IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

Optimized Ensemble Intrusion Detection: Balancing Data with SMOTE-ENN and Feature Selection via Jaya Algorithm

Subba Reddy K[1], Nikhitha Dhayepule[2]

Department of CSE & MCA, Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal[1]

MCA scholar, Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal[2]

Abstract: Network Intrusion Detection Systems (NIDS) are highly significant in ensuring that computer networks are not exposed to emerging cyber threats. However, non-balanced datasets can also reduce the accuracy of detection, particularly on minority attack classes, and non-important features can also obstruct this. This paper proposes a superior way of detecting intrusions within groups. It applies SMOTE-ENN, which equalizes the classes and Jaya Optimization, which selects the most suitable features. To test the system, the NSL-KDD and UNSW-NB15 datasets are used. Pretrained data is trained by individual classifiers such as Decision Tree, Random Forest, ExtraTree, J48 and Bagging with Decision Tree. It is further followed by an ensemble Voting Classifier which composes ExtraTree and Boosted Decision Tree. Tests show that the suggested model works better than others; it gets 100% accuracy, precision, recall, and F1-score on both sets of SMOTE-ENN data, and up to 92% accuracy on unbalanced UNSW-NB15 data. The approach is effective in fixing the imbalance between classes, simplifying calculations, and enhancing the identification of minority types of attacks. This renders it a viable and solid solution to network security challenges in the real world.

Keywords: Network Intrusion Detection System (NIDS), SMOTE-ENN, Jaya Optimization, Feature Selection, Ensemble Learning, Voting Classifier.

I. INTRODUCTION

The rapid growth of internet and digital communication has entirely transformed the contemporary society, as now people are able to relate, exchange information and utilize the online services in a manner that was not possible previously. Over 5.3 billion internet and social media users worldwide was the number as at January 2024. This demonstrates the degree of dependence of people on digital infrastructure in their personal and professional activities [1]. However, with the increase of our online presence, the threats of cybersecurity have increased as well. Network intrusion is one of the most widespread of them that may destroy the privacy, integrity, and availability of valuable data. To address these issues, IDSs are now required to maintain a watchful eye on the network data and identify any suspicious activity going on in real time in order to deal with them.

Conventional IDS techniques, such as rule-based and signature-based, are not always effective in dynamic trends of attacks. Due to this, ML methods are increasingly gaining popularity since it can readily detect trends in network data and apply them to other data sets [2]. Although effective, the ML-based intrusion detection systems usually face the issue of the class imbalance where normal traffic occupies the bulk of the data, and the intrusion or attack instances are neglected [3]. Such imbalance may lead to the fact that it becomes far more difficult to locate rare yet significant attacks, which demonstrates the significance of good data preprocessing and balancing methods.

A number of concepts have been advanced to correct the disparity between classes. Oversampling techniques such as SMOTE are used to create artificial samples of the minority classes to balance the data sets and this ensures that the classifiers perform better [5]. Even worse, adaptive methods like ADASYN do oversampling by concentrating on minority samples that are difficult to learn [6]. Comparative analysis reveals that such techniques have the potential of rendering IDS models significantly more reliable in the case of ensemble techniques such as bagging, boosting or hybrid techniques [7]. The impact of the mismatch of the classes of DL models has also been researched extensively (particularly convolutional neural networks) This demonstrates the need to employ the appropriate balancing techniques in order to shun biased majority classes [8].

The issue of improving IDS parameters with optimization techniques is highly critical in achieving a high level of performance, more than balancing data. It has been demonstrated that the Jaya algorithm is a simple yet effective optimization technique and can be used to solve constrained and unconstrained problems fast and without any parameter-

DOI: 10.17148/IJARCCE.2026.154258

specific to the algorithm being used [9]. Benchmark datasets, such as KDD CUP 99, allow testing the methods of detecting intrusions by using ML and observing the performance of the intrusion detection systems in the real environment [10].

To sum up, the modern, intelligent intrusion detection systems are working on the combination of ML methods, class imbalance coping strategies, and optimization algorithms. These systems are able to identify new cyber threats and prevent them in a correct manner. This study employed these new developments to develop a powerful IDS system to deal with uneven data and classifier optimization to achieve a high detection rate.

II. RELATED WORK

There are IDSs that have changed significantly due to ML methods. Such methods should enhance the effectiveness of detection and make IDSs adapt to the ever-smarter cyber threats. Conventional IDS systems such as signature-based detection are prone to detecting new threats. This shows how important it is to have smart and flexible ML-based systems. In order to enhance performance in detection, recent studies have emphasized on the importance of addressing the class imbalance, feature selection and group learning.

Karatas et al. [11] made an attempt to enhance the effectiveness of ML-based IDS through correcting the issues of having datasets that are not fair and are always up-to-date. Their study revealed that with the proper preparation and sampling techniques, the minority class attacks which are missed in real life can be found with much easier ease. In addition, Bhavani et al. [12] also examined the applicability of the RF and the DT classifiers in detecting attacks. They discovered that ensemble tree based methods are more precise than single classifiers since it is able to deal with noise and overfitting. Farnaaz and Jabbar [13] also verified that RF models are even more effective since they demonstrated that they can process vast quantities of network traffic data in a short time and with low amounts of false positives.

