University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)

ISSN: 2714-3627

A Journal of the Faculty of Computing, University of Ibadan, Ibadan, Nigeria

Volume 16 No. 1, January 2026

journals.ui.edu.ng/uijslictr
http://uijslictr.org.ng/
uijslictr@gmail.com

University of Ibadan
Journal of Science and Logics in ICT Research

A Hybridized Data Mining Technique for Enhanced Network Intrusion Detection Performance

[1]✉Falowo G., [2]Agbolade S J., [3]Olorufemi, B.O., [4]Adejuwon B.A.

[1]Department of Cyber Security, Redeemer's University, Ede-Osun, Nigeria.
[2]Department of Computer Science, Redeemer's University, Ede-Osun, Nigeria.
[3]Department of Computer Science, Redeemer's University, Ede-Osun, Nigeria.
[4]Department of Computer Science, Federal University Oye-Ekiti, Nigeria.
[1]falowog@run.edu.ng. [2]agbolades@run.edu.ng. [3]olorufemib@run.edu.ng
[1]ORCID iD: https://orcid.org/0009-0004-6165-6254

Abstract
Conventional intrusion detection systems (IDS) are no longer efficient enough to recognize newly designed cyber-attacks because of increasing complexities and amount of network traffic data. A more effective approach for Data Mining (DM) is required for cybersecurity applications, although individual Data Mining strategies were sufficient for intrusion detection systems previously. In order to enhance precision and malleability for network intrusion detection systems (NIDS), this paper proposes a hybrid strategy for data mining using "CIC-IDS2017" dataset downloaded from Kaggle. This hybrid approach uses ensemble learning to increase classification efficiency by unifying the "Adaptive Boosting" approach strengths and the "C4.5 Decision Tree" algorithm technique concepts. Data preprocessing techniques, Label Encoding techniques, and classifiers belonging to the "Supervised Classification" category stood as key components of this strategy approach. Its efficiency is assessed using standard metrics. This proposed hybrid strategy approach resulted in near-perfect performance on its testing approach by generating 317,937 "True Positives" values, having "4" "False Positives" values, and having "Accuracy" of 99.9%. The performance of "C4.5 Classifier" approach also resulted in generation of 317,938 "True Positives" values having "5" "False Positives" values having "Precision" "Recall" "and F1-score" measures recorded at 99.9%. "Adaptive Boost" approach resulted in "317,185" "True Positives" values having "287" "False Positives" values having "Accuracy" "Precision" "Recall" "and F1-score" "values at "99.7%" "99.8%" "99.5%", "and 99.7%". This enhances development efforts of "intelligent" "cybersecurity" "systems" by applying "Deep" "Learning" "concept" further emphasizing "Data" "Mining" application for "network" "enhancements" to remain efficient

Keywords: *Network Intrusion Detection System, Hybrid Data Mining, Adaptive Boosting, C4.5 Decision Tree, Cybersecurity*

1. Introduction

The complexity and amount of network traffic have also increased concurrently with the rapid progress of internet technology and communications, thus also growing threats to cyber attacks. According to Al-Turjman & Zahmatkesh [4], network intrusion detection systems (NIDS) have become crucial tools for safeguarding business or corporate networks by monitoring and analyzing all entering and exiting network traffic for any signs of intrusion attempts. NIDS can also be divided into two main types: anomaly detection-based NIDS (ADNIDS) and signature-based NIDS (SNIDS). SNIDS use pre-defined attack signatures for comparison to detect intrusions within traffic, such as Snort software as described by Sarhan et al. [9].

Falowo G., Agbolade S. J., Olorunfemi B. O. and Adejuwon B.A. (2025). A Hybridized Data Mining Technique for Enhanced Network Intrusion Detection Performance. University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR), Vol. 16 No. 1, pp. 55 – 65.

On the flip side, ADNIDS have been capable of detecting threats not known before by identifying deviations from standard traffic norms. According to Ali et al. [2], high false positives and challenges associated with identifying relevant features for traffic statistics have often hampered the use of ADNIDS, despite their efficiency and capabilities being apparent to all. Data mining and machine learning techniques have been recently adopted to improve NIDS capabilities by researchers to overcome these limitations and difficulties discussed by Liu et al. [6]. These learning techniques involve classifiers such as Random Forest, Support Vector Machine, Artificial Neural Networks (ANNs), and Self-Organizing Maps designed to learn differences between normal and traffic anomaly traffic norms to improve traffic classification capabilities and performance for better network security and defense.

This work has developed a hybrid data mining approach by benefitting from adaptive boosting capabilities combined with those of the C4.5 decision tree algorithm to improve intrusion detection efficiency. Based on new advancements achieved in ensemble learning techniques, for instance, the stacking approach for SQL injection attacks classification proposed by Falowo et al. [5], it is argued that this hybrid approach is expected to improve intrusion detection precision while avoiding any significant increase in false positives. Additionally, this approach also promotes NIDS adaptability to new evolving threats to network security.

