International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Adaptive Intrusion Detection in Cloud Systems using Contextual Deep Clustering Approach with Rule-Driven Feature Selection and Real-Time Monitoring

Parepalli Nageswara Rao
Research Scholar,
Osmania University, Assistant Professor
Neil Gogte Institute of Technology, OU
nagcsengit@gmail.com

K. Radhika
Professor
Chaitanya Bharati Institute of Technology
IT Department, Hyderabad, India

Abstract: With the increasing reliance on cloud-based infrastructures and the rising complexity of cyber threats, conventional intrusion detection systems (IDS) often fall short due to their static architectures and reliance on predefined attack signatures. This study introduces a novel contextual deep clustering-based intrusion detection framework that leverages advanced similarity measures and feature reduction techniques to enhance the accuracy and efficiency of threat detection in dynamic network environments. The proposed method integrates three traditional clustering algorithms—K-Means, Shift Mean, and Hierarchical Clustering—to form a foundational understanding of data patterns, which are further optimized using a deep clustering algorithm called DSE-CF-CI (Deep Similarity Extraction for Cluster Formation using Contextual Information). This approach enables the identification of both known and novel attack vectors through personalized centroid computation and context-aware rule generation. Unlike conventional methods, which often rely on rigid distance metrics and pre-defined parameters, the proposed method dynamically adapts to evolving data structures, thereby improving detection accuracy and reducing false positives. To validate its effectiveness, the model is tested using the KDD CUP 99 dataset, a widely accepted benchmark in intrusion detection research. Multiple features such as service type, protocol type, guest login status, and host connection behaviors are analyzed through cluster analysis. The results demonstrate significant improvements in distinguishing between normal and anomalous traffic patterns. Key performance metrics show a 91.2% overall detection accuracy, with selected clusters achieving up to 99% precision. Additionally, the proposed framework supports real-time monitoring within a scalable cloud deployment architecture, incorporating host-level behavior analysis and anomaly detection through live system metrics like CPU usage, disk activity, and network traffic. Comparative analysis with existing intrusion detection methods—such as SVM, K-Means, Hierarchical Clustering, Mean Shift, and ACO-based models—further illustrates the superiority of the proposed system in terms of both accuracy and efficiency. The integration of a knowledge-based feature subset selection (KBFSS) module enhances the overall model by reducing redundant attributes, speeding up detection time, and maintaining high detection fidelity. Ultimately, the proposed framework represents a significant advancement in the field of cybersecurity, offering an intelligent, adaptive, and high-performance solution for intrusion detection in complex, data-rich cloud environments. It also establishes a foundation for future research into adaptive rule-based systems that can autonomously evolve to meet the demands of increasingly sophisticated threats.

Keywords: Intrusion Detection, Deep Clustering, Cloud Security, Feature Selection, Contextual Similarity, Anomaly Detection, Real-Time Monitoring

1. INTRODUCTION

An intrusion attack refers to a scenario where one component within a system initiates actions that influence another component in a manner that may disrupt the overall system operations. These disruptive actions can originate from either internal users or external sources. Detecting and mitigating such intrusions is a critical measure in defending against cyberattacks. The comprehensive process of identifying these threats is known as intrusion

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

detection, and its practical implementation is typically carried out through Intrusion Detection Systems (IDS).

In this context, C. Guo et al. introduced a novel multi-level security model designed to act as a defensive mechanism against various forms of cyberattacks. These systems are generally implemented as modular software components, which can be integrated into existing infrastructures to enhance their security capabilities. Complementary research conducted by G.V. and Nadiammai et al. has further explored the features of such systems, emphasizing that an effective IDS should not only detect and block unauthorized access attempts but also trigger alerts when potential threats are identified.

Experimental studies have demonstrated that deploying a robust IDS can significantly improve system resilience. These systems can help restore normal operations after detecting malicious activities. In particular, research by P. Mishra et al. suggests that traditional access control and detection methods may no longer be sufficient, and emerging techniques are needed to address evolving threats. One such promising approach is deep clustering, which introduces advanced data grouping and pattern recognition capabilities that can enhance detection accuracy and system adaptability.

This section explores the general clustering techniques applied in intrusion detection, focusing on methods that categorize similar data patterns to identify anomalies. A critical analysis of these techniques is presented to highlight current limitations and evaluate the advantages of incorporating deep clustering strategies. These advanced methods hold significant potential for improving threat detection, enabling proactive responses to security risks in increasingly complex computing environments.

2. CLUSTERING BASED APPROACH

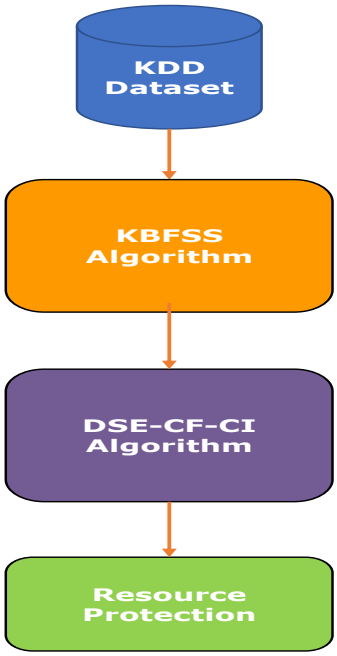


Figure 1 Clustering based Approach

2.1 Clustering Based on K-Means

The K-Means algorithm is one of the most commonly employed clustering techniques in data analysis and intrusion detection tasks. This method begins by randomly selecting initial centroids—referred to as cluster centers—based on the predefined number of clusters (*K*). The initial selection of these centroids plays a crucial role in influencing the final

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

grouping outcome. Therefore, analyzing the dataset beforehand is essential to increase the likelihood of choosing representative initial centroids.

Each data point and centroid is represented as a vector of the same dimensionality, corresponding to the number of features or attributes in the dataset. Let the complete dataset be denoted as DDD, where each data point is characterized by *n* attributes. This assumption enables the formal mathematical representation of the dataset as a collection of *n*-dimensional vectors:

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \quad \ldots (1)$$

Each data attribute and element within the dataset is associated with a specific domain and a defined range of possible values or entries. The data elements are typically represented using symbolic notation, which serves as a formal representation of the individual data items. Each data attribute and element within the dataset is associated with a specific domain and a defined range of possible values or entries. The data elements are typically represented using symbolic notation, which serves as a formal representation of the individual data items.

$$A_x[] = \bigcup_{i=1}^{m} D_i \quad \ldots (2)$$

K-Means is a clustering technique employed to form the predefined groupings within a dataset. Each cluster is associated with a centroid, typically denoted by the symbol CCC, which represents the center of that cluster. When the desired number of clusters is specified, the algorithm assigns data points to the nearest centroid, effectively partitioning the dataset. This relationship between data points and centroids can be mathematically characterized to guide the clustering process.

