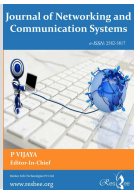
Vol.8 Issue 3 July 2025

Journal of Networking and Communication Systems

Received 21 June, Revised 24 June, Accepted 5 July



A Novel Unsupervised Framework for DDoS Attack Detection Using Adaptive HDBSCAN and Ensemble-Based Feature Selection

M Venkatesh
Department of ECE
VVITU, Nambur
venkatesh.munagala@gmail.com

Abstract: A Distributed Denial of Service (DDoS) attack floods a target system, server, or network with massive traffic from numerous sources, overwhelming it and disrupting normal functionality. Such attacks often leverage botnets—large groups of compromised devices—that generate malicious traffic at scale, making detection increasingly complex. This results in service unavailability for legitimate users, causing downtime, damage, or the exploitation of weaknesses. Existing models face challenges, including overfitting to training data, which limits their ability to generalize, and underfitting, which leads to missing critical attack patterns. Moreover, conventional detection methods struggle to adapt when attackers continuously evolve their strategies, leading to reduced robustness in real-world deployment. To overcome these challenges, this research paper presents an Adaptive Hierarchical Density-Based Spatial Clustering of Applications With Noise Approach (Ada-HDBSCAN) for detecting DDoS attacks using clustering. The process begins by simulating a cloud model and collecting input log data from the NSL-KDD dataset. Feature selection is performed using Gini Impurity-based Weighted Random Forest (GIWRF) and Sequential Forward Search (SFS) techniques to identify the most relevant features from the input data. This hybrid feature selection strategy ensures that redundant or noisy attributes are removed, thereby improving both efficiency and accuracy. Subsequently, clustering-based detection is carried out using Ada-HDBSCAN, which classifies the data as either DDoS-affected or normal. Ada-HDBSCAN is an innovative approach developed by adaptively modifying the clustering parameters of Hierarchical Density-Based Spatial Clustering of Applications With Noise (HDBSCAN). The developed Ada-HDBSCAN method achieved an Adjusted Rand Index (ARI) of 0.966, a V-measure of 0.959, and a Silhouette score of 0.978, respectively.

Keywords: Cloud security, Feature Selection, DDoS Attacks Detection, Clustering, Unsupervised Learning Techniques

Nomenclature

Abbreviation	Description
DDoS	Distributed Denial of Service
NSL-KDD	Network Security Laboratory – Knowledge Discovery and Data Mining dataset
GIWRF	Gini Impurity-based Weighted Random Forest
SFS	Sequential Forward Search
ADA-HDBSCAN	Adaptive Hierarchical Density-Based Spatial Clustering of Applications with Noise
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ML	Machine Learning
IDS	Intrusion Detection System
IRS	Intrusion Response System

1. Introduction

Cloud computing has emerged as a transformative technology that has recently garnered significant interest from both industry professionals and academic researchers [1]. As a model of Internet-based computing, it offers users scalable, cost-effective, and convenient access to computing power and data storage. This on-demand availability of resources allows businesses to quickly adapt to varying workloads and operational demands without the need to manage physical hardware or software infrastructure. One of the core strengths of cloud computing lies in its ability to maintain high service availability. This is achieved through the use of multiple data centers and robust backup mechanisms, which help to minimize

service disruptions and enhance reliability for end-users [2]. However, the same distributed and shared nature of cloud systems that provides resilience also introduces multiple attack vectors, making them prime targets for adversaries. Despite its advantages, security concerns remain a major hurdle to the widespread adoption of cloud computing. Many organizations and research institutions are still hesitant to fully entrust third-party service providers with their digital assets. Unlike traditional IT infrastructures, where data remains within the administrative control of the organization, cloud computing shifts this responsibility to external entities [3]. This shift of responsibility has heightened the urgency for intelligent and automated security solutions that can adapt to dynamic network conditions. As the volume of data continues to grow rapidly, effective data processing techniques are becoming increasingly important. Feature selection is one such technique used during the data pre-processing stage. It plays a critical role in improving the efficiency of classification tasks by identifying and retaining the most relevant features of a dataset while eliminating redundant or irrelevant ones [4]. By reducing the dimensionality of data, feature selection not only decreases computational load but also enhances classification accuracy. This ultimately contributes to more effective intrusion detection and overall system performance [5]. This shift of responsibility has heightened the urgency for intelligent and automated security solutions that can adapt to dynamic network conditions.

DDoS attacks are among the most prevalent forms of cyberattacks, aiming to disrupt or completely deny services to legitimate users. These attacks typically target network infrastructure or servers by overwhelming them with excessive traffic, ultimately degrading performance or rendering services unavailable. A key factor contributing to the frequency of DDoS attacks is the variety of methods through which they can be executed. Notably, they can be launched at virtually any layer of the Open Systems Interconnection (OSI) communication model, making them highly versatile and challenging to mitigate [6]. For example, volumetric attacks at the network layer exploit bandwidth, while application-layer attacks target specific services, each requiring different mitigation strategies. DDoS attacks pose serious security threats by flooding systems with malicious requests, leading to system crashes, reputational damage, and substantial financial losses for the affected organizations [7]. Traditional detection and prevention mechanisms often fall short when faced with the scale and sophistication of modern DDoS attacks, particularly when large volumes of data are involved. In recent years, attackers have increasingly used encrypted traffic and botnet-as-a-service platforms, further complicating timely detection. To address these limitations, Machine Learning (ML) techniques have gained prominence in cybersecurity. ML algorithms are capable of learning patterns from historical data and making predictions based on those patterns, which makes them effective in detecting both known and unknown (zero-day) threats. Unsupervised learning holds particular significance in cloud security, where threats are constantly evolving. By identifying anomalies that deviate from established patterns, it can reveal previously unknown attack vectors or zero-day vulnerabilities, making it a vital tool in proactive threat detection [8]. Unlike supervised approaches that depend heavily on labeled datasets, unsupervised learning reduces reliance on prior knowledge and is thus more scalable for real-world applications. Clustering groups data by maximizing similarities within clusters and differences between them. It's an unsupervised method used in fields like data mining, bioinformatics, and ML without prior class labels [9].

This research paper proposes a novel method for detecting DDoS attacks through clustering, utilizing the Ada-HDBSCAN approach. The methodology begins by simulating a cloud environment, and the input log data is collected from the NSL-KDD dataset. Thereafter, to enhance the quality of the input data, feature selection is applied using a combination of GIWRF and SFS, effectively identifying the most significant attributes for classification. This dual-stage feature selection framework ensures robustness against noisy and redundant attributes, providing a cleaner input for clustering. Following this, the refined data undergoes clustering using the Ada-HDBSCAN algorithm, which categorizes traffic as either normal or DDoS-affected. Ada-HDBSCAN is developed by modifying the clustering parameters of the HDBSCAN method adaptively, making it more effective in identifying complex attack patterns within dynamic network environments. The adaptiveness of this method allows it to adjust clustering thresholds in response to shifting traffic characteristics, thereby improving detection accuracy over static clustering approaches.