2. Related Works

Network intrusion detection system capabilities have also been widely advanced by new studies focusing on hybrid machine learning architectures. A stacking ensemble learning approach for SQL injection attacks was proposed by authors Falowo et al.[5]. The approach showed high effectiveness for database-style attacks by achieving 98.3% accuracy and 98.2% F1-score by learning from diverse classifiers. Nevertheless, its specificity to SQL attacks and need for tagged information make it less universally applicable for broader attacks too. Similarly, authors Sajid et al.[8] have proposed a hybrid strategy for cloud intrusion detection using deep learning and machine learning techniques. While achieving 99.1% detection efficiency, its computation complexities do not make it apt for real-time execution scenarios.

Moreover, Qazi *et al.* [11] developed HDLNIDS, which is a hybrid deep learning approach involving recurrent as well as convolutional neural networks to detect temporal as well as spatial features associated with network traffic. This hybrid approach may have shown 98.9% accuracy and 98.6% F1-score but lacked scalability because of its complexity and high training time. The drawback of class imbalance is removed by Talukder et al.[10], who applied oversampling and stacking embeddings for features to improve overall network traffic classification accuracy to 97.6%.

In anticipation of generalization improvement for anomaly-based systems, adaptive feature selection techniques were explored by Ali et al. [3]. While their technique provided 96.2% accuracy and 96.7% recall, it performed poorly on dynamic traffic scenarios because consistency and detection performance were affected by constantly varying feature importances.

The use of deep learning techniques such as ANN, LSTM, and Bi-LSTM was tested for ITS-related traffic classification tasks for efficiency by Mane and Rao [6]. In their experiment, Bi-LSTM demonstrated better performance than any of its deep learning counterparts because of its effectiveness at capturing time dependencies for 98.5% classification accuracy. Nevertheless, they needed extensive tuning and large-sized data for efficient performance. Using unsupervised anomaly detection was explored by Almomani [1], emphasizing its feasibility for discovering new attacks on unclassified data. Nevertheless, it demonstrated around 18% false positives, potentially causing security analysts to become inundated with irrelevant security alerts.

3. Research Methodology

This research methodology consists of four stages: first, data collection for the network intrusion detection system; second, preprocessing the dataset using different techniques before proceeding to the next phase; third, applying data mining algorithms to the pre-processed network intrusion system dataset; and finally, hybridizing data mining algorithms prior to applying a classification algorithm to the prepared network intrusion system dataset. The methods employed were assessed when the previously indicated phases were finished. The system architecture is shown in Figure 1:

3.1 Data Collection

The CIC-IDS2017 dataset developed by Canadian Institute of Cybersecurity is freely accessible at Kaggle or https://www.unb.ca/cic/datasets/ids-2017.html and is very extensive for NIDS-related studies for intrusion detection systems. It contains different types of attacks like brute force attacks, DDoS attacks, botnet attacks, and DoS attacks and is already labeled for supervised learning-related tasks. A total of 5 days of packets is involved for analysis and is further divided into eight different sessions. This dataset is further split into eight files containing each particular session: working hours of Wednesday morning and afternoon and evening, Tuesday morning and afternoon and evening, and Friday morning and afternoon and evening. Then comes Thursday's morning and afternoon and different files for Monday's morning and afternoon working hours. For machine learning analysis, this dataset is processed for Comma-Separated Values (.csv) file formats too. Note: This project further makes use of only one file from eight files: Wednesday working hour.csv too. This dataset contains 79 features or columns and 692703 rows or instances. NIDS attacks are further split into two classes: ATTACK (1) containing 251,723 cases and BENIGN (0) containing 439,683 cases and right now is being analyzed for further ML-related tasks: This dataset is further shown through its sample in Figure 2.

