MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 383-386

Improving Accuracy in IDS by Means of an Unsupervised Clustering Approach and Reducing False Positives

Pratik Jain[1], Khushbu Tikhe Konde[2], Sonali Abhijeet Padalkar[3], Nirmal Keshari Swain[4], Kavita Nagar[5], Narendra Kumar Sahu[6]
[1]IPS Academy, Institute of Engineering and Science, Indore, Email id: pratikjain@ipsacademy.org
[2]Shree L. R. Tiwari College of Engineering, Thane, Email id: tikhe.khushbu@gmail.com
[3]Shree L. R. Tiwari College of Engineering, Thane, Email id: sonalipadalkar88@gmail.com
[4]Vardhaman College of Engineering, Hyderabad, Email id: swain.nirmal6@gmail.com
[5]Government Women's Polytechnic College, Indore, Email id: nagar.kavita1510@gmail.com
[6]Government Women's Polytechnic College, Indore, Email id: naru_sahu@yahoo.com

ABSTRACT: The Intrusion Detection Systems (IDS) are essential in ensuring the protection of computer networks from attacks. However, high false positive rates represent a major obstacle towards improving their efficacy. The problem associated with IDS may be partly attributed to the inclusion of irrelevant attributes in benchmark datasets that are employed to test various classification approaches. In this regard, the current study focuses on investigating how the removal of count attribute will affect the performance of an IDS. Through exclusion of this attribute, it is expected that the number of false alarms will be greatly reduced without sacrificing the ability of an IDS to detect intrusions. Standard classification approaches were used to conduct experiments based on the modified version of KDD Cup 1999 data set. The performance of IDS was assessed through analysis of the detection accuracy, false positive rate, and other parameters. The results indicate that eliminating the count attribute is likely to contribute to a decline in the number of false positives.

Keywords: Intrusion Detection System, False Alarm Reduction, Feature Selection, Count Attribute Elimination, KDD Cup 1999 Dataset, Anomaly Detection.

I. INTRODUCTION

With the quick development of network technologies and digital services, people have been more exposed to cyber attacks. Thus, intrusion detection has become an integral part of any cybersecurity infrastructure. The purpose of intrusion detection systems (IDS) is to track all the activity on the network or particular hosts and detect abnormal behavior. Even though there have been numerous improvements in IDS technology, it is still incapable of detecting new kinds of attacks. Signature based intrusion detection systems can only recognize cyberattacks that have occurred previously and have been included in the database. To deal with this problem, the idea of anomaly based intrusion detection has appeared. It allows recognizing deviations in patterns of activity in networks or hosts, which indicates that a cyberattack has taken place. This method has proven to be effective since it can help recognize zero-day attacks. Nevertheless, one should note that anomaly based IDS faces the issue of excessive false positives. They refer to the cases when legitimate behavior is recognized as an intrusion. Such situations decrease the efficiency of an IDS and make security specialists work harder to analyze the situation and prevent potential threats. The objective of this study is to better detect anomalies and improve detection techniques so that they can differentiate between benign and actual attacks. Through the reduction of false positives, the method presented here hopes to enhance the effectiveness and efficiency of intrusion detection systems.

