CADSD specifically retains concise information about micro-clusters. One can depict the connection between micro-clusters and the base-cluster formed by the intersection of the micro-clusters. The arrangement is defined, with its center represented by the central point of a microcluster. The cluster's radius is a direct line extending from the middle of the circle to the outer border of the circle. The kernel region, encompassing half of the radius, is found within the inner circle, while the shell region is the space between the inner and outer circles.

Fig. 1 [14] depicts the correlation between micro-clusters and the corresponding base-cluster they represent. Two clusters intersect when the central region of one overlaps including at least the outer layer of another micro-cluster. In essence, two micro-clusters intersect when the distance between their center's radii is less than sum of the half of their radius. The resultant overlapping micro-clusters, labeled as I1, I2, and $I3, collectively form the base-cluster B1. Significantly, despite the connection between I4 and I5, they do not consider Overlapping due to the outer boundary of I4 only intersecting with the outer boundary shell I5. Together, the micro-clusters I4 and I5 constitute another base-cluster, B2 [14].

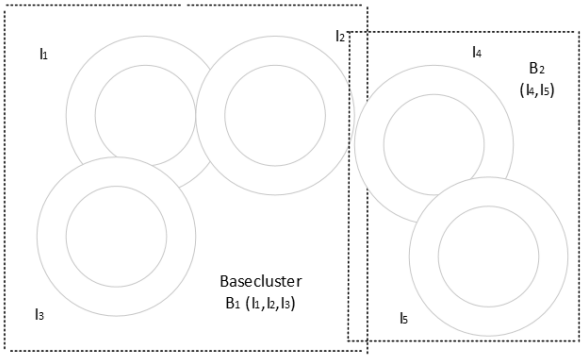

Fig. 1. Intersection of Micro-Clusters and Base-Clusters

A MC which is a microcluster is characterized by a set of values (mD; C; R; EL'; B). In this representation, 'mD' denotes Min-density, which corresponds to the count of samples within the micro-cluster. 'C' represents the center of the micro-cluster, calculated as the mean of the samples contained within it. 'R' represents the micro-cluster's radius. Edge list EL 'of the micro-cluster, specifying intersecting micro-clusters. 'B' serves as the base-cluster ID, a unique integer that identifies the specific base-cluster where the micro-clusters are situated. Additionally, if a micro-cluster has a min-density above a certain threshold and no intersections with other micro-clusters is also recognized as a base-cluster.

3.2 CADSD Algorithm and Its Procedure

This study is based on the concept that incoming malicious packet patterns will be different from the normal network traffic [5]. The following terminologies are associated with the algorithm.

- Cluster Head: A cluster head depicts a group of small clusters and shows how they combine to form larger clusters.
- Inner micro-cluster Imc: The micro-clusters contain several samples that meet or exceed the density threshold.
- Outerlier micro-cluster Omc: The count of samples within the microcluster falls below the density threshold.
- BaseCluster (B): Intersection of Several microclusters created a one base cluster.

- Min-density (mD): The quantity of samples within a single Inner micro cluster.
- Max-density (MD): The quantity of samples within a single Outer micro cluster.
- Prospective microclusters Pmc: When the sample count falls below the density threshold, it becomes an Outer micro-cluster. Any possible Pmc is kept in a buffer.
- RadiusMax (RadiusM) and RadiusMin (RadiusMN): Expert knowledge of the application is used to determine the highest and lowest radius of the microclusters. The maximum radius is essential to ensure the uniformity and separation of the micro-clusters, while the minimum value of radius is for generating clusters comprising a satisfactory quantity of data points.
- Min-Threshold: The minimum density required for a sample to constitute a core microcluster.
- Outlier-Threshold: The maximum density of an outer-micro-cluster required for the formation of a regular cluster. In this scenario, the Outlier-Threshold exceeds the Min-Threshold.
- Regular clusters: Regular clusters are stored in memory when the quantity of data points within one cluster surpasses the Outlier-Threshold.
- Outlier clusters: Outliers are recognized by considering both the min-density threshold (Min-Threshold) for micro-clusters with lower density and the max-density threshold (Outlier-Threshold) for the base cluster with lower density.
- Stream-block: Duration of one-datastream Sb.

The execution of the proposed CADSD algorithm started once the system parameters (Min-Threshold, Outlier-Threshold, RadiusMax, RadiusMin, and Sb were configured. Subsequently, the program awaits the arrival of data samples. The algorithm is implemented right from the start and doesn't require any preprocessing.

Upon the arrival of new data traffic, the algorithm initiates a search to identify the specific micro-cluster that the sample is associated with. The search involves both the micro-clusters stored in the buffer and in memory. The details linked to the intended cluster are revised once the search is accomplished. Conversely, a new microcluster is created if the search is unsuccessful. Base-cluster modification triggers an update in the base-cluster. The specified step is referred to as the increment stage. In this stage, new data continuously arrive and undergo processing, and the base clusters are dynamically generated online in the suggested CADSD algorithm. These base-clusters undergo continuous updates, adapting in sync with the incremental adjustments to the micro-clusters.