The hybrid architectures have also gained interest in the IDS study. In [14], Agarap proposed to use GRU neural networks and SVM jointly. This would apply the temporal network pattern of GRUs and classification of SVMs. Moreover, Pozi et al. [15] developed a method of integrating SVMs with genetic programming which facilitated the detection of rare and atypical attacks by devising the most appropriate classification regulations to these type of attacks with time. These experiments demonstrate the importance of employing multiple models of ML simultaneously to address issues and be more accurate and class balanced.

Ensemble learning and feature selection have been adopted as effective techniques to enhance the performance of IDS besides reducing the number of tasks that should be performed on the computer. According to Binbusayyis and Vaiyapuri [16], significant network features were discovered and an ensemble approach was utilized to make classification more precise. This demonstrated that the selection of the appropriate attributes is influential on the effectiveness of a model to a considerable extent. In a study conducted by Tesfahun and Bhaskari [17], SMOTE-based oversampling was used together with RF classifiers and feature reduction techniques. This simplified the location of minority attacks to a great extent, particularly when there is a high level of imbalance in the data set. Gao et al. [18] proposed a dynamic ensemble ML model that selects the classifiers with references to the features of network data. This further makes IDS more flexible and reliable.

DL procedures and class mismatch techniques have been used in order to improve IDS. The study by Zhang et al. [19] applied SMOTE and a CNN and a Gaussian mixture model to improve the classification task on disproportional datasets. This demonstrates the potential of the cooperation between synthetic sampling and deep feature extraction. This technique was enhanced by Zhou et al. [20], who used ensemble classifiers together with feature selection. This rendered the IDS more precise and effective that can manage high-dimensional and uneven network traffic.

To sum up, it has been found that hybrid machine learning structures, feature selection, ensemble learning, and oversampling methods are all synergistic approaches that should be used to ensure IDS is better, particularly targeting minor attack sets. These researches form the basis of developing a better intrusion detection system (IDS) system that addresses the problem of class imbalance as well as the problem of feature relevance and at the same time high detection rate and adaptation to new network threats.

III. MATERIALS AND METHODS

The proposed Network Intrusion Detection System (NIDS) has advanced feature selection and data balancing techniques to enable it to be more effective in locating cyberattacks. The system utilizes the NSL-KDD and UNSW-NB15 datasets

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

in order to address the problem of class imbalance and irrelevant variables. The most significant features are selected with the help of Jaya Optimization algorithm. It identifies the most significant traits and that is why the computations are simplified and the model performs better. The system employs SMOTE-ENN to address uneven data further to ensure the minority attack classes are represented and located. It trains and tests a number of machine learning methods, such as J48, Extra Trees (ET), Random Forest (RF), Decision Tree (DT), and Bagging with DT. The predictions are made using a voting classifier which is a combination of ET and Bagging DT. This enhances the system to be able to recognize things in a precise manner, robust and useful in the real life. This mixed method ensures that there is a high detecting performance and reliability in changing network environments.