The key contribution of this paper is demonstrated as follows,

- ***Proposed Ada-HDBSCAN Approach for DDoS attack detection*:** In this context, a clustering-based method for the detection of DDoS attacks is proposed using Ada-HDBSCAN. Ada-HDBSCAN is developed by integrating the HDBSCAN method with adaptive mechanisms, improving its ability to detect complex and evolving attack patterns. This contribution lies not only in enhancing accuracy but also in demonstrating the scalability and applicability of unsupervised clustering for large-scale cloud environments.

The remaining part of the paper is structured in the following manner: Section 2 outlines existing approaches to DDoS attack detection. Section 3 elaborates on the introduced Ada-HDBSCAN methodology.

Section 4 covers the experimental design and results analysis. Section 5 provides the conclusion and proposes future research directions.

2. Motivation

Detecting DDoS attacks is essential to maintain network uptime, stop service interruptions, and reduce financial damage. The increasing reliance on cloud services for critical business operations means even short periods of downtime can have severe financial and reputational consequences. Existing methods for detecting DDoS attacks encounter various challenges, including high false positive rates, difficulty in managing large and dynamic traffic volumes, and issues related to overfitting and underfitting. False positives can waste system resources by misclassifying normal traffic as malicious, while false negatives leave networks vulnerable to undetected attacks. Hence, the Ada-HDBSCAN method is proposed as a solution to these limitations, aiming to enhance the efficiency and accuracy of DDoS detection.

2.1 Literature Survey

Dasari, S. and Kaluri, R. [10] developed a Light Gradient Boosting Machine (LGBM) model for effective classification of DDoS attacks in distributed networks. This model demonstrated strong generalization capabilities and a reduced error rate. Its lightweight nature made it suitable for real-time processing, but scalability across heterogeneous datasets remained a limitation. However, this model failed to focus on the integration of additional ensemble ML and Deep Learning (DL) models, as well as optimization techniques that could have further improved forecasting accuracy. The Deep Clustering Variational Auto-Encoder (DCVAE) and the Deep Clustering Support Vector Data Description Variational Auto-Encoder (DC-SVDD-VAE) were designed by Nguyen, V.Q., et al. [11], to enhance the learning of latent features for network anomaly detection. This approach demonstrated high effectiveness and was well-suited for real-time applications. The ability of these models to extract latent features provided robustness against unseen attacks, highlighting the promise of deep clustering methods. Still, this approach did not explore optimal hyperparameter selection strategies, which are essential to ensure the models operate at their full potential. Nguyen, T.L., et al. [12] introduced the Spatial Location Constraint Prototype Loss (SLCPL) combined with a Fuzzy C-Means (FCM) method to detect unknown DDoS attacks. This method proved to be more reliable and effectively mitigated overfitting during both training and evaluation phases. By leveraging fuzzy clustering, the model was able to handle uncertainty in traffic behavior, but the reliance on a simplified CNN backbone limited its learning capacity. Nevertheless, the method did not explore more advanced Convolutional Neural Network (CNN) architectures to replace the relatively simple 1D AlexNet used as the core of the system. The Deep Nested Clustering Auto-Encoder (DNCAE) and Deep Clustering Hierarchical Auto-Encoder (DCHAE) models were developed by Nguyen, V.Q., et al. [13] for anomaly-based cyberattack detection. This technique required minimal processing time and significantly reduced computational complexity. Its efficiency made it attractive for large-scale data analysis, yet the lack of adaptive clustering mechanisms reduced robustness against evolving attack vectors. Nevertheless, the technique did not investigate alternative data representation learning techniques that could further enhance performance. John, A., et al. [14] proposed the Cluster-Based Wireless Sensor Network and Variable Selection Ensemble Machine Learning Algorithms (CBWSN_VSEMLA) framework for Denial-of-Service (DoS) attack detection using variable selection methods. While the approach improved scalability and reduced energy consumption. It also showcased the effectiveness of combining variable selection with ensemble learning for distributed systems. But this model did not include an Intrusion Response System (IRS) capable of actively responding to detected attacks by isolating or disconnecting compromised nodes from the network.

2.2 Major Challenges

The primary challenges in current DDoS attack detection are outlined below.

- The LGBM [10] model demonstrated high computational speed, low memory usage, and reduced overfitting. However, it struggled to scale across diverse datasets. This indicates that while lightweight models are beneficial for efficiency, they may not generalize well when applied to heterogeneous traffic environments typical of real-world networks.
- DCVAE and DC-SVDD-VAE [11] reduced dimensionality, improved feature separation, and minimized false alarms. However, these models failed to extend efforts toward developing additional data representation learning models to enhance latent feature extraction capabilities, specifically for network anomaly detection. As a result, their applicability in continuously evolving network environments remains constrained, especially when confronted with zero-day or stealthy attacks.

- DNCAE and DCHAE [13] were more accurate and enhanced clustering compactness. However, they failed to explore adaptive clustering to enhance robustness against evolving attack patterns. The absence of adaptive mechanisms means that these models risk becoming obsolete as attackers alter their strategies to bypass static detection boundaries.
- Detecting DDoS attacks presents significant challenges due to the high volume and velocity of incoming traffic, making it difficult to differentiate between legitimate surges and malicious activity. Rapidly evolving attack patterns, often leveraging distributed sources and encryption, further complicate detection. Moreover, the increasing use of IoT devices in botnets introduces massive traffic diversity, further challenging conventional detection frameworks.

3. Proposed Adaptive Hierarchical Density-Based Spatial Clustering of Applications with Noise for DDoS Attack Detection

This research paper introduces a novel method, referred to as Ada-HDBSCAN, for DDoS attack detection. Unlike conventional clustering methods that rely on fixed parameters, Ada-HDBSCAN introduces an adaptive mechanism that dynamically adjusts to the varying density of network traffic. The process begins by simulating a cloud model and gathering input log data from the NSL-KDD dataset [15]. The use of NSL-KDD ensures standardized benchmarking while avoiding duplicate records that could bias results, making it suitable for validating intrusion detection frameworks. Feature selection is then carried out using GIWRF [16] and SFS [17] techniques to identify the most relevant features from the input logs. This dual-stage feature selection reduces noise, improves interpretability, and enhances the clustering algorithm's ability to focus on discriminative attributes. Following this, clustering-based detection of DDoS attacks is performed using Ada-HDBSCAN, which categorizes the data as either DDoS-attacked or normal. By grouping traffic patterns based on density, Ada-HDBSCAN effectively distinguishes legitimate bursts of activity from malicious flooding attempts. Ada-HDBSCAN is a new approach that incorporates adaptive concepts for modifying the clustering parameters of the HDBSCAN [18]. A general overview of the Ada-HDBSCAN method for DDoS attack detection in clustering is shown in Fig 1. The schematic representation highlights the integration of feature selection, adaptive clustering, and traffic classification, forming a complete pipeline for unsupervised DDoS detection.