Anomaly detection is triggered following the consecutive increment process of two streaming data blocks, marked by an exceptionally brief period.

During this process, only the regular clusters, depicted as base-clusters, persist in memory. The allocated potential clusters, demonstrating regular arrangement, undergo incremental updates to ensure they adapt to the evolving and changing data stream. To enhance detection accuracy buffering is employed to prevent potential micro-clusters from mistakenly removed as inner anomalies.

Furthermore, a modified definition of an outlier is introduced. The term now encompasses not only inner outliers as in BOCEDS but also outer outliers, referring to base-clusters having limited data points. The expanded definition significantly adds to the overall enhancement of detection precision.

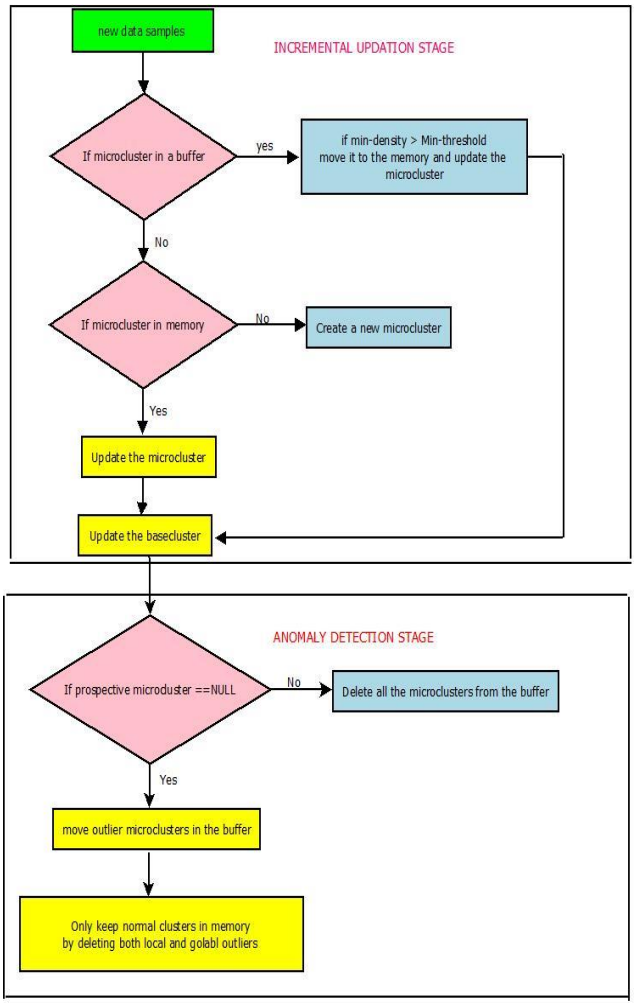

Fig. 2. Model of The Proposed CADSD Algorithm

In each interval, our dataset assumption constitutes most of the normal dataset. If there is a substantial influx of invasive data points within a specific time period, the maximum density formed by these intruding data points of the clusters might surpass the

highest density threshold. Consequently, these clusters could be inaccurately classified as regular, signalling a potential distortion in the evolving regular patterns. This scenario is acknowledged as a drawback for the proposed CADSD, as it has the potential to compromise the accuracy of the system when confronted with such abnormal data patterns. Fig. 2 offers a visual depiction of the structure of the algorithm.

3.2.1 Increment stage

Streaming data samples are executed by scanning them in a single run. Upon the arrival of a new sample, the program searches to find the corresponding micro-cluster it belongs to. Initially, it looks for the target micro-cluster within the prospective clusters within the storage and subsequently among the clusters stored in the computer's memory. When the search is accomplished, the specific cluster undergoes an update; alternatively, another micro-cluster is created. This process leads to updates in the base-clusters, represented as clusters, whenever a new Inner-microcluster is created or an existing Inner-microcluster is modified. The algorithm for the increment stage unfolds in two steps: (i) modifying the microclusters and (ii) adjusting the clusterhead.

i) Microcluster Updation

When a new sample is obtained, the algorithm finds the corresponding micro-cluster it is associated with. If a micro-cluster acquires a new data point, the algorithm undergoes recursive updates. The algorithm involves two main processes: "searching" and "updation". The algorithm first examines the prospective micro-clusters in the storage and then in the memory bank to identify the target microcluster. If the distance from the data point to the micro-cluster's center is shorter than the micro-cluster's radius, then the data point is deemed a member of that micro-cluster [14].

The min-density is increased by one if the datapoint is within the aimed micro-cluster and when the datapoint falls within the shell region the specific micro-cluster's radius and center are updated. No updation is required if the datapoint falls within the kernel region. A new micro-cluster is formed in the event of an unsuccessful search for the target micro-cluster, and its characteristics are set initially with the min-density set to 1, the radius established to RadiusMN, the central point set to the datapoint, and with the emptied edge list. The variable M is configured to 0 since it does not pertain to any base group.

Algorithm 1 starts the process. Upon receiving a new datapoint si', the micro-cluster is first sought in the buffer, specifically in the Prospective microcluster Pmc. If still the intended micro-cluster is not found in Pmc, the algorithm then proceeds to search for it in memory, within the set of micro-clusters (MC). A new microcluster is generated if the aimed micro-cluster is still not found in microclusters. The process involves the following steps of finding the micro-cluster in Pmc.