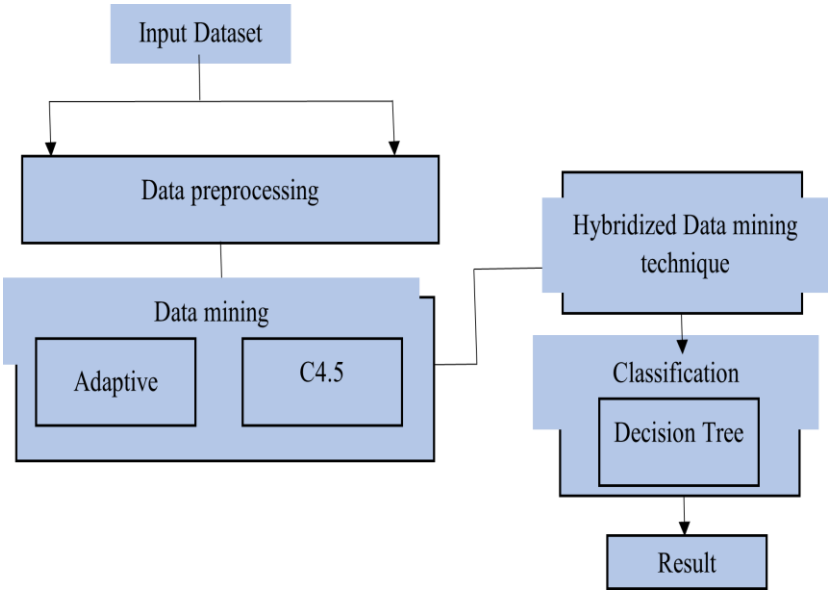
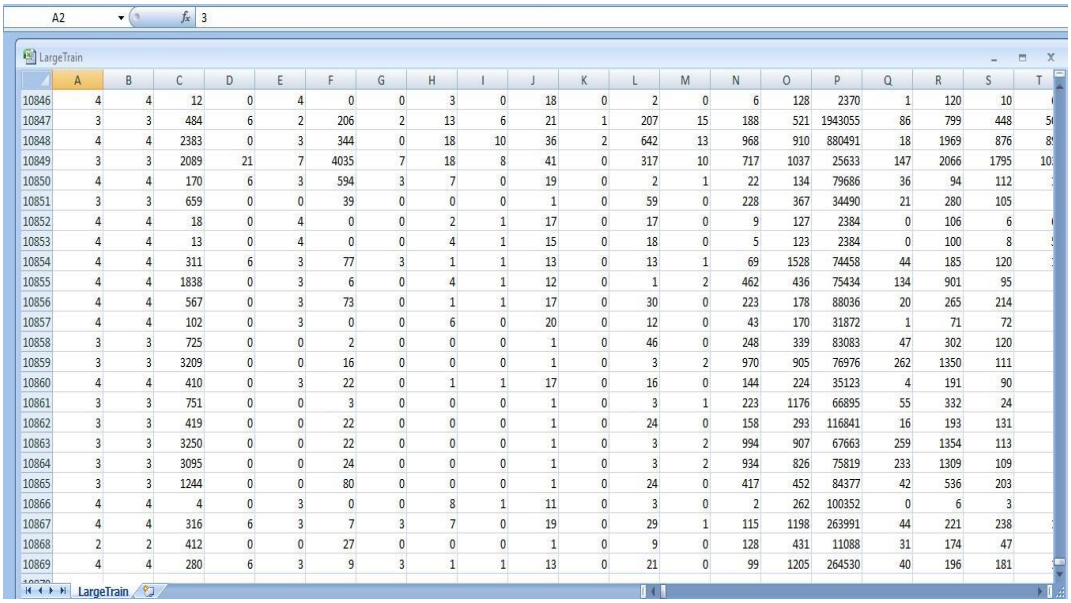


Figure 1: System Architecture



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
10846	4	4	12	0	4	0	0	3	0	18	0	2	0	6	128	2370	1	120	10	
10847	3	3	484	6	2	206	2	13	6	21	1	207	15	188	521	1943055	86	799	448	5
10848	4	4	2383	0	3	344	0	18	10	36	2	642	13	968	910	880491	18	1969	876	8
10849	3	3	2089	21	7	4035	7	18	8	41	0	317	10	717	1037	25633	147	2066	1795	10
10850	4	4	170	6	3	594	3	7	0	19	0	2	1	22	134	79686	36	94	112	
10851	3	3	659	0	0	39	0	0	0	1	0	59	0	228	367	34490	21	280	105	
10852	4	4	18	0	4	0	0	2	1	17	0	17	0	9	127	2384	0	106	6	
10853	4	4	13	0	4	0	0	4	1	15	0	18	0	5	123	2384	0	100	8	
10854	4	4	311	6	3	77	3	1	1	13	0	13	1	69	1528	74458	44	185	120	
10855	4	4	1838	0	3	6	0	4	1	12	0	1	2	462	436	75434	134	901	95	
10856	4	4	567	0	3	73	0	1	1	17	0	30	0	223	178	88036	20	265	214	
10857	4	4	102	0	3	0	0	6	0	20	0	12	0	43	170	31872	1	71	72	
10858	3	3	725	0	0	2	0	0	0	1	0	46	0	248	339	83083	47	302	120	
10859	3	3	3209	0	0	16	0	0	0	1	0	3	2	970	905	76976	262	1350	111	
10860	4	4	410	0	3	22	0	1	1	17	0	16	0	144	224	35123	4	191	90	
10861	3	3	751	0	0	3	0	0	0	1	0	3	1	223	1176	66895	55	332	24	
10862	3	3	419	0	0	22	0	0	0	1	0	24	0	158	293	116841	16	193	131	
10863	3	3	3250	0	0	22	0	0	0	1	0	3	2	994	907	67663	259	1354	113	
10864	3	3	3095	0	0	24	0	0	0	1	0	3	2	934	826	75819	233	1309	109	
10865	3	3	1244	0	0	80	0	0	0	1	0	24	0	417	452	84377	42	536	203	
10866	4	4	4	0	3	0	0	8	1	11	0	3	0	2	262	100352	0	6	3	
10867	4	4	316	6	3	7	3	7	0	19	0	29	1	115	1198	263991	44	221	238	
10868	2	2	412	0	0	27	0	0	0	1	0	9	0	128	431	11088	31	174	47	
10869	4	4	280	6	3	9	3	1	1	13	0	21	0	99	1205	264530	40	196	181	