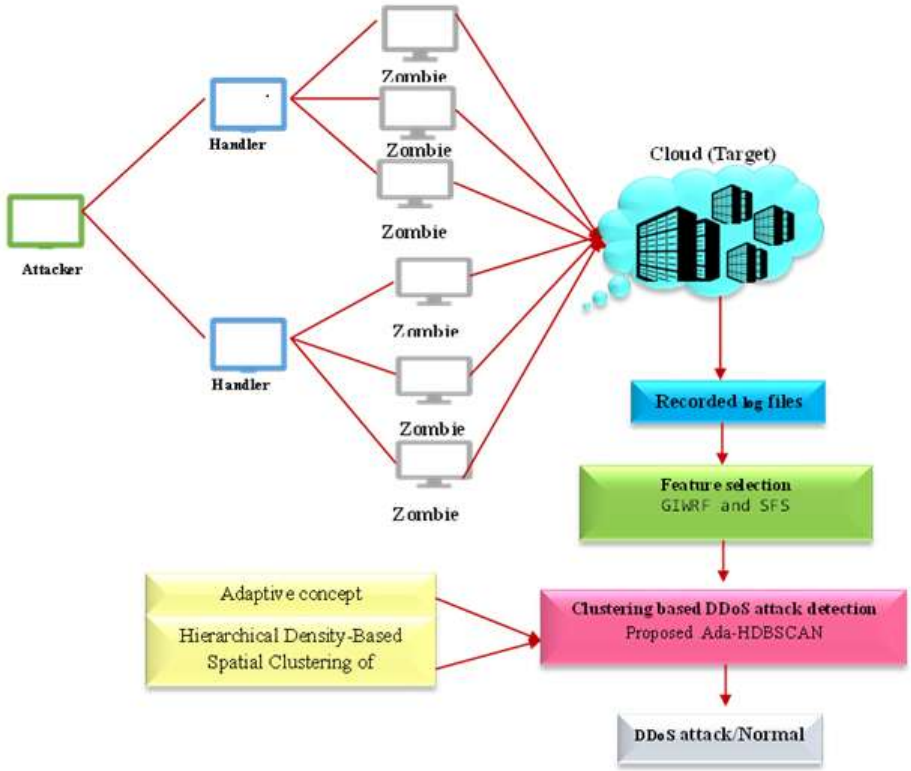


Fig 1. *Schematic representation of the developed Ada-HDBSCAN method for DDoS attack detection in clustering*

3.1 Recorded Log File

Initially, the logfile containing the IP address, protocol type, and related log details of the IP address is assimilated from the dataset. These log files serve as the primary evidence of network activity, capturing both normal traffic and potential attack traces that are crucial for analysis. The NSL-KDD dataset [15] offers a comprehensive set of features, including protocol type, duration, dst_bytes, src_bytes, service, flag, urgent, hot, wrong_fragment, land, and logged-in status, among others. The NSL-KDD dataset is articulated as G_p .

3.2 Ensemble-Based Feature Selection

Ensemble-based feature selection integrates multiple feature selection methods to identify the most relevant attributes in large, complex log datasets. Unlike single-method approaches, which may suffer from bias or miss subtle patterns, ensemble techniques aggregate different perspectives to achieve more reliable results. By combining the strengths of different algorithms, it minimizes bias and increases the robustness of the selected features. This robustness is especially important in cybersecurity, where attack traffic can mimic normal behavior, and subtle distinctions are critical. This method enhances the performance of downstream tasks such as intrusion detection and anomaly classification by focusing on the most informative data points. Ensemble-based feature selection improves model accuracy during analysis and reduces overfitting. Balancing generalization with precision, it ensures that the model adapts well to both training and unseen traffic patterns. In this approach, ensemble-based feature selection is fed with the input data G_p and produces a selected feature set labeled as Z . The resulting feature subset represents a refined view of the dataset, where redundant, noisy, and irrelevant attributes have been filtered out. The feature selection is carried out using techniques, including GIWRF [16] and SFS [17].

3.2.1 GIWRF Feature Selection

GIWRF [16] employs Gini impurity to build weighted decision trees, effectively identifying important features in high-dimensional and noisy cloud datasets. This makes it particularly useful in cybersecurity, where traffic logs often contain redundant or irrelevant attributes that can obscure meaningful attack patterns. Its integration with the Random Forest (RF) algorithm enables it to improve detection speed and manage complexity. Since Random Forests consist of multiple decision trees, GIWRF leverages ensemble voting to reduce variance while emphasizing the most influential attributes. RF evaluates feature importance through learned decision structures. Traditional ML models often assume balanced class distributions, leading to biased outcomes in imbalanced datasets. GIWRF mitigates this by introducing a weighting scheme during the impurity calculation, represented as $\alpha(a)$ improving the alignment between selected features and the target variable in skewed class distributions. This weighting ensures that minority attack classes are given appropriate importance, preventing critical features from being overshadowed by dominant normal traffic patterns. Gini impurity is a metric used to evaluate the effectiveness of a split in classifying samples into two categories within a decision tree node. Its formula is given by,

$$\alpha(a) = 1 - e_e^2 - e_g^2 \tag{1}$$

wherein e_e symbolizes the positive sample fractions, the negative sample fraction is designated as e_g , and any node in the decision tree is exemplified as a . A lower impurity value indicates a cleaner separation of classes, meaning the selected feature contributes more effectively to distinguishing between normal and attack traffic. An optimal split $\Delta\alpha_b(a,Q)$ at each node a in the ensemble of Q weighted trees contributes to the reduction of Gini impurity. The reduction attributed to each feature is then aggregated across all nodes in the forest, yielding the feature importance, and is described by,

$$I_f(b) = \sum_{Q} t_{e,g} \sum_{a} \Delta\alpha_b(a,Q) \tag{2}$$

Here, $t_{e,g}$ delineates the imbalanced class distribution, and the frequency with which a feature is utilized is quantified through its Gini importance and is signified as I_f . In practice, features with consistently high importance scores across trees form the backbone of the detection system, ensuring robustness and interpretability. The features selected using GIWRF is specified as M_1 .

3.2.2 SFS

In SFS [17], features are iteratively included until a predefined set is achieved to carry out feature selection. Unlike filter methods, which rank features independently, SFS accounts for inter-feature relationships by assessing performance improvements after each addition. SFS efficiently identifies

important features by beginning with an empty set and progressively adding one feature that enhances model performance. This greedy approach ensures that the most discriminative features are prioritized early, reducing the risk of overfitting. This method lowers computational cost, prevents overfitting, and boosts accuracy. Its step-by-step nature also provides transparency, making it easier to interpret why certain features are included in the final subset. SFS is straightforward, easy to interpret, and well-suited for high-dimensional data with many irrelevant features. The criterion for including a feature n^+ is expressed as,

$$n^+ = \arg\max T_{n \in W_n} \left(W_n + n \right) \tag{3}$$

In this context, the current set of selected features is delineated as W_n, the number of candidate features in W_n is epitomised as n, the evaluation criterion is denoted as $T(\cdot)$, and the feature subset selected using SFS is signified as M_2. The evaluation criterion can vary depending on the application, but in this work, clustering quality measures (e.g., Silhouette score) are emphasized to guide the selection process.

The final feature vector is formulated by combining the features attained using the GIWRF and SFS, as shown below,

$$Z = M_1 \cup M_2 \tag{4}$$

In the above equation, the selected features are indicated as Z.