$$C[] = \bar{\bar{\lambda}}_k[] = \frac{\bar{\lambda}_k[]}{\left|\lambda_i - \lambda_{i+1}\right|_{i=0}^{n}} \quad \ldots (3)$$

It is necessary to do an analysis on each of the dataset's domains in order to determine the centroid, which stands for the data element's representation.

$$\phi_X = \frac{\sum_{i=1}^{m} D_i}{\Delta D_i} \quad \ldots (4)$$

Therefore, for each of the characteristics included in the dataset, it is necessary to do an analysis of the total centroids because,

$$\bar{\phi} = \phi_1 \cup \phi_2 \cup ..... \cup \phi_n \quad (5)$$

The data elements within the dataset must be assigned to clusters in a manner consistent with their original representation. The K-Means algorithm considers the multidimensional nature of each data point—represented as feature vectors—when determining the appropriate cluster. These vectors are used to accurately compute the cluster centroids, ensuring that each data point is grouped with others exhibiting similar characteristics.

$$\lambda_i[1...k] \leftarrow \bar{\phi}[1...n] \quad (6)$$

The following phases of the K-Means method are described in more detail here:

Algorithm 1 : Traditional K – Means Algorithm

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Step 1: Determine which classes will utilise the system and describe the specific objectives for each class.

Step 2: Organize every datum stage by documenting the distance between there and each accumulating focus.

Step 3: Organize every datum stage by documenting the distance between there and each accumulating focus.Step 3: Define the aim of sustaining the accumulation at its present place, which is the most convenient one for this purpose.

Step 4: Update the accumulating focus by picking just the mean of a substantial number of vectors extracted from the accumulating focus.

Step 5: Repeat Measures 1 through 4 prior to going on to the next step, until all of the data points have been saved inside each group.

2.2 Clustering Technique with Shift Mean

The **Shift Mean Clustering technique** represents a conventional yet effective approach to data clustering. Unlike traditional methods that require predefining the number of clusters, this technique dynamically identifies cluster centroids by analyzing the mean variations across all data points within the dataset. The algorithm iteratively updates the cluster centers based on local mean shifts, allowing clusters to emerge naturally from the data distribution. One of the key advantages of this method is that it **does not require prior knowledge of the number of clusters or their initial centroids**, making it particularly useful in exploratory data analysis or in situations where the structure of the data is unknown.

This section presents the mathematical foundation of the Shift Mean Clustering technique. To initiate the clustering process, the total number of features (or attributes) in the dataset must be determined. The dataset is denoted by D, where each attribute contains a total of n instances. This can be formally expressed as:

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \quad \ldots (7)$$

It is generally accepted that each attribute belongs to its own domain, and the data pieces are labelled as such. This technique takes into consideration the various records that each characteristic has as well.

$$A_x[] = \sum_{i=1}^{m} D_i \qquad \ldots (8)$$

In the event that the first data point is, it is necessary to compute the distance between that location and the other data points.

$$\int_{i=1}^{n} D_x - \int_{i=1}^{n} |D_x - D_i| \Rightarrow \lambda[] \qquad \ldots (9)$$

Where, $\lambda[]$ is the total distance set. During the clustering process, the total distance set is the most critical parameter that may be changed. It decides which cluster should be made up of all of the data points that come from that one place on the graph..

$$C[] \leftarrow \frac{\sum_{j=1}^{n} \lambda_j}{\Delta |\lambda_j - \lambda_{j+1}|} \quad \ldots (10)$$

The following are the stages that are currently included in the Mean Shift algorithm:

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Algorithm 2: Traditional Mean Shift Algorithm
Step - 1. Start off with a circular sliding window cantered in some stage C Step 2: The sliding window has been adjusted such that it now faces locations with a higher population density. This was accomplished by shifting the focal point to the expression of the objects that are housed inside the window. Step 3: Continue making adjustments to the window that is sliding. On the basis of this, until there is no direction where a change may adapt additional things, there won't be any progress. Step 4: Duplicate Measure Step-1 Step-3 before all data points have been stored within practically any group

2.3 Hierarchical Clustering Method

Hierarchical clustering is another traditional and widely used method for data grouping. Its ability to adaptively adjust to the underlying data structure—without requiring a predefined number of clusters—gives it a notable advantage over earlier clustering techniques.

This approach builds a hierarchy of clusters by either progressively merging smaller clusters into larger ones (*agglomerative*) or by dividing a large cluster into smaller subsets (*divisive*). It is especially useful for uncovering nested structures within data and for visualizing relationships through dendrograms.

Hierarchical clustering operates under the assumption that the entire dataset is represented by the symbol D, and that each feature or attribute is expressed as a vector consisting of n numerical values:

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \quad \ldots (11)$$

The many qualities that make up a dataset are each given their own domain, and the data pieces that make up the dataset are labelled accordingly. This technique may also be used to represent the individual data pieces that make up a list.

$$A_x[] = \sum_{i=1}^{m} D_i \qquad \ldots (12)$$

If the first data point is a part of the cluster, then the remaining clusters should be designed according to the clustering method., C_X. As,

$$D_x \rightarrow C_X \qquad \ldots (13)$$

$$Iff \int_{i=1}^{n} |D_x - D_i| < \int_{i=1}^{n} |D_x - D_j|$$

$$Then\ D_i \rightarrow C_x \qquad \ldots (14)$$

$$Else, D_j \rightarrow C_x\ and\ D_i \rightarrow C_{x+1}$$

The expanded current hierarchical steps:

Algorithm 3: Hierarchical Traditional Method
Step-1: Begin by addressing each datum stage as an individual group. Step-2: Combine the first and second courses into a single group. Step-3: Whilst the bunches to be joined are identified as folks having all the smallest connection Step-4: It is necessary to repeat Measure Step 1 to 2 until virtually all of the data points have been stored within any bunch.

Conventional clustering techniques often face limitations due to their rigid distance metrics and inflexible assumptions about data distribution. These constraints reduce their

effectiveness in complex environments, such as intrusion detection systems (IDS), where data patterns are often non-linear and overlapping. As a result, there is an increasing demand for innovative and adaptive clustering approaches capable of addressing these shortcomings.

In response to this need, the present study introduces a deep clustering-based framework specifically designed for intrusion detection. This proposed method leverages the strengths of deep learning to extract high-level feature representations, thereby enhancing clustering accuracy and robustness. Compared to traditional techniques, deep clustering offers several distinct advantages.

One of its key benefits is its ability to minimize overlap between intrusion threat classes, which significantly improves detection precision. Moreover, it enables the system to identify high-risk categories, allowing for targeted and prioritized responses. Another critical advantage is its capacity to reveal structural similarities between different types of intrusion attacks, including subtle patterns that may not be evident using conventional methods.

This section proceeds by examining commonly used clustering techniques and their limitations, followed by a detailed discussion of the core characteristics and functional requirements of modern intrusion detection systems. The goal is to establish a clear rationale for adopting deep clustering as a superior alternative in the field of cybersecurity analytics.