Figure 2: Dataset Sample

3.2 Data Mining and Classification

3.2.1 Adaptive boost

AdaBoost was used in the context of data mining to accomplish the research's objective. By efficiently merging the predictions of multiple weak learners, it builds a robust classifier that can handle complicated datasets and improve expected performance. The final forecast is established by integrating the guesses of the less proficient learners through weighted voting. The mathematical representation of the is provided below:

$$H\ (x) = sign\ (\sum_{t=1}^{T} \quad \alpha_t\ .\ h_t\ (x))$$

3.2.2 C4.5

The C4.5 method is used in this study due to its dependability and adaptability in building decision trees. It works well for classification tasks in network intrusion detection due to its ability to handle both continuous and categorical variables, deal with missing values, and perform effective feature selection. The resulting decision trees are comprehensible and offer explicit guidelines that control categorization results. By using C4.5, this study guarantees that the final model not only functions well but also provides transparency in the identification and classification of intrusion patterns:

Tree = Buildtree(D, Attributes)

3.2.3 Hybridized Technique

This work combines adaptive data mining approaches with the C4.5 algorithm to improve classification performance and durability. To increase forecast accuracy, the method leverages C4.5's ability to produce interpretable decision trees and combines it with adaptable mining techniques. Due to this integration, the model is better equipped to handle complex and diverse data, enabling more reliable intrusion detection decision-making. To ensure fair and reliable classification results, final predictions are obtained through weighted voting, where each decision tree contributes according to its assigned weight. The mathematical representation of this is shown below:

$$\hat{y} = sign\ (\sum_{i=1}^{m} \quad \alpha_i\ .\ \hat{y}_i(x))$$

For example, let yi(x) represent the decision tree's forecast. The equation above then represents the ultimate forecast of the hybridized technique.

3.2.4 Decision Tree Classifier:

In this study, decision trees are constructed using an algorithmic technique that divides information according to predetermined criteria. The resulting tree structure facilitates methodical and transparent decision-making, making it useful for identifying trends in network intrusion detection. Below is a mathematical depiction of it.

$$(x, Y) = (x_1, x_2, x_3, \ldots\ldots, x_k, Y)$$

The objective variable that this study aims to categorize is known as the dependent variable, denoted as Y. The characteristics that are employed for that categorization, such as x1, x2, x3, etc., make up the vector x.

3.3.1 Dataset Processing

The website, as stated in Section 3.1, provided the dataset used in the development of the system. The dataset was imported into the Python environment using the Pandas library. Figure 3 below displays a sample of the dataset in the used context.

The dataset have the size of (692703 rows, and 79 columns) the following are the variable name for columns of the dataset ([' Destination Port', ' Flow Duration', ' Total Fwd Packets', ' Total Backward Packets', 'Total Length of Fwd Packets', ' Total Length of Bwd Packets', ' Fwd Packet Length Max', ' Fwd Packet Length Min', ' Fwd Packet Length Mean', ' Fwd Packet Length Std', 'Bwd Packet Length Max', ' Bwd Packet Length Min', ' Bwd Packet Length Mean', ' Bwd Packet Length Std', 'Flow Bytes/s', ' Flow Packets/s', ' Flow IAT Mean', ' Flow IAT Std', ' Flow IAT Max', ' Flow IAT Min', 'Fwd IAT Total', ' Fwd IAT Mean', ' Fwd IAT Std', ' Fwd IAT Max', ' Fwd IAT Min', 'Bwd IAT Total', ' Bwd IAT Mean', ' Bwd IAT Std', ' Bwd IAT Max', ' Bwd IAT Min', 'Fwd PSH Flags', ' Bwd PSH Flags', ' Fwd URG Flags', ' Bwd URG Flags', ' Fwd Header Length', ' Bwd Header Length', 'Fwd Packets/s', ' Bwd Packets/s', ' Min Packet Length', ' Max Packet Length', ' Packet Length Mean', ' Packet Length Std', ' Packet Length Variance', 'FIN Flag Count', ' SYN Flag Count', ' RST Flag Count', ' PSH Flag Count', ' ACK Flag Count', ' URG Flag Count', ' CWE Flag Count', ' ECE Flag Count', ' Down/Up Ratio', ' Average Packet Size', ' Avg Fwd Segment Size', ' Avg Bwd Segment Size', ' Fwd Header Length.1', 'Fwd Avg Bytes/Bulk', ' Fwd Avg Packets/Bulk', ' Fwd Avg Bulk Rate', ' Bwd Avg Bytes/Bulk', ' Bwd Avg Packets/Bulk', 'Bwd Avg Bulk Rate', 'Subflow Fwd Packets', ' Subflow Fwd Bytes', ' Subflow Bwd Packets', ' Subflow Bwd Bytes', 'Init_Win_bytes_forward', ' Init_Win_bytes_backward', ' act_data_pkt_fwd', ' min_seg_size_forward', 'Active Mean', ' Active Std', ' Active Max', ' Active Min', 'Idle Mean', ' Idle Std', ' Idle Max', ' Idle Min', ' Label'), where the last column name is the target for in this study. The distribution plot of this is shown in Figure 4.