3.3 DDoS Attack Detection using Proposed Ada-HDBSCAN

DDoS attack detection focuses on recognizing unusual traffic patterns intended to overload systems or networks. Unlike signature-based methods that rely on known attack profiles, anomaly-based clustering approaches are better equipped to detect new and evolving threats. Generally, this is done by methods like traffic monitoring, Machine Learning (ML), and behavior analysis. Each of these methods has strengths and weaknesses—while monitoring provides real-time visibility, ML adds predictive capabilities, and behavior analysis helps capture deviations from normal baselines. Detecting attacks promptly allows for swift response, preventing major disruptions and maintaining service availability. Current methods often face challenges like high false positives and face difficulty differentiating legitimate traffic spikes from attacks. Here, clustering-based detection of DDoS attacks is carried out employing Ada-HDBSCAN. In this context, the selected feature Z is utilized for DDoS attack detection using the proposed Ada-HDBSCAN method, which effectively analyzes the data to classify the output as either normal or under attack. This process leverages the refined feature set to reduce noise and focus the clustering process on attributes most indicative of malicious behavior. Ada-HDBSCAN is an innovative technique that combines adaptive techniques for updating the clustering parameters of the HDBSCAN [18]. Clustering groups similar network traffic patterns to spot anomalies. In DDoS detection, it distinguishes normal traffic clusters from abnormal ones caused by attacks. This separation ensures that benign traffic patterns, even during legitimate surges, are grouped distinctly from coordinated attack traffic. By analyzing these clusters, systems can detect unusual traffic spikes, enabling the identification and mitigation of DDoS attacks without depending on predefined attack signatures.

3.3.1 Hierarchical Density-Based Spatial Clustering of Applications with Noise Clustering

HDBSCAN is developed by modifying the clustering parameters of HDBSCAN adaptively. This adaptive adjustment ensures that the algorithm can dynamically respond to changes in data distribution, which is critical in cloud-based environments where traffic patterns fluctuate rapidly. HDBSCAN [18] represents a sophisticated extension of density-based clustering techniques. By integrating Density-Based Spatial Clustering of Applications with Noise (DBSCAN) with hierarchical clustering strategies, it generates a multi-level cluster hierarchy that captures the intrinsic structure of data. This hierarchy allows the algorithm to zoom into both fine-grained and coarse-grained patterns, offering flexibility in analyzing traffic behavior. This approach offers significant advantages over K-means, particularly in scenarios involving non-convex clusters or heterogeneous density distributions, and does not require a priori specification of the number of clusters. HDBSCAN, unlike *K*-means, does not require a predefined number of clusters. Instead, it determines the optimal clustering based on the intrinsic density structure of the data. This eliminates human guesswork and reduces the likelihood of biased clustering outcomes. It also effectively identifies and removes noise points, addressing the issue of over-segmentation that DBSCAN may encounter in uniformly dense regions. Noise handling is crucial here because real-world network logs often contain incomplete, irrelevant, or anomalous entries that could otherwise distort clustering results. Furthermore, HDBSCAN is capable of detecting clusters with varying densities, non-convex shapes, and arbitrary boundaries, offering greater robustness than DBSCAN. Such flexibility ensures that even irregular or fragmented attack traffic patterns can be grouped accurately. In this context, the selected

features Z is provided as an input for HDBSCAN clustering. HDBSCAN follows these steps to construct the hierarchical clustering structure,

Step 1: Core distance calculation: For a data point z_y, the distance to the closest neighbor within a set of minimum points $\min_{pnt}$ is computed as the core distance k_{cr}. If this distance is less than or equal to the neighborhood radius ε, i.e., $k_{cr}(z_y) \le \varepsilon$, then z_y is considered an ε-core object. This step ensures that only dense regions of data points are considered as potential cluster cores, filtering out sparse or noisy points.

Step 2: Mutual reachability distance: To compute the mutual reachability distance between two points z_y and z_x, the following expression is used,

$$k_{mutrea}(z_y, z_x) = MAX\left(k_{cr}(z_y), k_{cr}(z_x), k(z_y, z_x)\right) \tag{5}$$

The Euclidean distance between the points is signified as $k(z_y, z_x)$. Mutual reachability modifies the raw Euclidean distance by accounting for local density, ensuring that clustering is not biased by varying densities across the dataset.

Step 3: Mutual reachability graph construction: In constructing the mutual reachability graph $P_{\min_{pnt}}$, each data point is represented as a vertex, and edges are created between vertices with weights determined by the mutual reachability distances between the points. This graph representation provides a structured way to model relationships between traffic records, laying the groundwork for hierarchical clustering.

Step 4: Minimum Spanning Tree (MST): To create a connected structure with the least total edge weight, the MST of $P_{\min_{pnt}}$ is calculated. Additionally, self-loops weighted by core distances are added to accommodate noise and isolated points, leading to the formation of an extended MST. The MST effectively condenses the graph while preserving critical connectivity, making the clustering process computationally efficient.

Step 5: Cluster extraction: Edges of the extended MST are removed in descending order of their weights to prune the graph, yielding a hierarchical dendrogram. Cluster extraction is performed based on the stability measure $C(S_d)$, which is characterized as,

$$C(S_d) = \sum_{z_f \in S_d} \left(\frac{1}{\varepsilon_{MIN}(z_f, S_d)} - \frac{1}{\varepsilon_{MAX}(S_d)} \right) \tag{6}$$

wherein, the value $\varepsilon_{MIN}(z_f, S_d)$ exemplifies the minimum scale that S_d belongs to the cluster z_f, whereas $\varepsilon_{MAX}(S_d)$ denotes the maximum scale before the cluster's division or disappearance. A higher stability score reflects a cluster that persists across multiple scales, indicating its reliability. Clusters exhibiting the highest stability are chosen as the final results. The clustering process prioritizes clusters with superior stability scores for final selection. This ensures that the final clusters represent meaningful traffic groupings rather than transient or weak associations. By clustering network traffic patterns based on similar complexity, this method establishes a structured framework that supports targeted secondary analysis and boosts the accuracy of attack detection. This grouping helps isolate anomalous activities more efficiently, improving the overall effectiveness of identifying DDoS and other network attacks.

3.3.2 Modification of Clustering Parameters

Modifying clustering parameters [19] involves tuning key settings such as neighborhood radius and the minimum number of neighboring points. These parameters directly influence the sensitivity of the clustering process—small values may split genuine clusters, while large values may merge distinct traffic types into a single group. These adjustments enhance the detection of data patterns, improve cluster accuracy, and tailor the clustering algorithm to fit specific dataset characteristics and analytical goals more effectively. In DDoS detection, this adaptiveness is vital, since network traffic exhibits significant variability depending on user behavior, time of day, and the type of attack launched. Modifying clustering parameters involves adjusting values like distance thresholds, similarity metrics, or neighborhood size to more effectively group network traffic patterns. Such flexibility allows Ada-HDBSCAN to outperform static clustering methods, which often fail when faced with heterogeneous traffic densities. These changes improve the ability to differentiate between legitimate and malicious traffic, enhancing the accuracy of DDoS attack detection while minimizing false positives and missed threats.

3.3.2.1 Calculation of Neighbourhood Radius

The calculation of the neighborhood radius ε involves measuring the Manhattan distances between a point and its surrounding neighbors. Unlike Euclidean distance, Manhattan distance is less sensitive to outliers, making it particularly useful when analyzing network data that contains sharp variations. By analyzing the growth rates of these distances and identifying significant changes known as mutation points, an appropriate ε value can be selected. This data-driven approach avoids manual tuning and ensures that the radius dynamically reflects local density variations. This method offers several advantages, such as enhancing clustering precision, reducing sensitivity to noise, adapting to variations in local data density, and improving the overall robustness of the clustering process [19]. As a result, the algorithm can consistently form meaningful clusters even in scenarios where traffic is irregular or highly skewed.