3 PROBLEM FORMULATION

In this section, the primary objective is to identify the most critical challenge that must be addressed to improve the overall effectiveness of the proposed work. The subsequent stages of the project are undertaken after the necessary data and contextual information have been thoroughly gathered and analyzed.

As a foundational step, it is essential to establish a mathematical relationship between the total number of attributes in the dataset and the dataset itself, denoted by DDD. This relationship serves as the basis for all subsequent analytical and computational procedures. For the purposes of this study, it is assumed that the dataset is represented by the symbol DDD, with each data instance comprising a defined set of attributes.

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \quad \dots (15)$$

It is possible to think of each of the attributes in the dataset as having its own domain, and this domain is represented by the number of entries the attribute has. In addition to that, data items are denoted with.

$$A_x[] = \sum_{i=1}^{m} D_i \quad \dots (16)$$

The beginning point of the collection is often understood to be the first data point that is included in a dataset. In addition to these two data points, the dataset contains a further two pieces of information. X and Y represent the gap that exists between these two data points.

$$\left|D_x - D_y\right| \Rightarrow \lambda_1 \quad \dots (17)$$

And,

$$\left|D_y - D_z\right| \Rightarrow \lambda_2 \quad \dots (18)$$

And,

$$\left|D_x - D_z\right| \Rightarrow \lambda_3 \quad \dots (19)$$

In the situations, where

$$\lambda_1 < \lambda_2 \quad \dots (20)$$

And,

$$\lambda_3 > \lambda_1 \qquad \dots (21)$$

And,

$$\lambda_3 < \lambda_2 \qquad \dots (22)$$

The clustering algorithms employed in the previous compilation were limited in their ability to accurately distinguish between core groups and outliers within the dataset. This shortcoming presents a critical issue that needs to be addressed in order to enhance the precision and reliability of clustering outcomes. To overcome this limitation, a more robust and adaptive clustering mechanism is required.

The following section introduces and discusses a novel system specifically designed to improve the clustering of both well-defined groups and anomalous outliers. This system aims to refine group formation while effectively isolating outlier instances, thereby contributing to a more accurate and insightful data analysis process.

4 PROPOSED METHOD: MATHEMATICAL FOUNDATION

This section presents a thorough investigation of the various aspects of the clustering process. It includes a comprehensive analysis of the limitations that impact the performance and accuracy of clustering algorithms. Understanding these constraints is essential for optimizing the effectiveness of clustering techniques in practical applications.

For the purpose of this study, the entire dataset is denoted by DDD, where each data instance is characterized by a set of nnn attributes. The clustering process involves examining the relationships among these attributes to identify meaningful groupings. By extending or fine-tuning the clustering parameters, it is possible to establish more accurate associations between data points, ultimately leading to improved cluster formation and anomaly detection.

$$D[] \rightarrow < A_1, A_2, A_3, ...A_n > \qquad \dots (23)$$

Each attribute and data element belongs to its own domain, which has a predetermined number of records, and attributes are labelled with while data items are denoted by.

$$A_x[] = \sum_{i=1}^{m} D_i \qquad \dots (24)$$

The similarity metric that is used is the distance that exists between each pair of data points. The whole distance covered is also shown by the symbol.

$$\lambda[] = \int_{i=1}^{n} |D_i - D_{i+1}| \qquad \dots (24)$$

It is possible to compute the distance between each element and the data points using the formula.

$$\bar{\lambda}[] = \int_{i=1}^{n-1} |\lambda_i - \lambda_{i+1}| \qquad \dots (25)$$

The similarities between two things $\bar{\lambda}[]$ help describe the relationship that exists between an element and its data points. The iterative process of solving Equation 26 may also be used to assess the degree to which the contextual and deeper characteristics are comparable.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

$$\bar{\lambda}_k[] = \int_{i=1}^{n-k} \left| \bar{\lambda}_i - \bar{\lambda}_{i+1} \right| \quad \dots (26)$$

When calculating the cluster centroids, it is necessary to take into consideration both the similarities between the elements and the data points as well as the distance that separates them.

$$C[] = \bar{\bar{\lambda}}_k[] = \frac{\bar{\lambda}_k[]}{\left| \lambda_i - \lambda_{i+1} \right|_{i=0}^{n}} \quad \dots (27)$$

Henceforth, in the next section of the work, the proposed algorithm is elaborated.

5 CONTEXTUAL DEEP CLUSTERING

A **similarity function**, also referred to as a **similarity measure**, is a mathematical tool used to evaluate the degree of resemblance between two entities. Although there is no universally accepted definition, similarity measures are generally considered to be the conceptual inverse of distance (or dissimilarity) measures. While distance measures emphasize the magnitude of difference between two items—often assigning higher values to dissimilar pairs—similarity measures highlight shared characteristics and assign higher scores to items that are more alike.

One widely used example of a similarity measure is the **Cosine Similarity**, which assesses the angular distance between two vectors and is particularly useful in information retrieval and text mining. It quantifies how similar two vectors are, regardless of their magnitude, by measuring the cosine of the angle between them.

In artificial intelligence and machine learning, functions like the **Radial Basis Function (RBF) kernel** are also considered forms of similarity measures, especially in kernel-based learning algorithms. These functions enable the transformation of input data into a higher-dimensional space where patterns become more distinguishable.

This section outlines the proposed approach, detailing its constituent stages and the role of similarity measures in enhancing clustering accuracy and interpretability.

Algorithm 4: Deep Similarity Extraction for Cluster Formation using Contextual Information Algorithm **(DSE-CF-CI)**
Step - 1. Let the primary dataset. D[] ={ A1,A2,...An}
Step - 2. For each attribute set, A[i]
Step - 3. For every single data point that falls inside the domain, D[i]
Step - 4. Determine the distances $\lambda[]$ to the equation. For every single distance
Step - 5. Determine the degree of similarity $\bar{\lambda}[]$ between the two sets of data using equation 3c. For each and every one of
Step - 6. Perform the calculations for the deep similarity measures in accordance with Equation 4.
Step - 7. Using Equation 5, determine the centres of each cluster.
Step - 8. Use Step 3 to begin forming the clusters, and continue doing so until all of the components have been included in the clusters.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

The proposed technique incorporates the contextual sensitivity of various features present within the dataset, allowing for the generation of individual-specific cluster centroid representations. By tailoring the clustering process to account for the nuanced influence of each variable, this strategy significantly enhances the adaptability and precision of the intrusion detection model.

Such a context-aware clustering framework is expected to improve classification accuracy by aligning the centroid formation with the unique data patterns associated with each subject or instance. This personalized centroid generation facilitates more accurate threat detection and reduces false positives.

The subsequent section presents the experimental results obtained from the implementation of the proposed methodology. The research highlights the effectiveness and relevance of the approach in comparison to traditional clustering methods, reinforcing its potential in real-world intrusion detection scenarios.