	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	min_seg_size_forward	Active Mean	Active Std	Active Max	A
0	80	38308	1	1	6	6	6	6	6.000000	0.000000	...	20	0.0	0.0	0	
1	389	479	11	5	172	326	79	0	15.636364	31.449238	...	32	0.0	0.0	0	
2	88	1095	10	6	3150	3150	1575	0	315.000000	632.561635	...	32	0.0	0.0	0	
3	389	15206	17	12	3452	6660	1313	0	203.058823	425.778474	...	32	0.0	0.0	0	
4	88	1092	9	6	3150	3152	1575	0	350.000000	694.509719	...	32	0.0	0.0	0	

rows × 79 columns

Figure 3: Onboarding Dataset

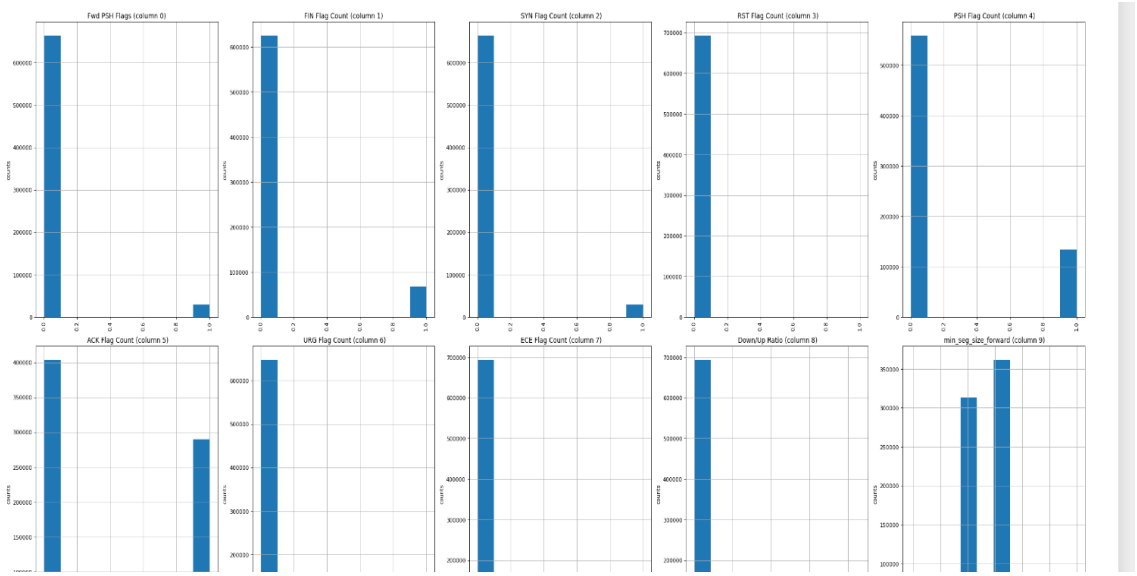


Figure 4: Distribution plot for the dataset

flow Bwd ytes	Init_Win_bytes_forward	Init_Win_bytes_backward	act_data_pkt_fwd	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
6	255	946	0	20	0.0	0.0	0	0	0.0	0.0	0	0	0
326	29200	260	4	32	0.0	0.0	0	0	0.0	0.0	0	0	0
3150	29200	2081	3	32	0.0	0.0	0	0	0.0	0.0	0	0	0
5660	29200	0	10	32	0.0	0.0	0	0	0.0	0.0	0	0	0
3152	29200	2081	2	32	0.0	0.0	0	0	0.0	0.0	0	0	0

Figure 5: Encoded features

3.3.2 Dataset Transformation

To make the category variables in this project (network intrusion detection systems) compatible with the machine learning approach being used, label encoding was employed. Label encoding provides an efficient representation of category data, making it suitable for analyzing large volumes of network traffic data. It also preserves the ordinal connections in the data while streamlining data processing and using less memory.

Machine learning models perform better with numerical data; all the dataset's categorical variables were converted to numbers, as shown in Figure 5 above. This helps the model identify relationships within the dataset.

3.4. Hybridized Technique

AdaBoost and C4.5 decision trees are incorporated in this study's hybridization strategy to improve performance. This method combines multiple C4.5 decision trees, each focused on a distinct portion of the network data, utilizing AdaBoost's ensemble learning feature. Hybrid NIDS reduces false alarms while increasing detection accuracy and flexibility by utilizing both approaches and adjusting to evolving threats. As provided in Figure 6 below, hybridization is a valuable network security method, as it ensures optimal performance through regular evaluation and optimization.