(i) Determining the Manhattan distance from other points to the current point: Let $\left(q_j\right)$ be the current point and $\left(r_j\right)$ a point from the bounding box query. The Manhattan distance Dis_j is determined employing equation (7), wherein the overall count of points in the bounding box is delineated as x.

$$Dis_j = \sum_{j=1}^{x} \left| q_j - r_j \right| \tag{7}$$

This step provides the baseline measure of how spread out the neighboring data points are around the reference point.

(ii) Evaluating the growth rate across each consecutive distance interval: To compute the growth rate m_j, distances are first sorted as $\{Dis_1, Dis_2, \ldots Dis_x\}$. The growth rate is derived according to Equation (8), which x refers to the number of data points enclosed by the bounding box.

$$m_j = \left(Dis_{j+1} - Dis_j\right) \Big/ Dis_j \tag{8}$$

Growth rate analysis highlights abrupt changes in distance spacing, which signal potential cluster boundaries.

(iii) Compute the change value: Given the growth rate list $\{m_1, m_2, \ldots m_{x-1}\}$, the change value p_j is determined within each pair of consecutive growth rates according to Equation (9).

$$p_j = \left| m_{j+1} - m_j \right|, j = 1, 2, \ldots x-2 \tag{9}$$

This step strengthens robustness by capturing relative shifts in density, rather than relying only on absolute distance differences.

(iv) Calculating Eps: The mutation point is determined by locating the index of the maximum growth rate change p_j, and the first distance value after this point is chosen as the Neighborhood Radius ε.

This ensures that Eps is chosen objectively based on the data itself, rather than arbitrary thresholds, making clustering adaptive to real-world variability.

Identifying mutation points by examining the differences between consecutive growth rates is a more robust technique. Methods based only on consecutive distance growth rates tend to be noisy, especially with minor distance variations. By calculating the change between growth rates instead, noise is reduced, which makes mutation point detection more dependable. This approach ultimately enhances cluster quality, allowing Ada-HDBSCAN to detect both small-scale anomalies and large-scale attack patterns more effectively.

3.3.2.2 Calculation of the Minimum Number of Neighbouring Points

To determine the minimum number of neighbouring points $P_{\min_{pnt}}$, bounding boxes are created around each point with a radius defined by ε. This ensures that the density around each data point is measured consistently, which is critical for detecting localized anomalies in traffic patterns. The count of neighbors within each bounding box is recorded, and the neighbors are then sorted. From this sorted list, the first quartile (X_1) and third quartile (X_3) are calculated. Finally, $P_{\min_{pnt}}$ is set to the average of X_1 and X_3. This averaging strategy provides a balanced threshold, avoiding the extremes of too strict or too lenient clustering.

(i) Within the bounding box constructed in the earlier step, each point $\left(q_j, r_j\right)$ is enclosed by a newly constructed bounding box. The number of points g_j falling inside this new box is counted. The bounding box size corresponds to the calculated ε. These counts of points inside each new bounding box define the set Pts_{cnt} shown in Equation (10).

$$Pts_{cnt} = \{g_1, g_2, \ldots g_x\} \tag{10}$$

This step ensures that the calculation of neighboring density is localized and adaptive to each data point's environment.

(ii) First, sort the Pts_{cnt} in ascending order, then determine the upper quartile $X_3 = Pts_{cnt}\left[c_1(x+1)\right]$ and the lower quartile $X_3 = Pts_{cnt}\left[c_2(x+1)\right]$, as described in Equations (11) and (12). Here, the random number is delineated as a, c_1 and c_2 designates the adaptive polynomials generated using the silhouette score v.

$$c_1 = v^2 + v + a \tag{11}$$

$$c_2 = -v^2 - v + (1 - a) \tag{12}$$

By incorporating the silhouette score into the process, the method ties feature selection directly to clustering quality, ensuring that the chosen MinPts parameter aligns with optimal cluster separation.
(iii) Determine the mean value of X_1 and X_3, as illustrated in Equation (6).

$$X_{13} = \left(X_1 + X_3\right)/2 \tag{12}$$

Taking the average rather than a single quartile ensures that the method is neither biased toward sparse traffic nor overly tuned to dense traffic, thus improving stability.

Averaging X_1 and X_3 in the quartile method minimize the impact of extreme values, providing a more balanced approach for calculating $\min_{pnt}$. While X_2 (the median) is a central measure, it may not effectively capture the entire data distribution, especially if the data is skewed. X_1 corresponds to low-density regions, while X_3 represents high-density areas. Solely using X_2 can lead to cluster over-expansion in dense areas or incorrectly identify small clusters as noise in sparse regions. Therefore, the averaging technique prevents over-generalization in dense clusters while preserving sensitivity to rare anomalies in sparse regions. By averaging X_1 and X_3, the method becomes more adaptive to varying densities, making it more robust against noise and outliers. This adaptiveness is essential in DDoS detection, where traffic volumes can fluctuate rapidly between low and high-density phases during an attack.

Furthermore, the result from the developed HDBSCAN demonstrates a DDoS attack or normal.

4. Results and Discussion

An assessment of the Ada-HDBSCAN method for DDoS attack detection is carried out in this section, with performance results benchmarked against prevailing approaches on the basis of several metrics. This comparative evaluation highlights not only the accuracy of the proposed model but also its scalability and robustness under diverse traffic conditions.

4.1 Experimental set-up

Python 3.9.11 is used for implementing the developed Ada-HDBSCAN technique for DDoS attack detection. The experiments were executed in a controlled environment to ensure reproducibility, with sufficient computational resources allocated to handle the NSL-KDD dataset efficiently.

4.2 Performance Metrics

The evaluation of the performance of Ada-HDBSCAN is conducted using metrics such as the Adjusted Rand Index (ARI), V-measure, and Silhouette score [20]. These three complementary metrics provide a holistic view of clustering quality: ARI measures similarity with ground truth, V-measure captures homogeneity and completeness, and Silhouette evaluates cohesion and separation.

4.2.1 Adjusted Rand Index (ARI)

ARI [20] is an evaluation metric used to assess how similar two clustering solutions are. It shares some mathematical properties with prediction accuracy, but differs in that it can be used without the need for original class labels. This makes ARI particularly suitable for unsupervised tasks like DDoS detection,

where true attack patterns may not always be labeled. The Rand Index evaluates similarity by examining all sample pairs and determining if they are grouped similarly or differently in both the predicted and actual clustering. The raw score is then adjusted to remove random chance, resulting in the ARI. The ARI (A_{ARI}) is calculated as,

$$A_{ARI} = \frac{(B - Expected\ B)}{(MAX(B) - Expected\ B)} \tag{13}$$

here, the rand index is delineated as B. A higher ARI score indicates strong alignment between predicted clusters and actual traffic patterns, validating the effectiveness of the clustering approach.