The distance 'd' calculated between Si' and the microcluster closest center L'(mD, C, EL', R, B), where, L'Ɛ (Pmc). One of the following steps will be executed for the distance calculation between 'd' and 'R':

- If distance < Radius, si' remain in the region of L'. Update the microcluster C'. Assign the min-density to mD+1.
- If R/2 < d<R (si' remains in the region of L').

Update C by Eq. (1) and R by Eq. (2) where 'n' is the number of data samples within the stream of data.

$$C_{new} = \frac{(mD * C + L')}{(mD + 1)} \quad (1)$$

$$R_{new} = \min\left(R\frac{\frac{2\,d}{R} - 1}{2n}\right), \text{Max}\, Radius \quad (2)$$

- If d ≥ R

The expected microcluster is not in the buffer it is searched in the MCcore and MCoutlier.

The procedure for identifying the target micro-cluster in both MCcore and MCoutlier is similar to that in Pmc. A new micro-cluster is created if the target micro-cluster is not found, with the smallest radius and sample as its center. First initiate to 1 the edge list, min-density, and base-cluster ID.

Algorithm 1: Initializing Stage

Input: si' sample datastream, MC (micro-clusters), P_{mc} (prospective micro-clusters), RadiusMinimum, RadiusMaximum.

Stage 1: closest micro-cluster L'(L' ϵ P_{mc})

If the distance d is < between si' and L' then

radius of L'

goto stage 2

else

goto stage3

[End]

Stage 2: Modify the mD of L', mDnew – mD + 1

If (Radius/2 < (datasample, center of L') < Radius) [C is the center of L']

Using eq1 and eq2 to update the center and the radius.

[end]

Stage 3: search the nearest microcluster L', (L' ϵ MC)

Suppose distance d among si' and C' < the radius of C'

jump to step 4

else

jump to step 5

[End]

Stage 4: renew the mD of L', mDnew = mD + 1

if (Radius/2 < (si', C) < Radius)

renew L' using Eq. 1 and Eq. 2

[End]

Stage 5: generate a new microcluster L' (mD, C, R,

EL', B)

mD = 1, C = L', R = RadiusMinimum, EL' = {}, B =0

ii) ClusterHead Updation

clusters are referred to as baseclusters, and intersecting core micro-clusters act as cluster heads, collectively forming baseclusters. The clusterhead structure updates the information if any new microcluster is formed or any details about core micro-clusters change, we call this updated or new microcluster 'C'. Similarly, if any changes in the core micro-cluster, or new creation of the microcluster or its center are changed, then the cluster head structure and the baseclusters are updated accordingly. The CASDSD algorithm enables online updates to the baseclusters by modifying the clusterhead.

To update the edge list or for the modification of the edge list of these core microclusters as described in algorithm 2, identify all core micro-clusters intersecting with C. Eq. (3) [14] is employed to calculate the distance of intersection, denoted by 'd'. The count of baseclusters will be updated to a new value for the emerging cluster of the edge list.

distance' = Radius + Radius'/2 if Radius ≥Radius'

Radius' + Radius'/2 if Radius'>Radius (3)

Algorithm 2: Update the Cluster

Begin: core microcluster, Imc

While new core micro-cluster is formed or the center of an existing core micro-cluster C is altered

do

Stage 1: for every L' which is the other microcluster(core) in Imc, then,

Calculate the distance between L and L'

Calculate the intersecting distance denoted as d' between L and L' in Imc using eq (3)

If d ≤ d', then sum the L' and L in edge list of L and L'

[end]

Stage 2: If any changes are made to the edge list of a micro-cluster, adjust the count of base clusters throughout the entire cluster.

[end]

3.2.2 Anomaly detection stage

Cumulative processing of two datastreams occurs when the next phase of anomaly detection is triggered, Si' (Si'1, Si'1+1, ..., Si'1+n) and Si'+1(Si'1+n+1, Si'1+n+2, ..., Si'1+2n).

During this cumulative processing phase, only regular clusters are left in memory while potential microclusters are transferred to the storage. The reserved regular clusters symbolize recurring patterns that undergo ongoing updates to adapt to evolving and changing data patterns. The use of buffering techniques prevents potential micro-clusters from being mistakenly labeled as inner outliers, thereby considerably enhancing the precision of the detection accuracy. Furthermore, a modified way to call an outlier is presented, where the term "outlier" now includes not just inner anomalies as called in BOCEDS [6] but also outer anomalies as base clusters based on a limited number of the data points.

As we load the incoming data into memory with each new sample, we expect that the cores previously generated in Si' will have sufficient time to await the arrival of samples from Si'+1, facilitating their transformation into core microclusters. However, for the core microclusters exclusively generated in Si'+1, even if they are not outliers, there may not be enough time for these samples to wait until they form core microclusters. Buffering is presented to address this problem.