4.6 PROPOSED IDS BASED CLOUD DEPLOYMENT MODEL

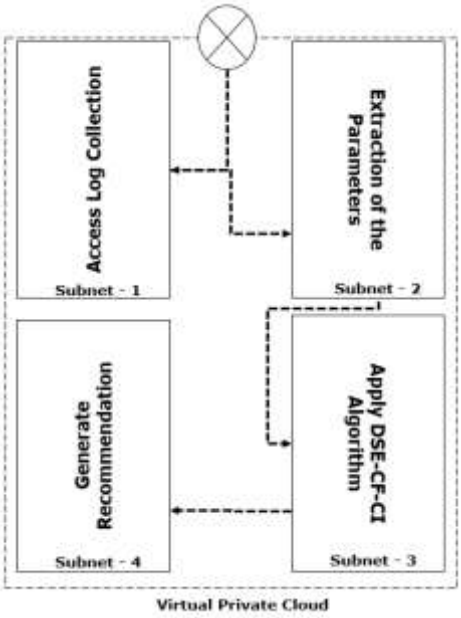


Figure 2: Proposed Deployment Model

The proposed algorithm is deployed within a **multi-layered cloud architecture**, comprising various interconnected components designed for real-time monitoring and threat detection. A **host-based Intrusion Detection System (IDS)** is integrated into the architecture to continuously monitor the dynamic behavior of the system at the host level.

In addition to tracking general system activities, the algorithm is capable of identifying **active processes** and **the specific assets accessed by those processes**. This fine-grained visibility enables the detection of abnormal or unauthorized behavior with greater precision. For example, the system can detect unauthorized modifications to sensitive resources—such as changes made to a **password database** by applications like word processors—indicating potential intrusion or malicious activity.

This implementation supports proactive intrusion detection and strengthens overall security posture by enabling context-aware monitoring within a scalable cloud environment. In practical scenarios, many users operate devices that, instead of explicitly displaying malware activity, exhibit **dynamic system behavior**—such as unusual resource usage or unauthorized access patterns. These behaviors often require deeper analysis to uncover hidden threats. While testing environments (or frameworks) are frequently employed to evaluate different

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

programs, doing so typically demands considerable effort to ensure that applications behave as intended and adhere to expected security protocols.

Additionally, the proposed framework supports **real-time, live monitoring**, allowing security analysts and automated systems to continuously observe and respond to suspicious activities as they unfold. This capability enhances situational awareness and provides a proactive defense mechanism against evolving cybersecurity threats.

Figure 3: IDS Instance Live Monitoring

Figure 3 presents the live monitoring output of the IDS instance, capturing key system performance metrics such as disk write operations, disk read operations, network activity, and CPU utilization during intrusion scenarios. These real-time indicators provide critical insights into the system's behavior under attack, enabling early detection and rapid response.

In the following section, the effectiveness of the proposed framework will be evaluated using multiple benchmark datasets and various cloud-based service environments. This evaluation aims to assess the adaptability, accuracy, and performance of the system across diverse intrusion patterns and deployment conditions.

7 RESULTS AND DISCUSSIONS

This section presents an analysis of the results derived from various clustering experiments, with a particular focus on the numerical evaluation of the class-type variable in the KDD dataset. The proposed method incorporates this variable into the clustering process, and for clarity and interpretability, only a selected subset of variables is showcased in this study to provide a representative snapshot of system behavior. Special attention is given to the 'service type' attribute, which is analyzed in relation to the 'label link' variable. This enables the identification of distinct service patterns associated with normal and malicious activities. It is important to distinguish the role of Intrusion Detection Systems (IDS) in the broader context of cybersecurity. Unlike firewalls, which primarily function to restrict access by limiting the number of systems that can connect to a network, IDS focuses on detecting, analyzing, and responding to potential threats and intrusions that may bypass traditional perimeter defenses. A graphical representation of the observed results is provided in this section to visualize the impact of the clustering approach and highlight the effectiveness of the proposed method in identifying intrusion patterns.

Table 1: Cluster Analysis - service=link

Cluster Group (CG)	Proposed Method Euclidian Distance	Actual Euclidian Distance
CG_2	0.0	0.0

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

CG_5	0.0	0.0
CG_6	0.0	0.0
CG_7	0.0	0.0
CG_8	0.0	0.0
CG_3	0.14	0.0
CG_9	0.18	0.0
CG_4	0.0	0.14
CG_0	0.0126	0.15
Main	0.15	0.18
CG_1	0.0	0.0126

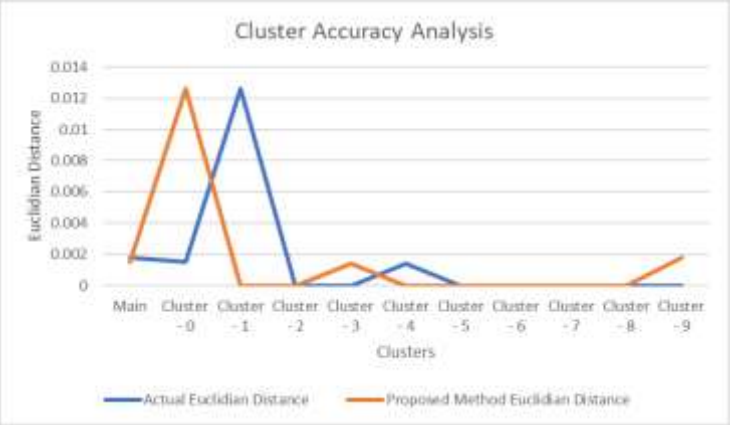


Figure 4: Cluster Analysis – Service = LINK

Secondly, the service type attribute for the label X11 is elaborated here

Table 2: Cluster Analysis X11

Cluster	Actual Euclidian Distance	Proposed Method Euclidian Distance
CG_2	0.	0.
CG_5	0.	0.
CG_6	0.	0.
CG_7	0.	0.
CG_8	0.	0.
CG_9	0.07	0.
CG_0	0.36	0.
CG_3	0.56	0.
Main	0.	0.07
CG_1	0.	0.36
CG_4	0.	0.56

The results are visualized graphically here.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