4.2.2 V-Measure

The V-measure [20] compares the clustering results with true class labels or another clustering to evaluate the effectiveness of the clustering technique. Unlike ARI, which focuses on pairwise comparisons, V-measure provides insight into cluster quality by balancing cluster purity and completeness. The V-measure is derived from the harmonic mean of homogeneity (w), which checks the purity of each cluster with respect to class labels, and completeness (y), which examines how well data points of the same class are contained within the same cluster. V-measure (A_{v-me}) is mathematically determined as,

$$A_{v-me} = (1+u)\frac{w \cdot y}{u \cdot w + y} \tag{14}$$

Here, the weight factor is exemplified as u. This ensures that neither homogeneity nor completeness dominates the score, leading to a fair evaluation of clustering quality.

4.2.3 Silhouette Score

The Silhouette score [20] evaluates clustering performance by measuring how well each point is grouped within its own cluster compared to other clusters. It essentially validates whether the cluster assignments are both compact (intra-cluster similarity) and well-separated (inter-cluster dissimilarity). The mathematical representation of the Silhouette score (A_{Sil}) is computed as,

$$A_{Sil} = \frac{s_h - c_h}{MAX(c_h, s_h)} \tag{15}$$

wherein, the average distance between the h^{th} data point and all other points within the same cluster is specified as c_h, and the average distance between the h^{th} data point and all points in the nearest neighboring cluster is signified as s_h. Scores close to +1 indicate strong clustering quality, values around 0 suggest overlapping clusters, and negative values imply misclassification.

4.3 Dataset Description

The work utilizes log data from the NSL-KDD dataset [15], an improved version of the original KDD dataset. This dataset is widely used in intrusion detection research, making results comparable with existing approaches. The test sets are free of duplicates, ensuring that the performance of the learners is not skewed by techniques that may perform better on more frequent data. By eliminating redundancy, the dataset challenges models to generalize better, reflecting real-world conditions where repeated patterns are less predictable. The dataset's size of 55.87MB is ideal for conducting experiments without the need for random sampling. This consistency helps maintain reliable and comparable evaluation results across different studies. The dataset includes various attributes such as duration, destination bytes (dst_bytes), service, protocol type, urgency (urgent), wrong fragments (wrong_fragment), hot, flag, land, source bytes (Src_bytes), logged-in status, and others. These features collectively capture both network-level and application-level behaviors, making the dataset suitable for testing multi-faceted detection approaches such as Ada-HDBSCAN.

4.4 Experimental Parameters

Table 1 illustrates the experimental parameters of the Ada-HDBSCAN.

Table 1. *Experimental parameters of Ada-HDBSCAN*

Parameters	Values
Number of clusters	6
Metric	Euclidean
Cluster_selection_epsilon(Threshold)	0.3
P	2.0
Cluster_selection_method	'eom'
Alpha	1.0
Core_dist_n_jobs	-1

4.5 Assessment in Terms of Ada-HDBSCAN

This section presents an evaluation of the proposed Ada-HDBSCAN method for DDoS attack detection, focusing on scalability and ablation studies. The goal is to validate whether the adaptive modifications introduced in Ada-HDBSCAN translate into consistent improvements across different clustering scenarios.

4.5.1 Scalability Analysis

The effectiveness of the Ada-HDBSCAN method for DDoS attack detection is assessed and compared with existing models across varying numbers of clusters. Scalability analysis is particularly important because real-world DDoS traffic often exhibits dynamic variations in density and distribution, requiring detection models to remain effective under diverse conditions. Fig 2 presents the evaluation of the Ada-HDBSCAN method for DDoS attack detection, with varying the number of clusters. Ada-HDBSCAN is compared against other existing techniques, including LGBM [10], DCVAE and DC-SVDD-VAE [11], DNCAE and DCHAE [13], and CBWSN_VSEMLA [14]. These models were selected as baselines because they represent a mix of traditional machine learning, deep learning, and clustering-based approaches, providing a fair comparison. The analysis of the Ada-HDBSCAN model with respect to ARI is demonstrated in Fig 2 a). For the number of cluster as 6, the existing models, such as LGBM, DCVAE and DC-SVDD-VAE, DNCAE and DCHAE, and CBWSN_VSEMLA attained the ARI of 0.878, 0.919, 0.928, and 0.938; meanwhile, the developed Ada-HDBSCAN gained the ARI of 0.966. his significant improvement reflects the ability of Ada-HDBSCAN to maintain strong consistency between predicted clusters and actual traffic patterns, even in complex network environments. Here, the developed Ada-HDBSCAN outperforms the CBWSN_VSEMLA model, achieving a 2.87% higher ARI. In Fig 2 b), the evaluation of the Ada-HDBSCAN technique on the basis of V-measure is shown. The introduced Ada-HDBSCAN achieved the V-measure of 0.938 for the number of cluster as 5. The prevailing models, like LGBM, DCVAE and DC-SVDD-VAE, DNCAE and DCHAE, and CBWSN_VSEMLA, recorded the V-measure of 0.848, 0.878, 0.900, and 0.920. Ada-HDBSCAN demonstrates superior performance, exceeding the DNCAE and DCHAE models' V-measure by 4.05%. This demonstrates that the adaptive clustering strategy prevents both underfitting (fragmentation) and overfitting (over-merging), striking an effective balance. Fig 2 c) depicts the assessment of the Ada-HDBSCAN approach in terms of the Silhouette score. The existing approaches, including LGBM, DCVAE and DC-SVDD-VAE, DNCAE and DCHAE, and CBWSN_VSEMLA, attained the Silhouette score of 0.877, 0.889, 0.910, and 0.919 for the number of cluster as 4, whereas the devised Ada-HDBSCAN technique obtained the Silhouette score of 0.948. A higher Silhouette score confirms that Ada-HDBSCAN produces clusters that are compact and well-separated, meaning it effectively distinguishes between legitimate and malicious traffic flows. The Silhouette score of Ada-HDBSCAN exceeds that of the DCVAE and DC-SVDD-VAE model by 6.24%, highlighting its superior performance.