The microclusters in Si' +1 with min-densities lower than the Min-Threshold are stored and buffered. As the data in the subsequent two blocks of streaming data are associated with the microclusters, those with a minimum density equal to or more than the Min-Threshold will be stored in memory, while those with

a minimum density below the Min-Threshold will be ignored.

During the anomaly detection stage, outliers can be classified as inner and outer outliers. The identification of a max-density (MD) in a base-cluster serves as the criterion for determining whether the base-cluster is regular, aligning with the assumption proposed by [5]. As per this assumption, attacks form a smaller proportion of the total data traffic that can be differentiated from regular data patterns.

During the increment stage, three processes take place: i) clearing the buffer, ii) storing the prospective microclusters in the storage, and iii) eliminating anomalies.

i) Algorithm 3: Buffer Removal

```
Input: Prospective microclusters, Pmc
If Pmc is not Null
Set Pmc = Null
[end-if statement]
```

ii) Algorithm 4: Saving Prospective Microclusters

```
Input: For all microclusters
Select every outlier microclusters Otmp-outlier from datastream S i+1,  O tmp-outlier ⊂ Omc
For every outlier microcluster O in
Otmp-utlier, do
Add the O into the buffer Pmc = Pmc U O
End
```

iii) Algorithm 5: Deletion of Outliers

We call the microcluster an outlier if they identified as an Outlier-microcluster and if the Base-clusters max-density value is less than the Outlier-Threshold. The maximum density of a base cluster is determined by the sum of all the minimum densities of its core microclusters. When baseclusters are identified as anomalies, they are removed from memory, leading to a reduction in the count of baseclusters.

```
Input:  For each microcluster,
Stage 1: Search all the outlier-microclusters, O(mc) in memory
        Remove O(mc)
Stage 2: for each base-cluster Oi  do
        If MD of  Oi outlier < outlier_threshold
        Select all the core microcluster
        i_Imc in O_i, do
        I_I_mc = (i-1)_Imc U i_Imc
        [End If statement]
[End For Statement]
Outlier = Omc U i_Imc
Stage 3: for every core microcluster  I  in i_Imc, do
        Delete the microcluster I
        Delete the microcluster I from EL' which contains the core microcluster I.
[End statement]
Decrease the number of  baseclusters.
End
```

4. Datasets

4.1 UNSW-NB 15 Dataset

To streamline computation, symbolic features were excluded, and min-max normalization was applied to the numerical features. The dataset was divided into blocks of 500 samples, with each block designated as Sb = 500.

4.2 NSL-KDD Dataset

The second chosen dataset for our experiment is the NSL-KDD dataset. Out of the total 125,973 samples, 67,342 were classified as normal data, while 58,631 were attacks. The NSL-KDD dataset exhibits a significant imbalance similar to the KDD dataset.

4.3 Results and Findings

i) UNSWNB-15

The CADSD parameters for the UNSWNB-15 datasets are configured as follows: Min-Threshold is set to 9, Outlier-Threshold to 18, RadiusMax to 0.315, and RadiusMin to 0.313. Within a predetermined timeframe, we examine 1000 data samples, which are represented by two data stream blocks. Three metrics are used to evaluate the effectiveness of detection: accuracy, false positive rate (FPR), and detection rate (DR).

By modifying the Min-Threshold from 6 to 10 and the outlier-threshold from 14 to 23, we examine how these threshold adjustments influence the DR and FPR of the proposed CADSD. With RadiusMax set to 3.15 and RadiusMin set to 0.313, Table 2 illustrates the sensitivity of the Min-Threshold and Outlier-Threshold under various configurations.

When the outlier-threshold is set below 16, the detection rate drops, indicating that some attack

samples might not be noticed. This is most likely the result of incorrectly classifying as normal base-clusters base-clusters with a high number of samples exceeding the Outlier-Threshold (considered as real outliers). It is advisable to establish the minimum value of the Outlier-Threshold considering the size of the largest outlier base clusters. Moreover, choosing a Min-Threshold below 8 results in a diminished detection rate, as it may incorrectly identify some regular clusters as outliers. Increasing the Min-Threshold from 8 to 10 leads to an increase in false positives. Additionally, regular micro-clusters may lack adequate data to swiftly change into core micro-clusters. Therefore, finding the optimal configuration for Min-Threshold and Outlier-Threshold should consider the maximum size of the outlier base clusters and the minimum size of the core microcluster.

In this scenario, 99% for DR and 10% for FPR, show slight variations around respectively while adjusting the Outlier-Threshold from 16 to 23 and the Min-Threshold from 8 to 11. The optimal outcomes 99.35% detection rate and a 9.34 % false positive rate can be obtained by setting the Min-Threshold to 9 and the Outlier-Threshold to any value in [15,16,17].

The influence of RadiusMax and RadiusMin on the UNSWNB-15 dataset in various configurations is shown in Table 3, where the Min-Threshold is set to 9 and the Outlier-Threshold is set to 18. If RadiusMin is set below 0.3115, the detection rate falls. This might be because base-clusters produce erroneous regular patterns because regular micro-clusters lack sufficient samples to form core micro-clusters. Additionally, since RadiusMax is used to guarantee the smoothness and separation of the micro-clusters, setting it to a high value is not advised. When RadiusMin is changed from 0.31 to 0.315 and RadiusMax from 0.315 to 0.33, the system performs well.Setting RadiusMax between 0.315 and 0.32 results in a detection rate of 99.12%.