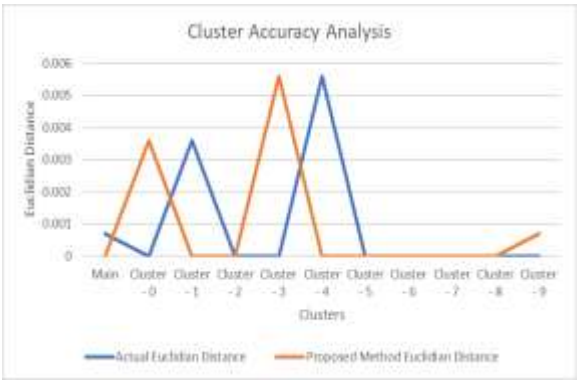


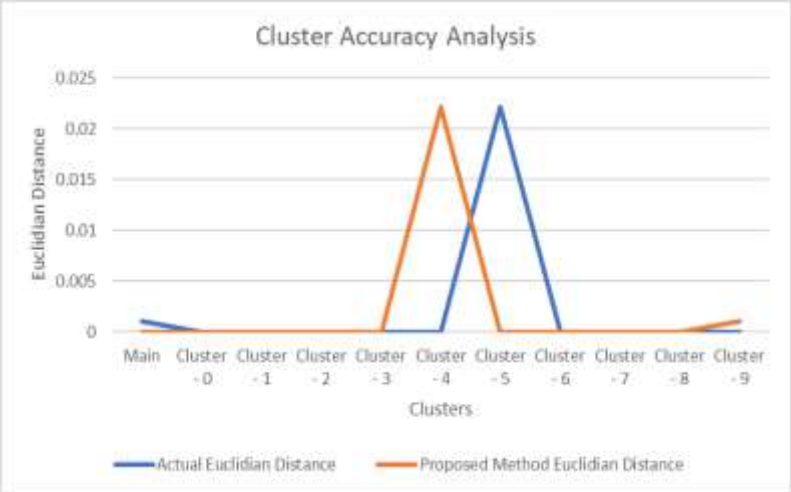
Figure 5: Cluster Analysis – Service = X11

Third, the service type attribute for the label urp_i is elaborated here.

Table 3: Cluster Analysis - service= urp_i

Cluster	Actual Euclidian Distance	Proposed Method Euclidian Distance
CG_2	0.10	0.
CG_5	0.	0.
CG_6	0.	0.
CG_7	0.	0.
CG_8	0.	0.
CG_9	0.	0.0221
CG_0	0.0221	0.
CG_3	0.	0.
Main	0.	0.
CG_1	0.	0.
CG_4	0.	0.10

This paper objective is to examine the traffic that is emanating from the different subnets in an effort to spot any abnormal patterns of behaviour or intrusion attempts. After it has been determined that an assault has occurred, an alert may be sent to the secretary. On a particular network, an instance of this system will be put up so that it can keep an eye out for any telltale indicators that an intruder may be attempting to break in. The results are visualized graphically here.



International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Figure 6: Cluster Analysis – Service = URP_I

Fourth, the guest login attribute for the label "1" is elaborated here.

Table 4.4: Cluster Analysis – Guest_Login =1

Cluster	Actual Euclidian Distance	Proposed Method Euclidian Distance
CG_2	0.10	0.
CG_5	0.	0.
CG_6	0.	0.0012
CG_7	0.0012	0.
CG_8	0.	0.
CG_9	0.	0.0284
CG_0	0.2067	0.
CG_3	0.	0.
Main	0. 0284	0.
CG_1	0.02067	0.
CG_4	0.	0.10

Due to the rapid evolution of malicious software, the adoption of anomaly-based intrusion detection systems (IDS) has become increasingly prevalent for identifying previously unknown or zero-day attacks. This approach involves first establishing a baseline model of normal system behavior, which is then used as a reference to detect deviations that may indicate malicious activity.

In contrast to traditional signature-based IDS, which rely on known attack patterns, machine learning-based systems offer a more generalized and adaptive detection capability. These systems can learn from a wide range of behaviors and dynamically adjust to emerging threats, making them more effective in environments where attack patterns evolve rapidly.

The outcomes of applying this approach are presented through graphical visualizations in this section, offering a clear depiction of the model's performance and its ability to distinguish between normal and anomalous activities.

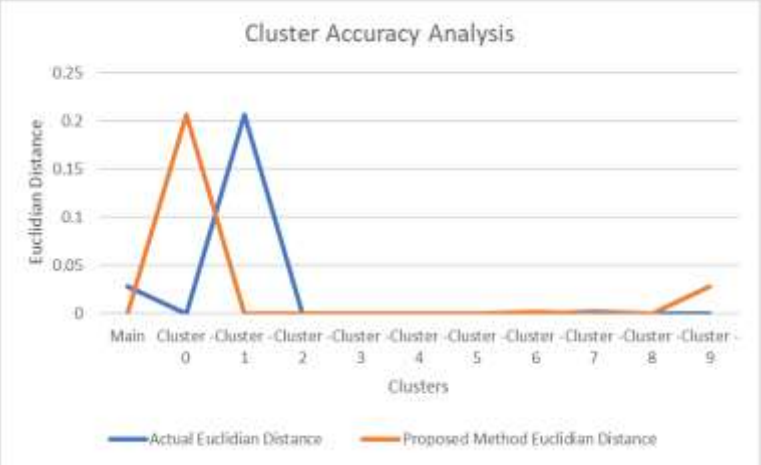


Figure 7: GUEST_LOGIN =1 Cluster Analysis

Table 5: Cluster Analysis – Protocol_Type=TCP

Cluster	Actual Euclidian Distance	Proposed Method Euclidian

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

		Distance
CG_1	1.0000	0.0000
CG_3	1.0000	1.0000
CG_4	1.0000	0.0000
CG_6	1.0000	1.0000
CG_7	1.0000	1.0000
CG_8	1.0000	1.0000
CG_9	1.0000	0.1625
Main	0.8375	0.8025
CG_0	0.8025	1.0000
CG_2	0.0000	1.0000
CG_5	0.0000	1.0000

The results are visualized graphically here.

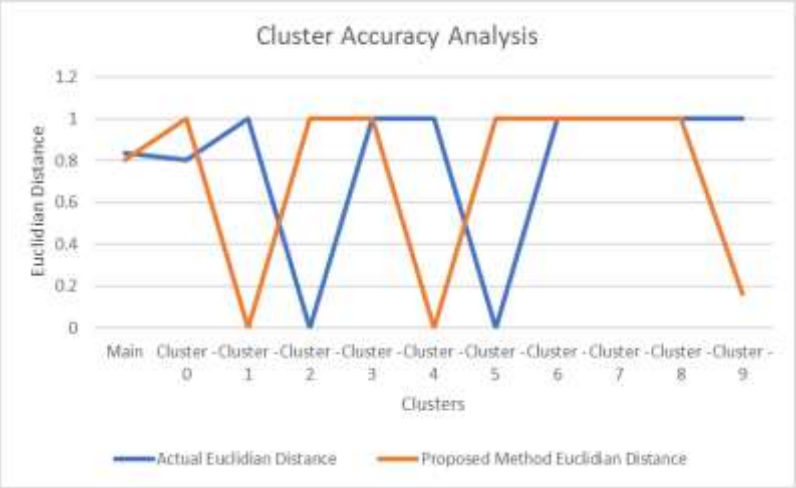


Figure 8: Cluster Analysis – Protocol_Type=TCP

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Sixth, the dst_host_diff_srv_rate attribute is elaborated here.