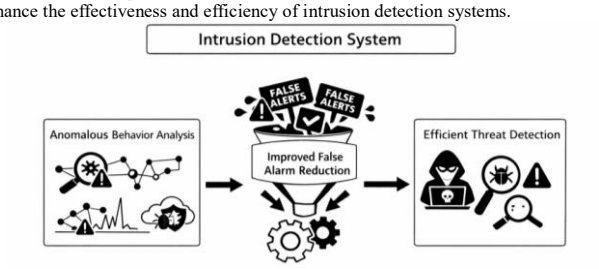

Figure 1.1

The Figure 1.1 shows the flow of work of an Intrusion Detection System (IDS) in order to improve its efficiency by means of performing analysis of anomalous behavior and reduction of false alarms. First, there comes Anomalous Behavior Analysis, where monitoring of the network or computer system activity takes place. Here, the normal behavior pattern is recognized, and any deviation from it is marked as an anomaly. In this way, the IDS will be able to detect unknown or 'zero day' attacks for which there are no signature-based detections available. Further, these anomalies go to the Improved False Alarm Reduction module. This step works as the filter, which separates real security threats from the non-threatening anomalies in the system activities. Using appropriate detection rules or decision-making mechanisms, it filters out false alarms, thereby minimizing their number. The filtered data goes to the next step - the Efficient Threat Detection. Here the actual threats are detected, and reports are made. As a result, all of this will lead to efficient intrusion detection. The given figure shows the design architecture of the IDS constructed on the basis of the KDD Cup 1999 Dataset. The proposed flow chart focuses on enhancing the performance of IDS in terms of detection accuracy while maintaining low false alarms. As shown in the figure, initially the network traffic data collected through the KDD Cup 1999 Dataset are introduced into the IDS. This set of data includes a variety of normal and attack traffic. Hence, it is appropriate for the modeling of normal behaviors as well as the learning process about different attacks that may occur in a network. The Anomalous Behavior Analysis phase deals with analyzing the selected features within the given set of data. Deviations from the learned normal behaviors are considered anomalies. These detected anomalies are further processed within the Improved False Alarm Reduction stage. It uses a more advanced algorithm for classifying those abnormalities to eliminate the threat of false alarm. This stage contributes significantly to eliminating unnecessary warnings provided by IDS. Finally, efficient threats are detected at the Efficient Threat Detection stage. This facilitates accurate identification of different attack types, including denial of service, probing, user-to-root, and remote-to-local attacks, available in the KDD Cup 1999 data set. The architecture shown in the figure below shows how the adoption of the KDD Cup 1999 data set, along with the application of anomaly detection and minimization of false positives, improves the effectiveness of intrusion detection systems.

II. Literature Survey

Dissanayake and Thayasivam (2025) proposed an ensemble deep learning IDS specializing in attack specific detection. Their fusion approach significantly reduced false positives and improved detection robustness**[1]. Layman and Roden (2023)** studied the impact of false alarm rates on security analysts. Their findings showed that high false positives reduce analyst efficiency and response accuracy, emphasizing the importance of false alarm reduction **[2]. Khandelwal and Shanker (2024)** developed a lightweight IDS using hybrid FPGA architectures. Their approach achieved low false alarm rates while maintaining real time detection efficiency [3]. **Gueriani et al. (2024)** surveyed deep reinforcement learning approaches for IDS in IoT environments. The authors emphasized that adaptive learning mechanisms can reduce false alarms in dynamic network conditions [4]. **Altulaihan et al. (2024)** proposed a machine learning based anomaly detection IDS for IoT networks. Using supervised learning, the system achieved high detection accuracy with reduced false positive rates [5]. **Bhavsar et al. (2023)** proposed an anomaly based IDS for IoT applications. The study demonstrated improved detection rates using machine learning while acknowledging challenges in false alarm control [6]. **Rahman et al. (2025)** provided a survey on IDS in IoT networks, highlighting anomaly based detection as the most promising approach for unknown attack detection while stressing the need for false alarm optimization [7]. **Yin et al. (2017)** introduced a recurrent neural network based IDS using the KDD dataset. The model effectively captured temporal dependencies in network traffic, achieving higher detection rates compared to traditional classifiers, though false positives remained a challenge [8]. **Naseer et al. (2018)** developed a deep neural network based IDS that improved anomaly detection accuracy. Their work highlighted the importance of deep feature representation but noted the computational overhead involved [9]. **Xiao et al. (2019)** presented a CNN based intrusion detection model with feature reduction techniques. The proposed approach improved detection accuracy while reducing false alarms, showing the effectiveness of dimensionality reduction [10]. **Jiang et al. (2020)** proposed a hybrid sampling technique combined with deep learning to handle imbalanced IDS datasets. Their method improved classification performance across minority attack classes [11]. **Liao et al. (2013)** presented a comprehensive review of intrusion detection systems, categorizing them into signature based, anomaly based, and hybrid systems. The study emphasized that anomaly based IDS are effective for detecting unknown attacks but often suffer from high false alarm rates, motivating further research into accuracy enhancement [12]. **Buczak and Guven (2015)** surveyed data mining and machine learning techniques used in IDS. They concluded that machine learning based anomaly detection improves detection capability but requires effective feature selection and classification techniques to reduce false positives [13]. **Tavallaee et al. (2009)** conducted a detailed analysis of the KDD Cup 1999 dataset and highlighted its strengths and limitations for intrusion detection research. The authors discussed issues such as redundant records and class imbalance, which significantly affect IDS evaluation. Their work laid the foundation for improving IDS performance using refined datasets and evaluation metrics [14]. **Farnia (2017)** proposed a one class SVM based anomaly detection system focused on minimizing false alarm rates. The study demonstrated that careful threshold selection significantly improves IDS reliability [15]. **Kim et al. (2016)** proposed an LSTM based host intrusion detection system that models system call sequences. Their ensemble approach improved detection accuracy and reduced false alarms, demonstrating the effectiveness of deep learning for sequential anomaly detection [16]. **Yang et al. (2020)** introduced a deep attention based network for payload inspection in IDS. The model improved attack detection precision by focusing on critical features, thereby reducing false alerts [17]. **Altulaihan et al. (2024)** focused on detecting DoS attacks using anomaly based IDS and achieved improved accuracy through feature optimization [18]. It demonstrated that feature selection significantly affects IDS false alarm rates and detection performance [19]. It explored IDS usability and emphasized that reducing false alarms improves operational effectiveness and trust in IDS systems [20]. It applied recursive feature elimination to anomaly based IDS and demonstrated improved classification accuracy with reduced false alarm rates [21]. It evaluated IDS performance using anomaly and signature based approaches and showed that hybrid systems reduce false alarm rates more effectively [22]. **)** introduced a False to True Alert Rate metric to evaluate IDS usability, highlighting the importance of meaningful alert generation [23].