When RadiusMin is configured at 0.313 and RadiusMax is set to 0.315, this setup results in the lowest false positive rate of 6.9% among configurations, while achieving a 99.12% detection rate. The minimum false positive rate of 5.9% is attained when RadiusMin is set to 0.31 and RadiusMax is set to 0.315. The detection rate of 98.82% is obtained when RadiusMax is set between 0.325 and 0.33.

When RadiusMin is set to 0.313 and RadiusMax is set to 0.315, it produces the optimal results, yielding a 99.35% detection rate and a 9.34% false positive rate. As depicted in Fig. 3, with the configuration of Min-Threshold = 9, Outlier-Threshold = 18, RadiusMin = 0.313, and RadiusMax = 0.315, the DR and FPR over time exhibit fluctuations. The system experiences the highest false positive rate during the first 10,000 samples of operation. But eventually, the FPR drops until the 60001–70000 sample period. Following that, the level of the trend off and then start to slightly fluctuate at 7.5%. Thus, after a specific number of sample executions by the system, the FPR can increase to 70.5%. Furthermore, on the UNSWNB-15 dataset, the system's detection rate and false positive rate are 6.9% and 99.12%, respectively, with the parameters Min-Threshold = 9, Outlier-Threshold = 18, RadiusMin = 0.313, and RadiusMax = 0.315. The accuracy of the system is 91.6%, which is an additional assessment metric (i.e. accuracy measure) to verify the system's functionality.

Table 2

Sensitivity of min-threshold and outlier-threshold under different configurations

		Outlier-Threshold									
Min-Th		14	15	16	17	18	19	20	21	22	23
7	DR	28.5	70.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5
	FPR	4.9	4.9	5.1	5.1	5.1	5.1	5.1	5.1	5.1	5.1
8	DR	64	86.6	88.4	92.6	92.6	92.6	92.6	91.2	91.2	92.1
	FPR	5.1	6.5	6.9	5.6	5.6	5.6	5.6	7	7.2	7.3
9	DR	64	90.2	90.5	99.12	99.12	99.12	99.12	93.6	94.6	95.3
	FPR	5.2	6.5	6.5	6.9	6.9	6.9	6.9	6.2	6.2	6.2
10	DR	64	70	70	95.2	95.2	95.2	95.2	99.1	99.1	99.1
	FPR	6.5	7.5	7.5	8.4	8.4	8.4	8.4	9.4	10.2	12.4
11	DR	64	80.4	80.4	99.4	99.4	99.4	99.4	99.4	99.4	99.4
	FPR	8.5	8.9	8.9	12.5	12.8	12.9	12.9	12.9	12.9	12.9

Table 3

Influence of radius-max and radius-min on the UNSW-NB15 dataset

Radius Max		Radius-Minimum									
		0.31	0.3115	0.312	0.3125	0.313	0.3135	0.314	0.3145	0.315	0.3155
0.315	DR	65.5	96.12	90.12	99.12	99.12	99.12	99.12	99.12	99.12	99.12
	FPR	5.9	6.2	6.2	6.95	6.9	7.2	7.5	6.99	6.98	7.1
0.32	DR	67.5	99.1	90.12	99.12	99.12	99.12	99.12	99.12	99.12	89.4
	FPR	6.2	6.5	6.67	7.2	7.4	7.69	7.8	7.92	8.1	8.2
0.325	DR	64.1	97.36	97.36	97.36	97.36	97.36	97.36	97.36	97.36	87.36
	FPR	6	6.59	6.4	6.99	6.2	6.65	6.64	6.63	6.63	6.61
0.33	DR	64	78.1	98.82	98.82	98.82	98.82	98.82	98.82	98.82	98.82
	FPR	7.8	8.82	8.8	8.39	8.39	8.35	8.45	8.45	8.6	8.9

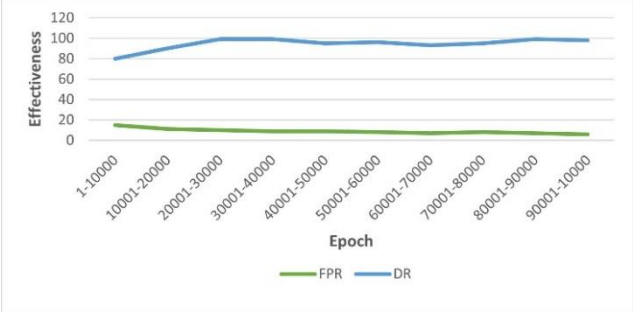

Fig. 3. The Proposed CADSD Method's DR and FPR Over The UNSWNB-15 Dataset

Table 4 provides a comparison of different approaches based on their technique (online/offline), approach, attack type, accuracy, DR, and FPR [18]. The comparison includes seven offline models and one hybrid model. The offline approaches discussed in [19,20,21,22] are examined, and no online approach utilizing the UNSW-NB15 dataset was identified in the literature. Apart from [23], which is a hybrid technique achieving a 100 percent detection rate, the proposed model surpasses all other approaches in terms of detection rate and stands out as an online approach.