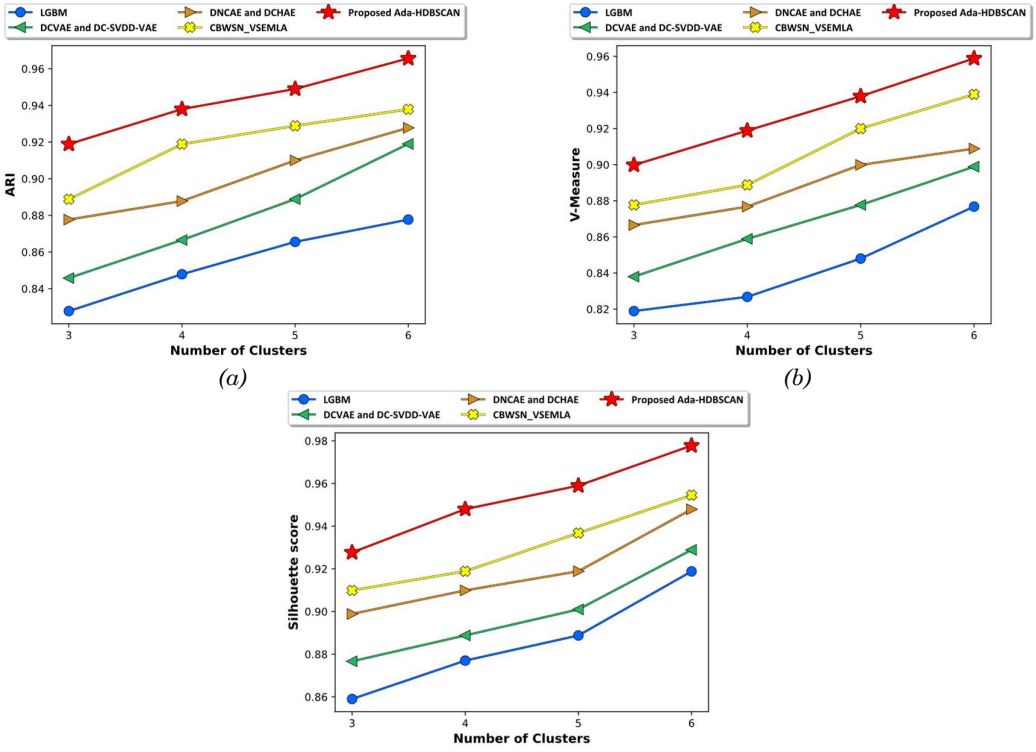


Fig 2. *Examination of the designed Ada-HDBSCAN model based on a) ARI, b) V-measure, and c) Silhouette score*

4.5.2 Ablation Analysis

The proposed Ada-HDBSCAN model's effectiveness is validated by comparing it to several scenarios, such as Input data (Rawdata)+Ada-HDBSCAN (Without feature selection), feature selection (GIWRF)+Ada-HDBSCAN (without SFS), and feature selection (SFS)+Ada-HDBSCAN (without GIWRF). This step-by-step breakdown highlights the contribution of each component in the pipeline, allowing us to isolate the benefits of ensemble feature selection and adaptive clustering. In Fig 3a, the analysis of Ada-HDBSCAN concerning ARI is illustrated. The Rawdata+Ada-HDBSCAN, GIWRF+Ada-HDBSCAN, and SFS+Ada-HDBSCAN, attained the ARI of 0.917, 0.928, and 0.946, for a number of cluster as 6. These results show that even without feature selection, Ada-HDBSCAN performs reasonably well; however, the addition of feature selection layers consistently boosts clustering accuracy. The Rawdata+GIWRF+SFS+Ada-HDBSCAN gained the ARI of 0.966. The evaluation of Ada-HDBSCAN based on V-measure is shown in Fig 3 b). The Rawdata+GIWRF+SFS+Ada-HDBSCAN achieved the V-measure of 0.938, for a number of cluster as 5, meanwhile Rawdata+Ada-HDBSCAN, GIWRF+Ada-HDBSCAN, and SFS+Ada-HDBSCAN obtained the V-measure of 0.889, 0.899, and 0.920, respectively. This indicates that GIWRF alone improves homogeneity by selecting discriminative features, while SFS enhances completeness by iteratively refining feature subsets. Fig 3 c) depicts the assessment of Ada-HDBSCAN on the basis of the Silhouette score. By assuming the number of clusters to be 4, the Rawdata+Ada-HDBSCAN, GIWRF+Ada-HDBSCAN, and SFS+Ada-HDBSCAN attained the Silhouette score of 0.890, 0.910, and 0.918, whereas the Rawdata+GIWRF+SFS+Ada-HDBSCAN gained the Silhouette score of 0.948. The consistent improvement across all metrics highlights the importance of using both GIWRF and SFS together, as they provide a balanced feature set that reduces noise, strengthens cluster cohesion, and improves separation between legitimate and attack traffic.

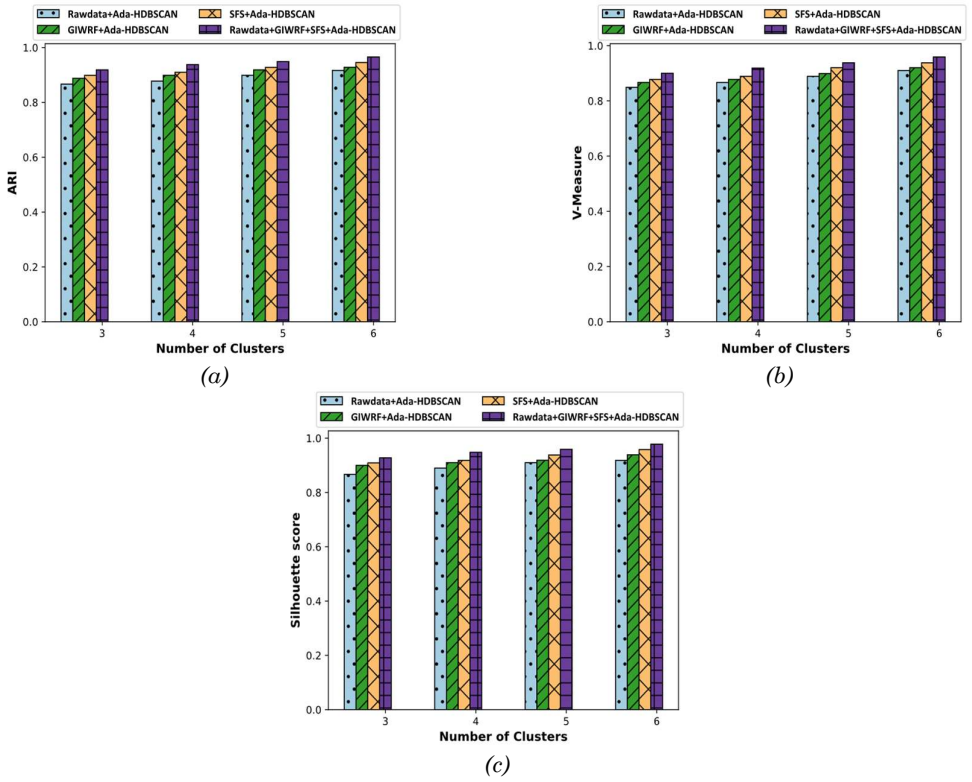


Fig 3. *Ablation analysis of the introduced Ada-HDBSCAN model in terms of a) ARI, b) V-measure, and c) Silhouette score*

4.6 Summary

This section presents a summary of Ada-HDBSCAN for detecting DDoS attacks, focusing on the number of clusters. Summarizing across multiple performance measures helps to provide a holistic validation of the model, ensuring that improvements are not isolated to a single metric. Table 2 portrays the discussion of the developed Ada-HDBSCAN for detecting DDoS attacks. The developed Ada-HDBSCAN achieved an ARI of 0.966, a V-measure of 0.959, and a Silhouette score of 0.978 for 6 clusters. Prevailing approaches, like LGBM, DCVAE and DC-SVDD-VAE, DNCAE and DCHAE, and CBWSN_VSEMLA attained the ARI of 0.878, 0.919, 0.928, and 0.938. Although these models perform reasonably well, their reliance on static assumptions or limited feature representation makes them less adaptable to evolving traffic distributions. The V-measure obtained by existing approaches, such as DCVAE and DC-SVDD-VAE, LGBM, CBWSN_VSEMLA, and DNCAE and DCHAE, is 0.877, 0.899, 0.909, and 0.939. Methods, like LGBM, DCVAE and DC-SVDD-VAE, DNCAE and DCHAE, and CBWSN_VSEMLA recorded the Silhouette score of 0.919, 0.929, 0.948, and 0.955. Ada-HDBSCAN improves DDoS attack detection by recognizing unusual traffic patterns, managing noisy data, and adapting to evolving network conditions. It detects anomalies effectively without requiring predefined cluster counts, and scales efficiently, making it suitable for advanced cybersecurity systems. This adaptability positions Ada-HDBSCAN as a practical solution for real-world deployment in cloud computing and IoT-based environments, where attack patterns are unpredictable and high-volume data streams must be analyzed in real-time.