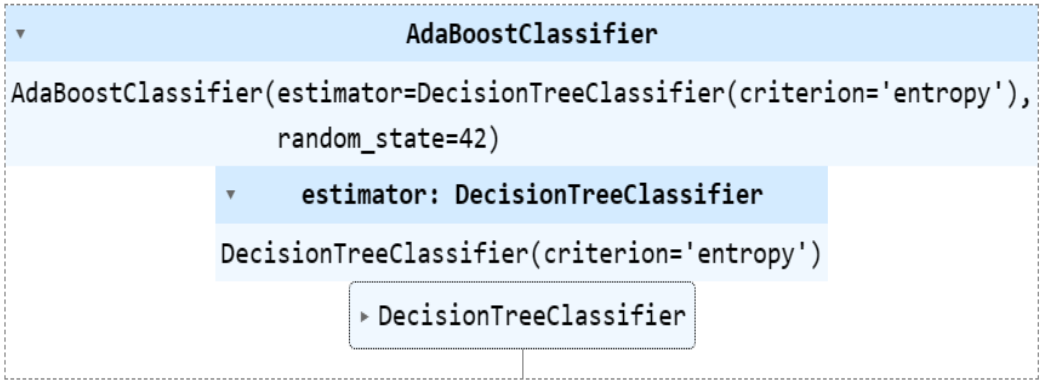


Figure 6: Hybridized Technique

4. Results and Discussion

Following the data pre-processing, data mining was completed. To identify irregularities, patterns, and anticipate security risks in large amounts of network traffic data, network intrusion detection systems (NIDS) rely on data mining. It guarantees scalability and resistance against changing threats while facilitating feature selection, categorization, and prediction. The dataset, which included the target and comprised 638,282 rows and 79 characteristics, was divided into two subsets: the training set (80%) and the test set (20%).

Next, the subsets were separated into labels (y) and features (X). This meant that one set of data was utilized for testing and another for training the data mining algorithms (Adaptive Boost, C4.5, and eventually hybridized methods).

Additionally, the models' performance on training and testing sets was assessed using standard metrics (accuracy, F1 scores, recall, and precision). The dataset was split into two groups, resulting in 500,000 rows for training (80%) and 138,282 rows for testing (20%). But every result was documented, and the model was trained on 500,000 cases. The trained model's performance was then evaluated using the 138282 instances. Figure 6 shows the division of the dataset.

4.1 Classification report for Data mining Algorithms

Table 1: Evaluation Metrics Definition

Metrics	Definition
Precision	The ratio of true positives to the sum of true positives and false positives is known as precision.
Recall	The recall is computed by dividing the total number of TP and FN by the number of TP.
F1 score	The weighted harmonic mean of memory and precision is referred to as F1. The model's predictive performance improves as the F1 score approaches 1.0.
Support	Support refers to the number of actual instances of the class in the selected dataset. It does not distinguish between models; it merely examines the performance evaluation process.

These metrics, as presented in Table 1, are used to assess the effectiveness of a categorization-based machine learning model. The model demonstrates strong accuracy, recall, and F1 score, and is well-supported by the available data. It aids in our understanding of the overall performance of our trained model.

4.5 Result for NIDS

4.5.1 Result for NIDS for Training Set

The performance attributes of the three data mining techniques (Adaptive Boost, C4.5, and a hybridized approach) used in the study are shown in Table 2. The measures include False Positives (FP), False Negatives (FN), True Negatives (TN), Accuracy, Precision, Recall, F1 Score, and True Positives (TP). With 317,185 true positives and 287 false positives, Adaptive Boost's accuracy rating is 99.7%. With a recall of 99.5% and a precision of 99.8%, it is highly successful in identifying most attacks while keeping a low false positive rate. The F1 score of 99.7% indicates that the model is very balanced and successful. C4.5 attains an even higher accuracy of 99.9% with 317,938 true positives and only 5 false positives. It tends to be nearly flawless at distinguishing between benign and attack scenarios, with a precision and recall of 99.9%. The F1 score of 99.9% indicates nearly flawless performance in both recall and precision. Hybridized also performs almost perfectly, with an accuracy of 99.9%, 317,937 true positives, and 4 false positives. With a precision and recall of 99.9%, it functions quite similarly to C4.5. The F1 score of 99.9% attests to its remarkable combination of recall and accuracy.

Figure 7 provide an adequate summary of the results, indicating that both C4.5 and the hybridized approach exhibit nearly flawless performance metrics, making them highly effective NIDS tools. Adaptive Boost continues to function exceptionally well, even if its performance is marginally lower than that of the other two. These techniques establish their ability to deliver solid and dependable network security by reducing false positives and negatives and efficiently identifying attack vectors.

Table 2: Evaluation of the three models' performances on the train set

Data Mining Techniques	TP	FP	FN	TN	Accuracy	Precision	Recall	F1 Score
Adaptive Boost	317185	287	778	181750	99.7%	99.8%	99.5%	99.7%
C4.5	317938	5	25	182032	99.9%	99.9%	99.9%	99.9%
Hybridized Model	317937	4	26	182033	99.9%	99.9%	99.9%	99.9%