The UNSWNB-15 dataset's evolving normal patterns, which are excellent for outlier detection, are highly suitable. The suggested method's accuracy is comparable to other approaches. Compared to earlier techniques, the proposed CADSD has a greater false positive rate (FPR) [20,21]. All these techniques make use of the UNSWNB-15 dataset, which differs greatly in terms of attack and normal data. Conversely, the proposed CADSD does not require human labeling and can identify attacks directly, making it more practical for use in real-world scenarios. Although they cannot be used online, the techniques in [19,20] are fully offline and can yield good results for FPR.

In Table 4, the suggested Cluster-based Anomaly Detection using the Streaming Data (CADSD) approach shows a detection rate that is nearly comparable to other methods. This is explained by the UNSWNB-15 dataset's evolving normal patterns, which are ideal for identifying outliers. The relatively high false positive rate for CADSD, however, suggests the potential occurrence of some normal samples being mistakenly classified as attack samples.

Many normal data instances of the program are misclassified as attacks during the early stages, given that normal patterns are being established from the beginning. Since other unsupervised techniques are trained on complete static datasets, their false positive rates are lower than CADSD's.

Table 4

The table compares methods based on different configurations on the UNSW-NB15 dataset

`	Technique	Approach	Offline/ Online	Accuracy	DR	FPR
proposed CADSD	Clustering for streaming data	Unsupervised	Online	90%	99.12%	6.90%
2023 [17]	IDS-INT	Supervised	Hybrid	99.2%	100%	-
2023 [8]	MSRC	Unsupervised	Offline	95.06%	94.59%	4.96%

2020 [31]	Autoencoder	Unsupervised	Offline	93.40	-	-
2024 [15]	DCRNN	Supervised	Offline	99.06%	98.99%	1.49%
	Decision Tree	Supervised	Offline	76.38%	96.12%	-
2021 [25]	SVM	Supervised	Offline	68.65%	92.71%	-
	ELM	Supervised	Offline	76.51%	92.75%	-
	SDAE-ELM3	Supervised	Offline	72.38%	87.42%	-

ii) NSL-KDD

In Section 4.3.1, the RadiusMin and RadiusMax parameters are set to 0.4 and 0.45, respectively, on NSL-KDD. based on the sensitivity test. The Outlier-Threshold is set to 14 and the Min-Threshold to 3. The suggested CADSD achieves a DR of 90%, an FPR of 7.2%, and an accuracy of 91% with these parameter settings.

The performance comparison between the suggested CADSD and other systems is displayed in Table 5. Regarding accuracy and detection rate, supervised methods [22] and the hybrid supervised and unsupervised method [24] perform better than the proposed CADSD. It still produces superior results for accuracy and detection rate than [25]. Furthermore, because supervised methods need precise classifiers to be trained over an extended period using labeled data, they cannot be applied to online applications. The accuracy of the suggested CADSD and the unsupervised offline methods are comparable [19,26,27,28].

Even though Vaiyapuri et al. [26] in comparison to the suggested CADSD, [27] has a higher detection rate than CADSD it also has a high FPR. Furthermore, when assessed the proposed CADSD outperforms the two-layer structure method [11] regarding the detection rate on the NSL-KDD dataset. Although their FPR are comparable, the proposed CADSD's false alarm rate is marginally higher. Nevertheless, it is necessary to highlight that the proposed CADSD can directly identify attacks without any pre-training of data. In contrast, the two-layer structure method [11] necessitates manual cluster labeling, designating new clusters as either normal or attack clusters.

The data presented in Table 5 indicates that supervised methods outperform unsupervised methods in terms of accuracy. The CADSD method proposed exhibits a superior detection rate compared to the two-layer structure method [11]. This implies that density-based microclusters are more effective in detecting anomalies than evolving Gaussian Mixture Models (GMMs) when applied to the NSL-KDD dataset.

Table 5

The table compares methods based on different configurations on the NSL-KDD dataset

Article	Method	Approach	Offline /Online	Accuracy	DR	FPR
proposed CADSD	Clustering for streaming data GMM based	Unsupervised	Online	91%	90.00%	7.20%
2018 [51]	incremental clustering	Unsupervised	Online	-	98%	2%
2022 [21]	RT-IDS	Supervised	Online	81.87%	70.71%	-
2019 [28]	Autoencoder	Unsupervised	Offline	91.74%	84.68%	-
2019 [29]	Autoencoder	Unsupervised	Offline	87.2%-92.4%	89.73%-92%	7.04%-24.21%
2020 [30]	Autoencoder	Unsupervised	Offline	91.46%	97%	13.40%
2020 [35]	K-means	Unsupervised	Offline	80.91%-90.20%	-	5.2%-8.2%
2021 [23]	Random Forest, Logistic Regres	Hybrid	Offline	93%	94%	9.60%

	sion, SVM					
	Decision Tree	Supervised	Offline	93.15%	93.27%	-
2021 [25]	SVM	Supervised	Offline	93.05%	93.02%	-
	ELM	Supervised	Offline	94.04%	93.18%	-
	SDAE-ELM3	Supervised	Offline	93.56%	93.10%	-