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 383-386

III. Problem Identification

An Intrusion Detection System (IDS) is highly important for securing computer networks as it detects intrusions in the network. Although Anomaly Based IDS is efficient in detecting unknown attacks, it has a major problem; the false alarm rate is too high and often misclassifies legitimate activities in the network as an attack. Such a situation affects the efficiency and effectiveness of any Intrusion Detection System. High false alarm rates increase the amount of alerts generated that increase computation and overload security personnel with unnecessary work. As more and more alerts are generated, the efficiency of response of the system is highly reduced, as it results in the delay of the identification of any other possible threats in the system. Therefore, the efficiency of any IDS depends on both the accuracy of its detection and the number of unnecessary alerts generated. As an example, in a practical world scenario, there might be cases when there is an increase in network activity because of normal activities, such as backup or updating of the software. In addition, an anomaly-based intrusion detection system lacking appropriate mechanisms for reducing false alarms may interpret these usual changes as Denial of Service attacks and produce many false alarms. Thus, the staff in charge of maintaining network security can spend unnecessary time looking into these events instead of detecting any real threats. For this reason, the primary concern of this research work is how to minimize false alarms produced by intrusion detection systems without affecting their detection ability. This improvement will not only lead to increased effectiveness of IDS but will also help optimize the use of resources as well as increase its performance in a timely and accurate manner. This paper pays attention to studying anomalies and developing IDS mechanisms in such a way that their performance can be analyzed on the basis of benchmarking datasets like KDD Cup 1999. In the current research, experimental tests were carried out using the dataset known as KDD Cup 1999 which was created for the purpose of the DARPA Intrusion Detection Evaluation Program organized by MIT Lincoln Laboratory in 1998. The well-controlled nature of this data and its popularity have made it a benchmark database in intrusion detection studies, which was later used in the KDD Cup 1999 contest. One of the central concerns of this study is a situation called false positive, whereby the normal operation of a network is wrongly identified as an intrusion. The concept of false positives is a significant problem in IDS, as it affects the efficiency and dependability of such systems. This dataset can be considered a simulated military network that included several cases of intrusions to evaluate the system performance in realistic scenarios. There are 41 attributes for each entry in the database to determine whether it is a case of normal operation or an anomaly. Some of these attributes include those relating to connection properties, content-related attributes, and statistics regarding the traffic. Based on these characteristics, the data instances are grouped as either normal or anomalies. Attributes consist of factors like connection time, protocol, service, status, source/destination bytes, login features, file accesses, and host traffic characteristics. These attributes are labeled as Z1 to Z41 for easy visualization. Based on this set of attributes, the intrusion detection system decides whether the input data sample belongs to legitimate network activity or not. In this case, the classification of data is done by identifying the patterns associated with the 41 attributes. To illustrate, a selected few samples from the KDD Cup 1999 dataset are chosen to explain how different behaviors can be identified using attribute values. From the comparison between the anomaly class and normal classes, it is observed that there are 41 features that determine the type of input, i.e., whether the input falls into the category of a normal class or anomaly class. The 41 features include: Feature 1: Duration, represented by Z1 Feature 2: Protocol Type, represented by Z2 Feature 3: Service, represented by Z3 Feature 4: Flag, represented by Z4 Feature 5: Source Bytes, represented by Z5 Feature 6: Destination Bytes, represented by Z6 Feature 7: Land, represented by Z7 Feature 8: Wrong Fragment, represented by Z8 Feature 9: Urgent, represented by Z9 Feature 10: Hot, represented by Z10 Feature 11: Number of Failed Logins, represented by Z11 Feature 12: Logged In, represented by Z12 Feature 13:Number Compromised, represented by Z13 Feature 14: Root Shell, represented by Z14 Feature 15: Su Attempted, represented by Z15 Feature 16: Num root, represented by Z16 Feature 17: Num File Creation, represented by Z17 Feature 18: Num Shells, represented by Z18 Feature 19: Num Access Files, represented by Z19 Feature 20: Num Outbound Cmmds, represented by Z20 Feature 21:Is Host Login, represented by Z21