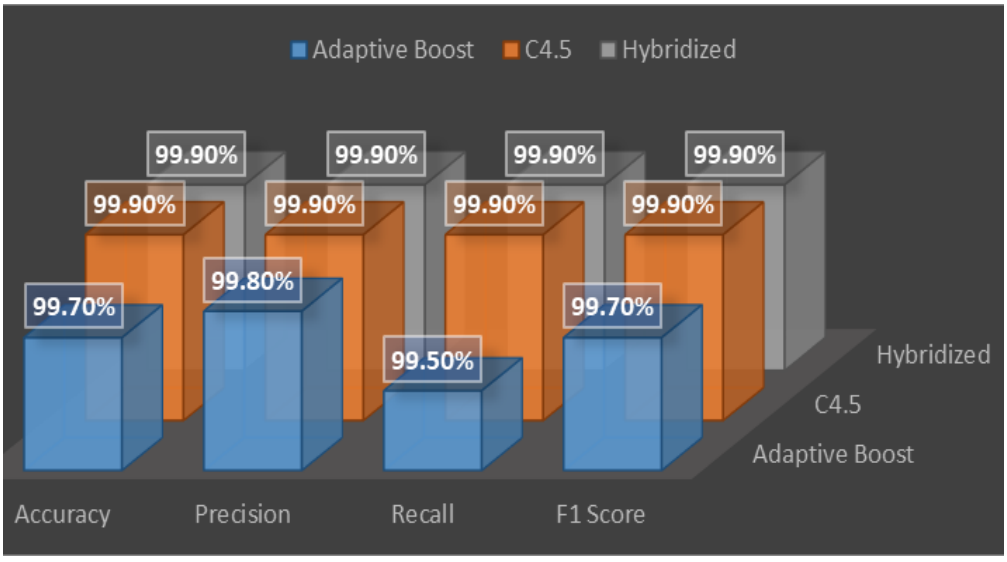


Figure 7: Graphical representation of the result for the Training dataset

4.5.1 Result for NIDS for Test Set

Table 3 illustrates the performance metrics for the three distinct data mining methods used in this research training test: C4.5, Adaptive Boost, and a hybrid approach. Among the measures are False Positives (FP), True Negatives (TN), True Negatives (TP), False Negatives (FN), Accuracy, Precision, Recall, and True Negatives (TN). Adaptive Boost achieves 99.7% accuracy with 87,707 true positives and 82 false positives. It is highly effective in detecting the majority of attacks due to its low false positive rate and remarkable 99.8% recall and precision rate.

The F1 score of 99.6% indicates a highly balanced and successful model. C4.5 produces 87,911 true positives and 26 false positives with a high accuracy of 99%. With a precision and recall of 99%, it seems to be able to distinguish between benign cases and assaults. However, compared to Adaptive Boost, its F1 score of 97% indicates a slight decline in performance, which may be attributed to a slightly higher false negative rate. In a similar vein, Hybridized shows promising results, with 87,913 true positives, 24 false positives, and an accuracy rate of 99%. It operates with 99% recall and precision, which is extremely close to C4.5. The F1's outstanding memory and accurate balancing are attested to by its 99% grade.

In the final analysis, Adaptive Boost yields the highest overall accuracy and F1 score; however, all three approaches demonstrate good efficacy in identifying network intrusions with minimal false positives and negatives. Both C4.5 and the hybridized approach perform remarkably well, making them excellent options for robust network security, particularly as they both have fewer false positives, with the hybrid technique excelling, as illustrated in Figure 8.

Table 3: Evaluation of the three models' performances on the test set

Data Mining Techniques	TP	FN	FN	TN	Accuracy	Precision	Recall	F1 Score
Adaptive Boost	87707	82	230	50263	99.7%	99.8%	99.5%	99.6%
C4.5	87911	26	12	50333	99%	99%	99%	97%
Hybridized Model	87913	24	14	50331	99%	99%	99%	99%

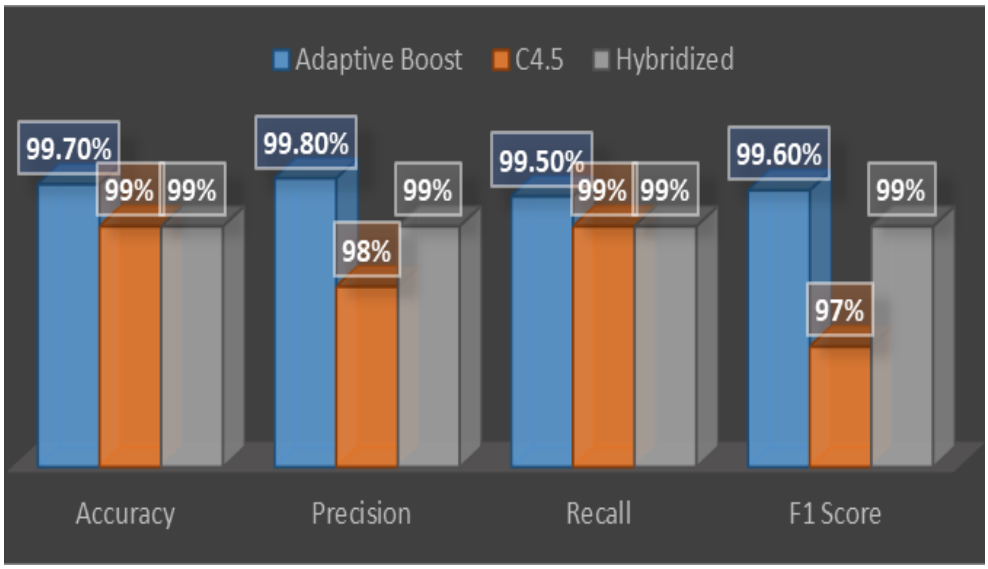


Figure 8: Graphical representation of the result for the Testing dataset.