The four techniques mentioned in Reference [22] are offline methods that require prior knowledge of dataset labeling for training classifiers. CADSD surpasses these techniques in terms of accuracy, DR, and FPR. Furthermore, CADSD technique is online and does not necessitate pre-training of classifiers. On the other hand, the approach proposed by Meenal et al. [24] is a hybrid of supervised and unsupervised methods, demonstrating superior accuracy and false positive rate compared to CADSD. However, it is not suitable for online applications due to its lower detection rate compared to CADSD and the requirement for long-term learning in its supervised stage for dataset labeling.

The CADSD algorithm produces the best detection rate results, as Table 6 demonstrates. Due to typical patterns fluctuating continuously and anomalies are quickly found and eliminated, accuracy is increased.

5. Limitations of the Proposed Algorithm

Although the suggested method offers notable benefits, there are inherent limitations that warrant additional evaluation. One possible concern is the approach operates under the presumption that the normal dataset constitutes most of the overall dataset. The most challenging situation occurs when there is a significant surge of attack data samples within one specific time period. In such cases, clusters formed by these particular samples might be mistakenly categorized as regular clusters, resulting in a noticeable decrease in the algorithm's efficacy. Yet another constraint of the suggested CADSD method is the need to predefine parameters of the system, including the inner outlier threshold and outer outlier threshold, relying on expert knowledge. Furthermore, the method still demonstrates significant memory usage. Subsequent efforts could focus on minimizing memory usage and automatically adapting the predefined thresholds to accommodate various applications for detecting network intrusion.

6. Conclusion and Future Work

A novel real-time method for detecting network anomalies is capable of adapting and evolving normal network patterns as the incoming network stream undergoes changes and developments. Additionally, the method incorporates micro-clustering, an essential element in identifying inner outliers through the establishment of a minimum threshold of density and the combining of micro-clusters to form intersections within base-clusters. The recognition of outer outliers within the base clusters is facilitated by the implementation of a global cluster threshold. To avoid the unintentional removal of temporary micro-clusters as legitimate anomalies, storage is used for holding these prospective micro-clusters until the subsequent data points are reached. Finally, two separate data streams are formed in the outliers detection stage, improving the overall effectiveness of CADSD in real-time anomaly detection in network data streams. Outliers are preserved as prospective micro-clusters within the next stream, which includes the latest data samples. If the stored micro-clusters within the buffer exhibit a lower density than the specified minimum threshold density, they are retained in memory during the processing of the next data stream otherwise they are eliminated from the memory. The use of this buffering technique significantly improves detection performance. The efficacy of the suggested technique has been successfully showcased through experimentation with two widely recognized network packet-based datasets: UNSWNB-15 and, NSL-KDD.

Looking ahead, future work could focus on addressing the identified limitations by exploring adaptive thresholding mechanisms that dynamically adjust based on network conditions. Additionally, integrating advanced machine learning techniques, such as deep learning models, could enhance the system's ability to detect complex anomalies and reduce false positives. As technological advancements continue, further improvements in algorithm efficiency and the incorporation of hardware accelerators like GPUs and FPGAs could significantly boost the performance and scalability of the proposed method. These enhancements will ensure the method remains robust and effective in the face of evolving cyber threats.

7. Acknowledgment

I am thankful to my co-authors for useful discussions.

8. Data Availability

The associated data for this paper (https://doi.org/10.57760/sciencedb.17020) can be accessed in the (Science Data Bank) database

(https://www.scidb.cn/en/detail?dataSetId=1c96c2371f80466f88ca0890fd17bada)

9. Author Contribution

MT came up with the idea for the article, she also performed the literature search, coding, and drafting, while the data analysis and synthesis were carried out by AA, NI, and KA. AA critically revised the work and made inputs where necessary.

10. Funding

The author received no support from any organization for the submitted work.

11. Conflict of Interest

There is no conflict of interest in this paper.

12. References

[1] McAfee, "New mcafee report, latest report from mcafee and csis uncov-ers the hidden costs of cybercrime beyond economic impact [online],", 2020.

[2] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse, "Adaptive and on-line network intrusion detection system using clustering and extreme learning machines", Journal of the Franklin Institute, vol. 355, no. 4, pp. 1752–1779, 2018.

[3] C. Yin, S. Zhang, Z. Yin, and J. Wang, "Anomaly detection model based on data stream clustering", Cluster Computing, vol. 22, pp. 1729–1738, 2019.

[4] S. U. Din and J. Shao, "Exploiting evolving micro-clusters for data stream classification with emerging class detection", Information Sci-ences, vol. 507, pp. 404–420, 2020.

[5] J. Dromard, G. Roudiere, and P. Owezarski, "Online and scalable un-supervised network anomaly detection method", IEEE Transactions on Network and Service Management, vol. 14, no. 1, pp. 34–47, 2016.