Z22 is guest login attribute, Z23 stands for attribute 'count', Z24 stands for srv_count attribute, Z25 stands for serror_rate attribute, Z26 stands for srv_serror_rate attribute, Z27 is represented as an rerror_rate attribute, Z28 is a srv_rerror_rate attribute, Z29 stands for same_srv_rate attribute, Z30 represents diff_srv_rate attribute, Z31 is a srv_diff_host_rate attribute, Z32 is a dst_host_count attribute, Z33 stands for a dst_host_srv_count attribute, Z34 is a dst_host_same_srv_rate attribute, Z35 stands for a dst_host_diff_srv_rate attribute, Z36 is dst_host_same_src_port_rate attribute, Z37 is a dst_host_srv_diff_host_rate attribute, Z38 is a dst_host_serror_rate attribute, Z39 stands for a dst_host_srv_serror_rate attribute, Z40 is dst_host_rerror_rate attribute, Z41 stands for a dst_host_srv_rerror_rate attribute, Class: Class A - normal, Class B - anomaly Now, depending on these 41 attributes, it will be determined whether the data is normal or anomaly. For example: Let us consider data from the KDD Cup 1999 dataset.

Table 4.1: Shows the details of Final Cluster Centeroids with count attribute

Attribute	Full Data (25192.0)	Cluster #0 (9695.0)	1 (15497.0)
Z_1	305.0541	533.1584	162.3509
Z_2	tcp	tcp	tcp
Z_3	http	private	http
Z_4	SF	S0	SF
Z_5	24330.6282	39374.1009	14919.3572
Z_6	3491.8472	115.1045	5604.3541
Z_7	0	0	0
Z_8	0.0237	0.0175	0.0276
Z_9	0	0	0.0001
Z_{10}	0.198	0.0018	0.3208
Z_{11}	0.0012	0.0002	0.0018
Z_{12}	0	0	1
Z_{13}	0.2279	0	0.3704
Z_{14}	0.0015	0.0001	0.0025
Z_{15}	0.0013	0.0002	0.0021
Z_{16}	0.2498	0.0005	0.4058
Z_{17}	0.0147	0.001	0.0233
Z_{18}	0.0004	0	0.0006
Z_{19}	0.0043	0	0.007
Z_{20}	0	0	0
Z_{21}	0	0	0
Z_{22}	0	0	0
Z_{23}	84.5912	166.3895	33.4178
Z_{24}	27.6988	9.9234	38.8191
Z_{25}	0.2863	0.7253	0.0117
Z_{26}	0.2838	0.7212	0.0101
Z_{27}	0.1186	0.2467	0.0385
Z_{28}	0.1203	0.2486	0.04
Z_{29}	0.6606	0.163	0.9718
Z_{30}	0.0624	0.1195	0.0266
Z_{31}	0.0959	0.0013	0.1551
Z_{32}	182.5321	245.2073	143.3221
Z_{33}	115.063	12.5472	179.1975
Z_{34}	0.5198	0.0554	0.8103
Z_{35}	0.0825	0.1512	0.0396
Z_{36}	0.1475	0.0636	0.1999
Z_{37}	0.0318	0.0067	0.0476
Z_{38}	0.2858	0.7215	0.0133
Z_{39}	0.2798	0.7179	0.0058
Z_{40}	0.1178	0.2368	0.0434
Z_{41}	0.1188	0.2478	0.0381
Class	Normal	Anomaly	Normal