Table 6: Cluster Analysis – dst_host_diff_srv_rate

Cluster	Actual Euclidian Distance	Proposed Method Euclidian Distance
CG_0	0.9198	0.1017
CG_3	0.1463	0.0866
CG_7	0.1068	0.0353
CG_1	0.1017	0.0994
CG_2	0.0994	0.1463
Main	0.0905	0.9198
CG_4	0.0866	0.0305
CG_6	0.0783	0.1068
CG_8	0.0353	0.0051
CG_5	0.0305	0.0783
CG_9	0.0051	0.0854

The results are visualized graphically here.

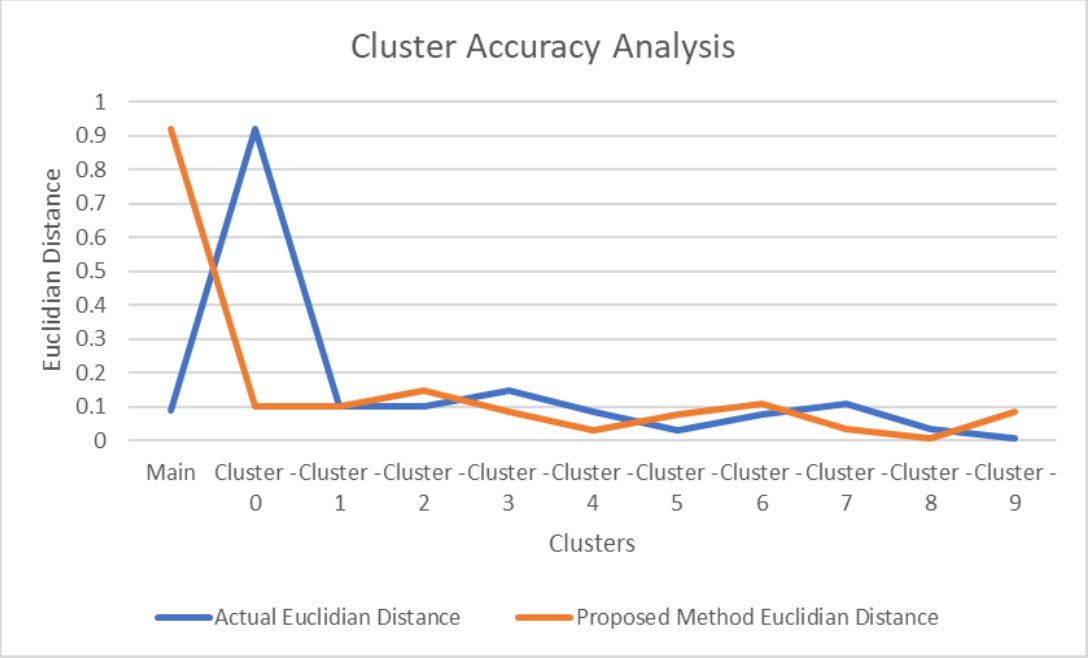


Figure 9: Cluster Analysis – DST_HOST_DIFF_SRV_RATE

Seventh, the service attribute for the label other is elaborated here.

Table 7: Cluster Analysis – SERVICE=OTHER

Cluster	Actual Euclidian Distance	Proposed Method Euclidian Distance
CG_0	0.9861	0.0456
CG_1	0.0456	0.0181
Main	0.0372	0.9861
CG_2	0.0181	0.0000

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

CG_4	0.0181	0.0000
CG_3	0.0000	0.0181
CG_5	0.0000	0.0000
CG_6	0.0000	0.0000
CG_7	0.0000	0.0000
CG_8	0.0000	0.0000
CG_9	0.0000	0.0372

The results are visualized graphically here.