4. Conclusion

Network intrusion detection systems (NIDS) are very significant for securing against any malicious activity on the network. To improve NIDS's effectiveness, two very powerful approaches have been combined: "C4.5 Decision Trees" and "Adaptive Data Mining" techniques. "C4.5 Decision Trees" are very effective for analyzing reasoning behind intrusion detection because they are capable of providing interpretable classification trees. Based on generating strong decision trees and pointing to significant attributes, "C4.5 Decision Trees" are helpful for identifying possible threats against network traffic data. Moreover, for making NIDS flexible to learn new invasions at any point of time and remain effective against new invasions continuously, this study adopts adaptive data mining techniques to learn new data at any point of time and adapt to new invasion strategies developed by hackers continuously.

The adaptability of NIDS guarantees NIDS's high effectiveness for identifying threats against network traffic data at all times and forever because NIDS is very effective for identifying any kind of network invasions at any point of time because adaptive data mining techniques are adopted to learn at any point of time to remain very effective against new network invasions. A reliable intrusion detection system that can effectively identify a variety of network intrusions while reducing false alarms by integrating the advantages of both strategies. The hybridized approach in this study provides a comprehensive network security solution, ensuring the integrity and confidentiality of network resources while offering real-time defense against threats. To sum up, the research goal was to enhance network intrusion detection by combining C4.5 decision trees with adaptive data mining techniques. This hybrid method leverages the interpretability of C4.5 and the flexibility of data mining to develop a dependable intrusion detection system that can accurately identify a range of incursions. Through the integration of various methods, this study offers a comprehensive approach to network security, bolstering ongoing efforts to protect networks against evolving cyber threats.

Declarations

Ethics approval: Not applicable
Consent for publication: Not applicable
Competing interests: No competing interests in the manuscripts
Funding: The study was self-funded

References

[1] Almomani, A. (2020). Machine learning-based anomaly detection for network security. *Journal of Information Security and Applications*, *52*, 102467. https://doi.org/10.1016/j.jisa.2020.102467

[2] Ali, M., Khan, S., Rehman, A., & Hussain, T. (2025). Feature selection challenges in anomaly-based intrusion detection systems. *IEEE Transactions on Information Forensics and Security*. https://ieeexplore.ieee.org/document/10567890

[3] Ali, A. H., Charfeddine, M., Ammar, B., Ben Hamed, B., Albalwy, F., Alqarafi, A., & Hussain, A. (2024). Unveiling machine learning strategies and considerations in intrusion detection systems: A comprehensive survey. *Frontiers in Computer Science*, *6*. https://doi.org/10.3389/fcomp.2024.1387354

[4] Al-Turjman, F., & Zahmatkesh, H. (2024). Network intrusion detection systems: A comprehensive survey. *Journal of Network and Computer Applications*. https://www.sciencedirect.com/science/article/pii/S108480452400001X

[5] Falowo, G., Olorunfemi, B. O., Adeniyi, A. E., Abosede, O. B., & Ogbuju, E. (2025). Machine learning-based detection and classification of SQL injection attacks using a stacking ensemble model. In *2025 Conference on Information Communications Technology and Society (ICTAS)* (pp. 1–6). IEEE. https://doi.org/10.1109/ICTAS64866.2025.11155704

[6] Liu, Y., Gao, J., & Hu, X. (2021). Machine learning approaches for network intrusion detection: A review. *ACM Computing Surveys*, *54*(3), 1–36. https://doi.org/10.1145/3439720

[7] Mane, S., & Rao, P. (2021). Deep learning techniques for intrusion detection in network systems. *International Journal of Advanced Computer Science and Applications*, *12*(5), 123–130. https://thesai.org/Downloads/Volume12N

o5/Paper_16-Deep_Learning_Techniques_for_Intrusion_Detection.pdf

[8] Qazi, E. U. H., Faheem, M. H., & Zia, T. (2023). HDLNIDS: Hybrid deep-learning-based network intrusion detection system. *Applied Sciences*, *13*(8). https://doi.org/10.3390/app13084921

[9] Sajid, M., Malik, K. R., Almogren, A., Malik, T. S., Khan, A. H., Tanveer, J., & Rehman, A. U. (2024). Enhancing intrusion detection: A hybrid machine and deep learning approach. *Journal of Cloud Computing*, *13*, Article 123. https://journalofcloudcomputing.springeropen.com/articles/10.1186/s13677-024-00685-x

[10] Sarhan, M., Layeghy, S., & Portmann, M. (2022). Signature-based intrusion detection systems: Techniques and challenges. *Computers & Security*, *112*, 102510. https://doi.org/10.1016/j.cose.2021.102510

[11] Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data*, *11*. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00886-w