Time taken to build model (full training data) : 1 seconds

=== Model and evaluation on training set ===

Clustered Instances
0 9695 (38%)
1 15497 (62%)

Log likelihood: -36.48805

The set of 41 attributes plays a crucial role in identifying malicious or anomalous behavior, as well as in determining the time required for detection. However, processing a large number of attributes can increase computational complexity and reduce system efficiency. Therefore, reducing the number of attributes is essential to improve the overall performance of the intrusion detection system. While performing attribute reduction or reorganization, the primary objective of maintaining high detection accuracy must be carefully considered. Anomaly detection is based on comparing observed values with their corresponding mean values. If multiple attribute values deviate significantly from the established mean, the data instance is classified as anomalous. One of the major challenges in IDS is the false positive problem, where normal traffic is incorrectly identified as an attack. This issue can be mitigated by eliminating less informative attributes, such as the count attribute, which may contribute to unnecessary false alerts. Furthermore, timely detection of attacks is critical, as identifying malicious activities at an early stage helps prevent potential damage. Hence, improving detection speed while reducing false positives remains a key focus of this research.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 383-386

IV. EXPERIMENT AND RESULTS

It has been found that the count attribute increases the number of false positives in the IDS. For effective evaluation of system performance, the most important metrics are detection rate and false alarm rate. While the latter metric is the number of normal examples labeled as attacks divided by the total number of normal examples, the former metric is the number of attack examples labeled as such divided by the total number of attack examples. Together, the mentioned metrics indicate how accurately the intrusion detection system identifies attacks and generates errors in its work. Therefore, they are critical for evaluating system performance, which can be done through calculation of the specified metrics with sample data. Given that the count attribute is not used efficiently and contributes to low accuracy, it can be excluded from further use as a tool for decreasing the number of false positives. On the other hand, a possible way to improve system performance is through the use of an One-Time Password (OTP)-based system. It would improve access controls, reducing any unauthorized actions, by sending the user an OTP either via their email or phone number. It can be seen that using OTP authentication will thus enhance security without generating false intrusion warnings.

Algorithm: User Registration and OTP Authentication Process

Phase 1: User Registration (General Information Form)

Input: General user information including credentials

Output: Database-stored user profile

1. Begin the program.
2. The system will present the registration form.
3. The user will input his/her general information like:
   - Name
   - Email
   - Phone number
   - Username
   - Password
4. Verify user data validity and completeness.
5. If any data entry is missing or invalid:
   - Provide an error notification.
   - The user should re-input the data.
   -
6. If the entered data is valid:
   - Securely encrypt the password.
   - The user profile will be saved into the database.
7. Counter for login attempts initialized at zero.
8. Conclude the registration process.
9. End Phase 1.

Phase 2: Login Verification and OTP Provisioning

Input: Username and password

Output: User authentication with OTP

1. Initiate login procedure.
2. Enter username and password.
3. Check whether entered credentials match the credentials stored in the database.
4. If there is a match:
   - Set login attempts count to zero.
   - Authorize user access.
   - End login procedure.
5. If there is no match:

o Increase the number of login attempts by 1.

6. Check the number of unsuccessful logins.
7. If the number of logins is less than five:
   - Say that the password is invalid.
   - Let the user log in again.
8. If there are five unsuccessful logins:
   - Create a One-Time Password (OTP).
   - Forward the OTP to the user's email.
   - Request the entry of OTP.
9. Test whether the OTP is valid.
   - If it matches, then:

   Re-set login attempts count to zero.
   Allow the user to set a new password or access.

o If it does not match, then:
   - Access is denied; try again.