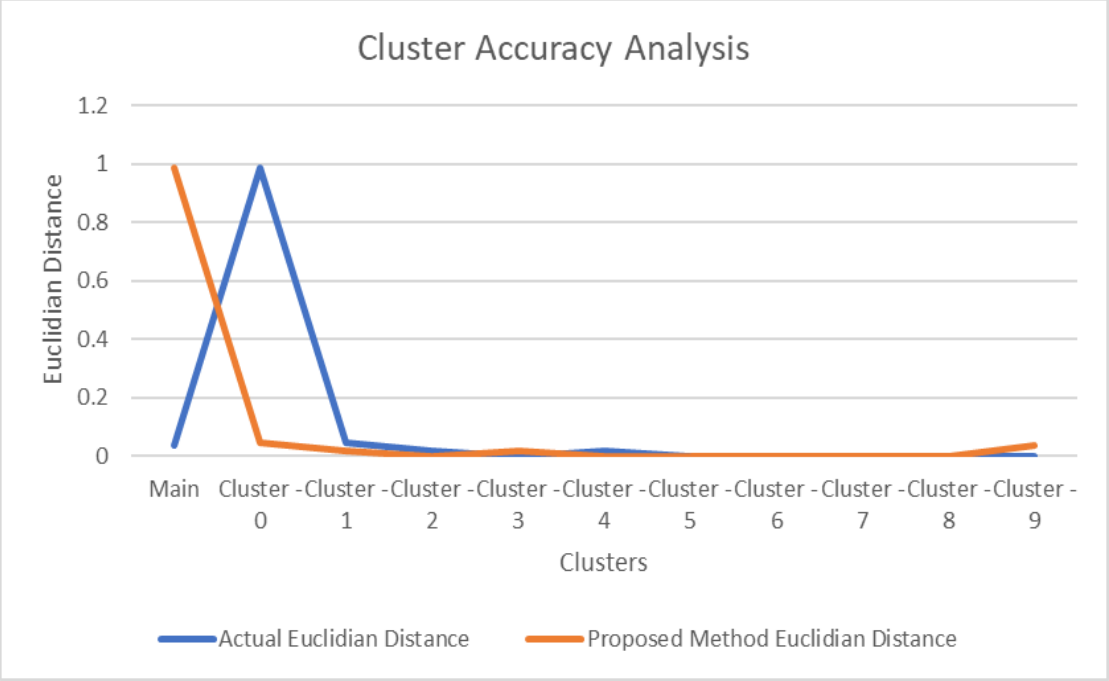


Figure 10: Cluster Analysis – SERVICE=OTHER

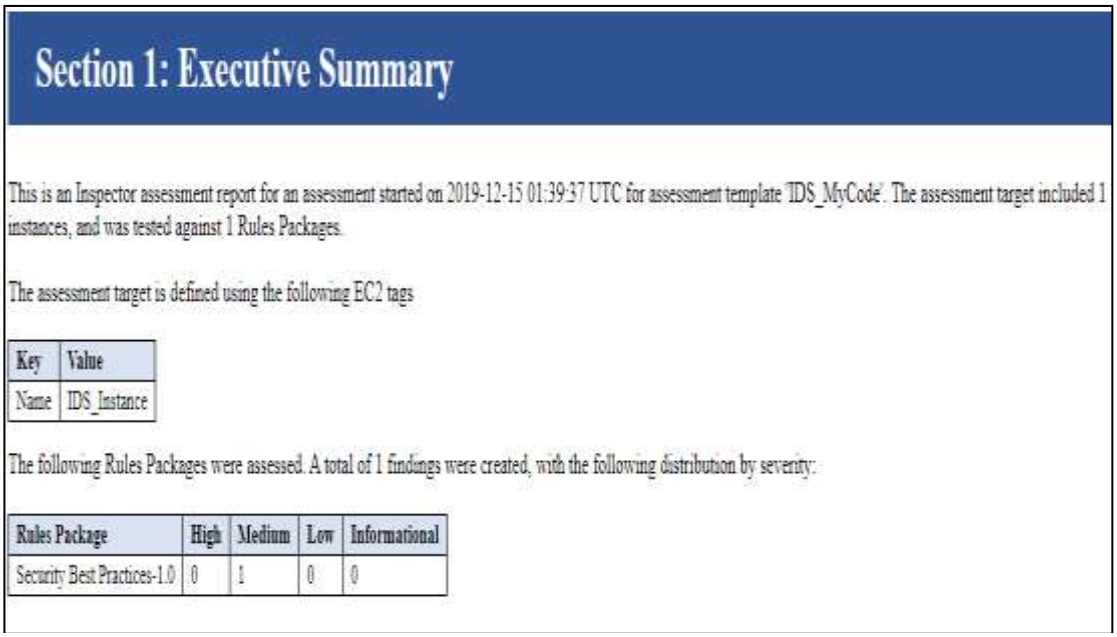
Eighth, the detection accuracy is furnished here.

Table 8: Cluster Analysis – class

Cluster	Actual Label anomaly (0) Normal (1)	Detected Label	Detection Accuracy (Matched Label is 100%) (%)
CG_7	0	0	99.93
CG_0	0	0	99.91
Main	0	0	99.79
CG_6	0	0	99.69
CG_8	0	0	99.49
CG_5	0	0	98.59
CG_2	0	0	97.40
CG_4	0	0	97.23
CG_1	0	1	60.00
CG_3	0	1	60.00

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

The overall detection process of the framework is almost 91.2% accurate, while the accuracy of the chosen clusters is 99%. The testing details for the cloud-based environment are also shown here.



Section 1: Executive Summary

This is an Inspector assessment report for an assessment started on 2019-12-15 01:39:37 UTC for assessment template 'IDS_MyCode'. The assessment target included 1 instances, and was tested against 1 Rules Packages.

The assessment target is defined using the following EC2 tags

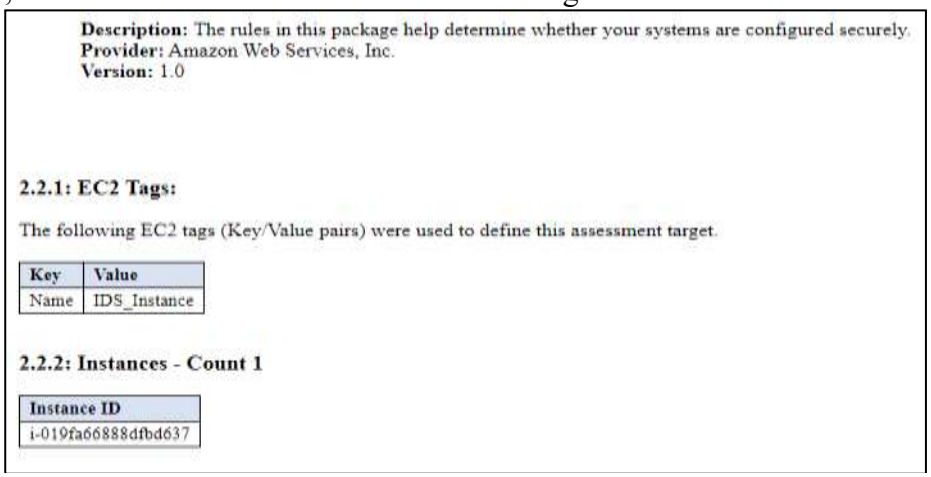
Key	Value
Name	IDS_Instance

The following Rules Packages were assessed. A total of 1 findings were created, with the following distribution by severity:

Rules Package	High	Medium	Low	Informational
Security Best Practices-1.0	0	1	0	0

Figure 11: Defined Cluster Analysis

Also, the detection results from the above findings are showcased here.



Description: The rules in this package help determine whether your systems are configured securely.
Provider: Amazon Web Services, Inc.
Version: 1.0

2.2.1: EC2 Tags:

The following EC2 tags (Key/Value pairs) were used to define this assessment target.

Key	Value
Name	IDS_Instance

2.2.2: Instances - Count 1

Instance ID
i-019fa66888dfbd637

Figure 12: Cluster Analysis – Detailed Analysis

Now that the data have been compared, the analysis of how they compare may be found. In this part of the report, the work will also provide a comprehensive analysis of the outcomes.

8 COMPARATIVE ANALYSIS

This part also includes the presentation of the comparative analysis of the many parallel research projects that were carried out using the suggested paradigm.

Table 4.9: Comparative Analysis

Proposed Method, Author, Year	Parameter Extraction	Data Analysis Method	Base Method	Detection Accuracy
SVM, W. L. Al-Yaseen, 2017 [8]	Yes	Classification & clustering	SVM	83%
Host Log Analysis, M.	No	Text Mining	K-Mean	89%

Zhu, 2017 [10]				
DSE-CF-CI, 2019	Yes	Deep Clustering	Deep Clustering	92%
Clustering, N. P. Shetty et al, 2016 [12]	No	Clustering	K – Mean	78%
Clustering, M. Zhu et al., 2017 [13]	No	Clustering	Hierarchical	83%
Clustering, A. Sultana et al. 2016 [14]	No	Clustering	Mean Shift	83%
ACO, P. Ravi Kiran Varma, 2018 [9]	No	Optimization	Multiple Clustering and ACO	90%

In most scenarios, the type of Intrusion Detection System (IDS) and the configuration of the firewall serve as key indicators of a system's overall security posture. While both play vital roles in network defense, their functions are fundamentally different. A firewall primarily acts as a barrier, controlling and restricting access between internal and external networks to prevent unauthorized connections. However, it does not detect or respond to attacks that may originate or occur within the system itself.

In contrast, an Intrusion Detection System (IDS) actively monitors network traffic and system activities, aiming to identify and alert administrators to suspicious or unauthorized behavior. Rather than simply blocking access, the IDS provides visibility into potential intrusions and generates alerts when anomalous patterns are detected, enabling proactive threat response.

Following the detailed discussion of the findings, the final section of this report will present the overall conclusions of the study, highlighting the contributions made by the proposed approach. The results and analytical insights derived from the research will also be summarized in this section to demonstrate the effectiveness and relevance of the methodology. One of the key drivers behind the increasing number of security vulnerabilities is the proliferation of internet-based services. As highlighted in various studies, the surge in cyberattacks has led many organizations to lose confidence in developing automated defense solutions, primarily due to the complexity and evolving nature of modern threats. This research proposes an innovative deep clustering-based approach for the detection and prevention of security vulnerabilities. The foundation of the proposed model lies in the extraction of salient features from intrusion events, which are then analyzed using statistical distance measures and feature vectors to form effective clusters capable of identifying anomalous activities.