[6] M. K. Islam, M. M. Ahmed, and K. Z. Zamli, "A buffer-based online clustering for evolving data stream", Information sciences, vol. 489, pp. 113–135, 2019.

[7] J. Ren and R. Ma, "Density-based data streams clustering over sliding windows", 2009 Sixth international conference on fuzzy systems and knowledge discovery, vol. 5, pp. 248–252, IEEE, 2009.

[8] C. Ruiz, E. Menasalvas, and M. Spiliopoulou, "C-denstream: Using domain knowledge on a data stream", Discovery Science: 12th International Conference, DS 2009, Porto, Portugal, October 3-5, 2009 12, pp. 287–301, Springer, 2009.

[9] R. Hyde, P. Angelov, and A. R. MacKenzie, "Fully online clustering of evolving data streams into arbitrarily shaped clusters", Information Sciences, vol. 382, pp. 96–114, 2017.

[10] I. ˇSkrjanc, S. Ozawa, T. Ban, and D. Dovˇzan, "Large-scale cyber-attacks monitoring using evolving cauchy possibilistic clustering", Applied Soft Computing, vol. 62, pp. 592–601, 2018.

[11] E. Bigdeli, M. Mohammadi, B. Raahemi, and S. Matwin, "Incremental anomaly detection using two-layer cluster-based structure", Information Sciences, vol. 429, pp. 315–331, 2018.

[12] C. Wang, H. Zhou, Z. Hao, S. Hu, J. Li, X. Zhang, B. Jiang, and X. Chen, "Network traffic analysis over clustering-based collective anomaly detection", Computer Networks, vol. 205, p. 108760, 2022.

[13] M. J. Idrissi, H. Alami, A. El Mahdaouy, A. El Mekki, S. Oualil, Z. Yartaoui, and I. Berrada, "Fed-anids: Federated learning for anomaly-based network intrusion detection systems", Expert Systems with Applications, vol. 234, p. 121000, 2023.

[14] W. Xiaolan, M. Manjur Ahmed, M. Nizam Husen, Z. Qian, and S. B. Belhaouari, "Evolving anomaly detection for network streaming data",Information Sciences, vol. 608, pp. 757–777, 2022.

[15] C. Khammassi and S. Krichen, "A ga-lr wrapper approach for feature selection in network intrusion detection", computers & security, vol. 70, pp. 255–277, 2017.

[16] A. Saeed, A. Ahmadinia, A. Javed, and H. Larijani, "Intelligent in-trusion detection in low-power iots," ACM Transactions on Internet Technology (TOIT), vol. 16, no. 4, pp. 1–25, 2016.

[17] B. Subba, S. Biswas, and S. Karmakar, "A neural network based system for intrusion detection and attack classification", 2016 Twenty Second National Conference on Communication (NCC), pp. 1–6, Ieee, 2016.

[18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning”, nature, vol. 521, no. 7553, pp. 436–444, 2015.

[19] G. Andresini, A. Appice, and D. Malerba, “Autoencoder-based deep metric learning for network intrusion detection”. Information Sciences, vol. 569, pp. 706–727, 2021.

[20] X. Duan, Y. Fu, and K. Wang, “Network traffic anomaly detection method based on multi-scale residual classifier”, Computer Communi-cations, vol. 198, pp. 206–216, 2023.

[21] G. S. C. Kumar, R. K. Kumar, K. P. V. Kumar, N. R. Sai, and M. Brahmaiah, “Deep residual convolutional neural network: An efficient technique for intrusion detection system”, Expert Systems with Applications, vol. 238, p. 121912, 2024.

[22] Z. Wang, Y. Liu, D. He, and S. Chan, “Intrusion detection methods based on integrated deep learning model”, Computers & security, vol. 103, p. 102177, 2021.

[23] I. Fosi´c, D. ˇZagar, K. Grgi´c, and V. Kri ˇzanovi´c, “Anomaly detection in netflow network traffic using supervised machine learning algorithms”, Journal of Industrial Information Integration, p. 100466, 2023.

[24] M. H. L. Louk and B. A. Tama, “Dual-ids: A bagging-based gradient boosting decision tree model for network anomaly intrusion detection system”, Expert Systems with Applications, vol. 213, p. 119030, 2023.

[25] X. Duan, Y. Fu, and K. Wang, “Network traffic anomaly detection method based on a multi-scale residual classifier”, Computer Communications, vol. 198, pp. 206–216, 2023.

[26] T. Vaiyapuri and A. Binbusayyis, “Application of deep autoencoder as a one-class classifier for unsupervised network intrusion detection: a comparative evaluation”, PeerJ Computer Science, vol. 6, p. e327, 2020.

[27] S. D. Bay, D. Kibler, M. J. Pazzani, and P. Smyth, “The uci kdd archive of large data sets for data mining research and experimentation”, ACM SIGKDD explorations newsletter, vol. 2, no. 2, pp. 81–85, 2000.

[28] L. F. Carvalho, S. Barbon Jr, L. de Souza Mendes, and M. L. Proenca Jr, “Unsupervised learning clustering and self-organized agents applied to help network management”, Expert Systems with Applications, vol. 54, pp. 29–47, 2016.