10. Finish login procedure.

The design of the user account registration and OTP verification process ensures that user accounts are created securely and also prevents any unauthorized access to their accounts. In the initial stage, the system will present the user with a form which they must fill out with basic information such as the user's name, e-mail address, telephone number, and passwords among others. The system then goes ahead to confirm whether all the required information has been filled. In case some of the information is left out, the user will be prompted to provide the correct information. Once the system receives all the required information from the user, the password is encrypted and the user's information is saved in the database along with an initialized login attempt counter. The second phase involves authentication and securing the system with the help of a one-time password (OTP). During login, the user needs to provide the username and password, which will be validated with the values stored in the database. In case they are verified, the attempt counter is reset to zero, and access is granted to the user. Otherwise, the system will increment the login attempt count. Until the attempt counter is less than five, the user can be provided with another chance to try the login process after being notified about the mistake. In case the attempt counter becomes equal to five, the system will send the user a one-time password through the registered email address. Now, the user needs to verify their identity by providing the generated OTP. After successful OTP validation, the counter is reset to zero, and the user is granted access to their account or the option to set a new password. Otherwise, the system will deny access to the user until he/she succeeds in the login procedure.

Table 4.2: Shows the details of Final Cluster Centeroids without count attribute.

MSW MANAGEMENT -Multidisciplinary, Scientific Work and Management Journal
ISSN: 1053-7899
Vol. 36 Issue 2, 2026, Pages 383-386

Attribute	Full Data (25192.0)	Cluster #0 (9719.0)	1 (15473.0)
Z_1	305.0541	533.1584	162.3509
Z_2	tcp	tcp	tcp
Z_3	http	private	http
Z_4	SF	S0	SF
Z_5	24330.6282	39277.0561	14942.3821
Z_6	3491.8472	114.8202	5613.047
Z_7	0	0	0
Z_8	0.0237	0.021	0.0255
Z_9	0	0	0.0001
Z_{10}	0.198	0.0017	0.3213
Z_{11}	0.0012	0.0002	0.0018
Z_{12}	0	0	1
Z_{13}	0.2279	0	0.371
Z_{14}	0.0015	0.0001	0.0025
Z_{15}	0.0013	0.0002	0.0021
Z_{16}	0.2498	0.0005	0.4064
Z_{17}	0.0147	0.001	0.0233
Z_{18}	0.0004	0	0.0006
Z_{19}	0.0043	0	0.007
Z_{20}	0	0	0
Z_{21}	0	0	0
Z_{22}	0	0	0
Z_{24}	27.6988	9.9546	38.8443
Z_{25}	0.2863	0.7236	0.0117
Z_{26}	0.2838	0.7194	0.0101
Z_{27}	0.1186	0.2461	0.0386
Z_{28}	0.1203	0.248	0.04
Z_{29}	0.6606	0.165	0.9719
Z_{30}	0.0624	0.1193	0.0266
Z_{31}	0.0959	0.0013	0.1553
Z_{32}	182.5321	245.2217	143.155
Z_{33}	115.063	12.5829	179.4335
Z_{34}	0.5198	0.0555	0.8114
Z_{35}	0.0825	0.1513	0.0393
Z_{36}	0.1475	0.0648	0.1994
Z_{37}	0.0318	0.0067	0.0476
Z_{38}	0.2858	0.7197	0.0132
Z_{39}	0.2798	0.7161	0.0058
Z_{40}	0.1178	0.2364	0.0433
Z_{41}	0.1188	0.2472	0.0381
Class	Normal	Anomaly	Normal

Time taken to build model (full training data) : 0.77 seconds
=== Model and evaluation on training set ===

Clustered Instances
0 9719 (39%)
1 15473 (61%)

Log likelihood: -36.48805

Table 4.3: Portrays the comparison of results **Filtered Clusterer** Centeroids algorithm.

Algorithm	Time taken with count attribute	Time taken without count attribute
Filtered Clusterer	1 seconds	0.77 seconds

V. Conclusion

The present study aims at examining the effect of attribute optimization on the operation of an anomaly-based intrusion detection system based on KDD Cup 1999 database with a view towards increasing detection efficiency and minimizing false positives. To this end, the Filtered Clusterer with Centroid-based Clustering method was used to study network traffic patterns and identify normal and abnormal network instances. According to experimental data, eliminating the count attribute helps considerably reduce computation time, but does not affect detection capacity. In particular, the count attribute was determined to cause an increase in the number of false positives and processing overheads. The exclusion of the attribute led to quicker algorithm convergence and helped generate better centroids, which positively impacted algorithm performance. Comparison analysis showed that the average detection time was decreased from 1.00 to 0.77 seconds, which is an essential improvement in terms of detection speed. Thus, the removal of some attributes proves to be beneficial for IDS performance, especially when working with massive datasets like KDD Cup 1999. Also, attribute elimination contributed to better discrimination between anomalies, preventing unnecessary computations. Overall, this paper has demonstrated how employing a combination of the Filtered Clusterer, centroid based clustering, and attribute elimination is capable of providing a more efficient intrusion detection process. This method can help to minimize the time taken to perform tasks and reduce false alarms within the system. Potential future studies could involve looking into more effective means of selecting features for the process.