9. Conclusion

As cloud computing infrastructures continue to expand, the threat landscape grows increasingly sophisticated, exposing critical vulnerabilities in existing security frameworks. Traditional intrusion detection systems, which rely heavily on static signatures and rule-based approaches, are often inadequate in detecting advanced and previously unseen attacks. In response to these challenges, this study presents a contextual deep clustering-based framework for intrusion detection that addresses the limitations of conventional methods. The proposed system utilizes a hybrid approach, combining the strength of traditional clustering methods with a novel deep learning-enhanced clustering algorithm (DSE-CF-CI). By incorporating feature sensitivity, domain-specific data behavior, and contextual similarity measures, the model constructs personalized cluster centroids tailored to dynamic input

patterns. This leads to a significantly higher detection rate and fewer false positives, as shown through experimental analysis on the KDD CUP 99 dataset. Key contributions of this work include the development of a similarity-aware rule engine, knowledge-based feature reduction (KBFSS), and a real-time cloud deployment model capable of monitoring system-level indicators to detect intrusions proactively. The system demonstrated an impressive overall detection accuracy of 91.2%, with selected clusters achieving 99% precision in detecting anomalous patterns. These results significantly outperform traditional models such as SVM-based classifiers, standalone clustering methods, and heuristic optimization approaches. Further, the study explores the operational significance of contextual information in refining the clustering process. By evaluating key attributes such as protocol type, guest login behavior, and host connection rate, the model adapts to subtle attack patterns that conventional methods might overlook. The live monitoring system, integrated within a cloud architecture, facilitates dynamic behavior tracking, allowing for immediate detection and response to anomalous activities. The comparative evaluation with existing models from recent literature underscores the advantages of the proposed framework. While traditional techniques often suffer from scalability and adaptability issues, the deep clustering model proves to be computationally efficient and capable of handling nonlinear, high-dimensional datasets. The rule engine embedded within the architecture enables intelligent inference, allowing the system to generalize across different attack types without needing manual intervention or retraining. In conclusion, this research not only presents an innovative solution for detecting intrusions but also lays the groundwork for the next generation of adaptive and intelligent IDS frameworks. The modularity and scalability of the proposed system make it well-suited for deployment in cloud environments, where agility and real-time responsiveness are critical. Future work can extend this approach by integrating reinforcement learning mechanisms for automated rule refinement and expanding the framework to support heterogeneous IoT-based networks.

References

[1]. Al-A'araji, Nabeel H., Safaa O. Al-Mamory, and Ali H. Al-Shakarchi. "Classification and Clustering Based Ensemble Techniques for Intrusion Detection Systems: A Survey." Journal of Physics: Conference Series. Vol. 1818. No. 1. IOP Publishing, 2021.

[2]. Alzahrani, Abdulsalam O., and Mohammed JF Alenazi. "Designing a network intrusion detection system based on machine learning for software defined networks." Future Internet 13.5 (2021): 111.

[3]. AZIZ, MOHAMMAD NASRUL, and TOHARI AHMAD. "CLUSTERING UNDER-SAMPLING DATA FOR IMPROVING THE PERFORMANCE OF INTRUSION DETECTION SYSTEM." Journal of Engineering Science and Technology 16.2 (2021): 1342-1355.

[4]. Dhingra, Madhavi, S. C. Jain, and Rakesh Singh Jadon. "Malicious node detection based on clustering techniques in network." Materials Today: Proceedings 47 (2021): 6676-6678.

[5]. Kocher, Geeta, and Gulshan Kumar. "Machine learning and deep learning methods for intrusion detection systems: recent developments and challenges." Soft Computing 25.15 (2021): 9731-9763.

[6]. Maheswari, M., and R. A. Karthika. "A novel QoS based secure unequal clustering protocol with intrusion detection system in wireless sensor networks." Wireless Personal Communications 118.2 (2021): 1535-1557.

[7]. Qaddoura, Raneem, et al. "A multi-stage classification approach for iot intrusion detection based on clustering with oversampling." Applied Sciences 11.7 (2021): 3022.
[8]. J. Lin, “Accelerating Density Peak Clustering Algorithm,” pp. 1–18, 2019.
[9]. Y. Shi and H. Shen, “Anomaly Detection for Network Flow Using Immune Network and Density Peak,” vol. 2019, pp. 1–10, 2019.
[10]. P. Ravi Kiran Varma, V. Valli Kumari, and S. Srinivas Kumar, “A Survey of Feature Selection Techniques in Intrusion Detection System: A Soft Computing Perspective,” in Advances in Intelligent Systems and Computing, Singapore: Springer Singapore, vol. 710, pp. 785–793, 2018.
[11]. N. Science, C. Phenomena, L. Li, H. Zhang, H. Peng, and Y. Yang, “Chaos , Solitons and Fractals Nearest neighbors based density peaks approach to intrusion detection,” vol. 110, pp. 33–40, 2018.
[12]. S. Otoum, B. Kantarci, H. T. Mouftah, "Adaptively supervised and intrusion-aware data aggregation for wireless sensor clusters in critical infrastructures", Proc. IEEE Int. Conf. Commun. (ICC), pp. 1-6, May 2018.
[13]. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, “A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection,” IEEE Communications Surveys & Tutorials, pp. 1–1, 2018.
[14]. J. Shen, J. Xia, Y. Shan, and Z. Wei, “Classification model for imbalanced traffic data based on secondary feature extraction,” IET Communications: IET Journals, vol. 11, no. 11, pp. 1725–1731, 2017.
[15]. M. Tabatabaefar, M. Miriestahbanati, J.-C. Grégoire, "Network intrusion detection through artificial immune system", Proc. Annu. IEEE Int. Syst. Conf. (SysCon), pp. 1-6, Apr. 2017.
[16]. M. Zhu and Z. Huang, “Intrusion detection system based on data mining for host log,” IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (lAEAC), pp. 1742–1746, 2017.
[17]. L. Al-Yaseen, Z. A. Othman, and M. Z. A. Nazri, “Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system,” Expert Systems with Applications, vol. 67, pp. 296–303, 2017.
[18]. Vijayanand, D. Devaraj, B. Kannapiran, "Support vector machine based intrusion detection system with reduced input features for advanced metering infrastructure of smart grid", Proc. IEEE 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), pp. 1-7, Jan. 2017.
[19]. Malhotra, V. Bali, K. K. Paliwal, "Genetic programming and k-nearest neighbour classifier based intrusion detection model ", Proc. IEEE 7th Int. Conf. Cloud Comput. Data Sci. Amp Eng. Conf., pp. 42-46, Jan. 2017.
[20]. N. P. Shetty, “Using clustering to capture attackers,” International Conference on Inventive Computation Technologies (lCICT): IEEE, vol. 3, pp. 1–5, 2016.