Table 2. *Discussion table of the devised Ada-HDBSCAN*

Variations	Metrics	LGBM	DCVAE and DC-SVDD-VAE	DNCAE and DCHAE	CBWSN_ VSEMLA	Proposed Ada-HDBSCAN
Number of clusters 6	ARI	0.878	0.919	0.928	0.938	0.966
	V-measure	0.877	0.899	0.909	0.939	0.959
	Silhouette score	0.919	0.929	0.948	0.955	0.978

5. Conclusion

A DDoS attack overwhelms a target server or network with excessive traffic from multiple sources, resulting in disruptions or downtime. Such attacks not only degrade service availability but also impose severe financial and reputational damage, making their early detection an urgent necessity in

cybersecurity. Existing methods face significant challenges in detecting such attacks, including difficulties in distinguishing between legitimate and malicious traffic, overfitting, poor generalization, high false positive rates, and limited adaptability to evolving attack strategies. These limitations highlight the need for models that can remain robust under dynamic traffic conditions while maintaining low error rates. This research proposes a novel approach for detecting DDoS attacks through clustering, specifically using Ada-HDBSCAN. The approach begins by simulating a cloud model, and then the input log data is collected from the NSL-KDD dataset. Feature selection is later performed using the GIWRF and SFS techniques to identify the most relevant features from the input data. This two-step feature selection process helps in filtering out redundant attributes while preserving those most indicative of malicious behavior. Following this, the Ada-HDBSCAN algorithm is used for clustering, classifying the data as either DDoS-affected or normal. Ada-HDBSCAN is an innovative method that utilizes adaptive techniques for modifying the clustering parameters of the HDBSCAN algorithm. The Ada-HDBSCAN method achieved an ARI of 0.966, a V-measure of 0.959, and a Silhouette score of 0.978. Future work will focus on developing optimization models for dynamic threat analysis, enhancing feature selection techniques, and optimizing clustering algorithms to better adapt to emerging DDoS attack patterns.

Compliance With Ethical Standards

Conflicts of interest: Authors declared that they have no conflict of interest.

Human participants: The conducted research follows ethical standards and the authors ensured that they have not conducted any studies with human participants or animals.

Reference

[1] A. A. Soofi, M. I. Khan and F.-e.-. Amin, “A review on data security in cloud computing,” International Journal of Computer Applications, vol. 96, no. 2, pp. 95-96, 2017.

[2] Z. Du, W. Jiang, C. Tian, X. Rong and Y. She, “Blockchain-based authentication protocol design from a cloud computing perspective,” Electronics, vol. 12, no. 9, p. 2140, 2023.

[3] M. Ali, S. U. Khan and A. V. Vasilakos, “Security in cloud computing: Opportunities and challenges,” Information sciences, vol. 305, pp. 357-383, 2015.

[4] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu and M. Dlodlo, “Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing,” EURASIP Journal on Wireless Communications and Networking, vol. 2016, no. 1, p. 130, 2016.

[5] W. WANG, X. DU and N. WANG, “Building a cloud IDS using an efficient feature selection method and SVM,” IEEE Access, vol. 7, pp. 1345-1354, 2018.

[6] M. Aamir, S. Mustafa and A. Zaidi, “Clustering based semi-supervised machine learning for DDoS attack classification,” Journal of King Saud University-Computer and Information Sciences, vol. 33, no. 4, pp. 436-446, 2021.

[7] W. Bhaya and M. EbadyManaa, “DDoS attack detection approach using an efficient cluster analysis in large data scale,” in Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), 2017.

[8] N. S. Kharbanda, “Comparative Review of Supervised vs. Unsupervised Learning in Cloud Security Applications,” 2024.

[9] K. Golalipour, E. Akbari, S. S. Hamidi, M. Lee and R. Enayatifar, “From clustering to clustering ensemble selection: A review,” Engineering Applications of Artificial Intelligence, vol. 104, p. 104388, 2021.

[10] S. DASARI and R. KALURI, “An effective classification of DDoS attacks in a distributed network by adopting hierarchical machine learning and hyperparameters optimization techniques,” IEEE Access, vol. 12, pp. 10834-10845, 2024.

[11] V. Q. Nguyen, V. H. Nguyen, L. T. Ngo and L. M. Nguyen, “Variational Deep Clustering approaches for anomaly-based cyber-attack detection,” Journal of Network and Computer Applications, p. 104182, 2025.

[12] T.-. L. Nguyen, H. Kao, T.-. T. Nguyen, M.-. F. Horng and C.-. S. Shieh, “Unknown DDoS Attack Detection with Fuzzy C-Means Clustering and Spatial Location Constraint Prototype Loss,” Computers, Materials & Continua, vol. 78, no. 2, 2024.

[13] V. Q. Nguyen, L. T. Ngo, L. M. Nguyen, V. H. Nguyen and N. Shone, “Deep clustering hierarchical latent representation for anomaly-based cyber-attack detection,” Knowledge-Based Systems, vol. 301, p. 112366, 2024.

[14] A. John, I. F. B. Isnin, S. H. H. Madni and M. Faheem, “Cluster-based wireless sensor network framework for denial-of-service attack detection based on variable selection ensemble machine learning algorithms,” Intelligent Systems with Applications, vol. 22, p. 200381, 2024.

[15] "NSL-KDD dataset," 2025. [Online]. Available: https://www.kaggle.com/datasets/hassan06/nslkdd. [Accessed August 2025].

[16] R. A. Disha and S. Waheed, "Performance analysis of machine learning models for intrusion detection system using Gini Impurity-based Weighted Random Forest (GIWRF) feature selection technique," Cybersecurity, vol. 5, no. 1, p. 1, 2022.

[17] S. Bashir, I. U. Khattak, A. Khan, F. H. Khan, A. Gani and M. Shiraz, "A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches," Complexity, vol. 2022, no. 1, p. 8190814, 2022.

[18] X. Wu, D. Wang, M. Yang and C. Liang, "CEEMDAN-SE-HDBSCAN-VMD-TCN-BiGRU: A two-stage decomposition-based parallel model for multi-altitude ultra-short-term wind speed forecasting," Energy, p. 136660, 2025.

[19] H. Sheng, Z. Huang, L. Ke, J. Zhang, Z. Zeng, M. Yasir and S. Liu, "Multi-scale vessel trajectory clustering: An adaptive DBSCAN method for maritime areas of diverse extents," Ocean Engineering, vol. 334, p. 121461, 2025.

[20] F. J. Abdullayeva, "Distributed denial of service attack detection in E-government cloud via data clustering," Array, vol. 15, p. 100229, 2022.