References

[1] N. Dissanayake and U. Thayasivam, “Attack-Specialized Deep Learning with Ensemble Fusion for Network Anomaly Detection”, arXiv:2510.12455, Oct. 2025.
[2] L. Layman and W. Roden, “A Controlled Experiment on the Impact of Intrusion Detection False Alarm Rate on Analyst Performance”, arXiv:2307.07023, Jul. 2023.
[3] Shashwat Khandelwal and Shreejith Shanker, “A Lightweight Multi-Attack CAN Intrusion Detection System on Hybrid FPGAs”, arXiv:2401.10689, Jan. 2024.
[4] A. G. Gueriani, H. Kheddar, and A. C. Mazari, “Deep Reinforcement Learning for Intrusion Detection in IoT: A Survey”, arXiv:2405.20038, May 2024.
[5] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, “Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms”, Sensors, vol. 24, no. 2, Art. 713, 2024.
[6] M. Bhavsar et al., “Anomaly-based intrusion detection system for IoT application”, Discover Internet of Things, 2023.
[7] Md. M. Rahman, S. Al Shakil, and M. R. Mustakim, “A survey on intrusion detection system in IoT networks”, Cyber Security and Applications, vol. 3, 100082, 2025.
[8] C. Yin, Y. Zhu, J. Fei, and X. He, “A deep learning approach for intrusion detection using recurrent neural networks”, IEEE Access, vol. 5, pp. 21954–21961, 2017.
[9] S. Naseer et al., “Enhanced network anomaly detection based on deep neural networks”, IEEE Access, vol. 6, pp. 48231–48246, 2018.
[10] Y. Xiao et al., “An intrusion detection model based on feature reduction and convolutional neural networks”, IEEE Access, vol. 7, pp. 42210–42219, 2019.
[11] K. Jiang, W. Wang, and A. Wang, “Network intrusion detection combining hybrid sampling with deep hierarchical network”, IEEE Access, vol. 8, pp. 32464–32476, 2020.
[12] H.-J. Liao, **et al.**, “Intrusion Detection System: A Comprehensive Review”, J. Netw. Comput. Appl., vol. 36, pp. 16–44, 2013.
[13] A. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection”, IEEE Commun. Surv. Tutor., vol. 18, pp. 1153–1176, 2015.
[14] T. Tavallaee et al., “A detailed analysis of the KDD CUP 99 data set”, Proc. IEEE Symp. Comp. Intelli. Sec. Def. Apps., pp. 1–6, 2009.

[15] F. Farnia, “Low-Rate False Alarm Anomaly-Based Intrusion Detection System with One-Class SVM”, École Polytechnique de Montréal Master’s Thesis, Oct. 2017 (updated 2024).
[16] G. Kim, H. Yi, J. Lee, Y. Paek, and S. Yoon, “LSTM-Based System-Call Language Modeling and Robust Ensemble Method for Host-Based IDS”, arXiv:1611.01726, Nov. 2016.
[17] S. Yang, P. Wu, and H. Guo, “DualNet: Locate Then Detect Effective Payload with Deep Attention Network”, arXiv:2010.12171, Oct. 2020.
[18] “Anomaly-Based Intrusion Detection Model Using Deep Learning for IoT Networks”, CMES, vol. 141, no. 1, pp. 823–845, Aug. 2024.
[19] “Anomaly-based intrusion detection system using recursive feature elimination for improved attack detection”, J. Netw. Comput. Appl., 2022.
[20] “Insights into anomaly-based intrusion detection systems usability”, ACM Trans. Secur. Priv., 2023.
[21] “Anomaly-Based Intrusion Detection in Strong Feature Spaces Using Recursive Feature Elimination”, Elsevier J., 2022.
[22] “Performance Evaluation of Intrusion Detection System Using Anomaly and Signature Based Algorithms to Reduction False Alarm Rate and Detect Unknown Attacks”, IEEE Conf., 2025.
[23] “False to True Alert Rate metric for IDS usability”, ACM Trans. Secur. Priv., 2023.