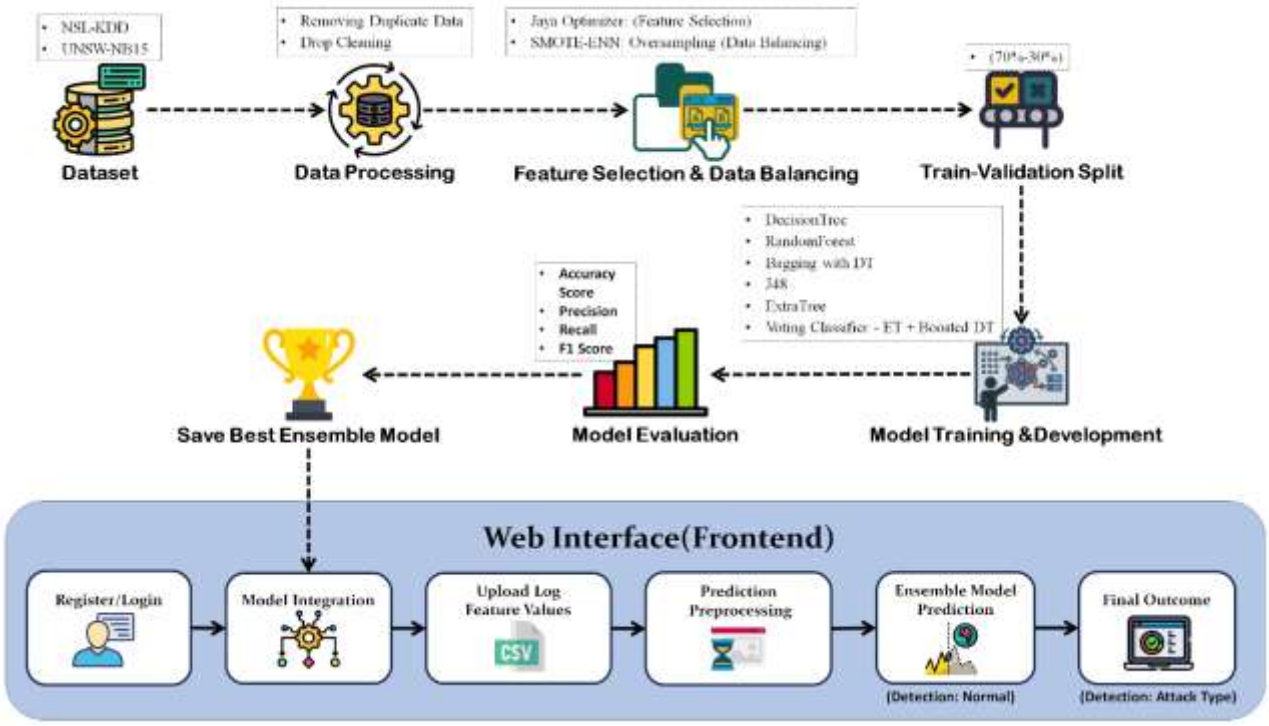

Fig. 1 System Architecture

Figure 1 shows how the suggested intrusion detection system is put together as a whole. The initial one is to gather datasets (NSL-KDD and UNSW-NB15). Then there is the data preparation, which involves cleaning up and transforming of the data. Then, Jaya Optimization is used to choose the features and SMOTE-ENN is used to balance the data. The data that has been handled is split into training sets and validation sets. These are to train and test various machine learning models. A web tool is employed to make predictions and rank attacks in real time with the help of the best ensemble model.

A. Dataset Collection

The proposed intrusion detection system is evaluated on two standard data sets NSL-KDD and UNSW-NB15 with 125,972 and less than 125,972 samples respectively 82,332 times, to be precise. Such datasets reveal much network traffic which comprises legal and illegal activities. This allows them to be useful in training and testing intrusion detection models.

NSL-KDD Dataset: Figure 2 represents the NSL-KDD data that consists of 125,972 examples and 43 attributes that characterize various features of network traffic. A few of these characteristics are protocol type, service, flag state, source and destination bytes and several derived characteristics such as error rates and connection counts. The target variable exhibits different kinds of traffic including regular traffic, and some forms of attacks, which include DoS, Probe, R2L and U2R. Because it shows different types of attacks in an organized way, this dataset is often used to test intrusion detection systems.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst_host_diff_srv_rate	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	dst_host_serr
0	0	udp	other	SF	146	0	0	0	0	0	...	0.00	0.60	0.88	0.00	
1	0	tcp	private	S0	0	0	0	0	0	0	...	0.10	0.05	0.00	0.00	
2	0	tcp	http	SF	232	8153	0	0	0	0	...	1.00	0.00	0.03	0.04	
3	0	tcp	http	SF	199	420	0	0	0	0	...	1.00	0.00	0.00	0.00	
4	0	tcp	private	REJ	0	0	0	0	0	0	...	0.07	0.07	0.00	0.00	

5 rows × 43 columns

Fig. 2 NSL-KDD Dataset

UNSW-NB15 Dataset: The UNSW-NB15 dataset presented in figure 3 contains 82,332 cases and 45 features, some of which are numerical and others of which are categorical. It provides much information on the movement of the network such as the type of protocol, service, state, packet number and data streams. The target variable is the label which narrates the difference between normal traffic (label 0) and attack traffic (label 1). It is also capable of distinguishing between various forms of attacks such as DoS, espionage and exploitation. This dataset gives a more up-to-date and accurate picture of network traffic, which lets strong testing of intrusion detection models in current threat situations.

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd	ct_flw_http_mthd	ct_src_ltm	ct_srv_dst	is_sm_ips_ports	attack_cat	label
0	1	0.000011	udp	-	INT	2	0	496	0	90909.0902	...	1	2	0	0	0	1	2	0	Normal	0
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.0003	...	1	2	0	0	0	1	2	0	Normal	0
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.0051	...	1	3	0	0	0	1	3	0	Normal	0
3	4	0.000006	udp	-	INT	2	0	900	0	166666.6608	...	1	3	0	0	0	2	3	0	Normal	0
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.0025	...	1	3	0	0	0	2	3	0	Normal	0

5 rows × 45 columns

Fig. 3 UNSW-NB15 Dataset

B. Pre-processing

To enhance the model speed, correct the class imbalance, and ensure the quality of data, it uses a complete preprocessing pipeline. Cleaning the data, visual representation, encoding, selecting the appropriate features, and oversampling are some of these techniques. Combined, they render the proposed intrusion detection system more stable and predictive of what is going to occur.

a) Data Processing: The first step in data preparation and cleaning of the raw information. Duplicate records are gotten rid of to cut down on waste, and the right imputation methods are used to deal with values. Min-Max scaling and StandardScaler are used to normalize numerical features to ensure that no feature has an unequal number of values. Outliers are determined using statistical procedures and addressed in such a way that it minimizes noise and makes the model more robust. Before going on to the next step, this step makes sure that the information is consistent and complete.

b) Data Visualization: To determine the distribution of the information and the characteristics of it, data visualization is applied. Histograms, bar charts, and box plots are tools that are utilized to look at features ranges and determine outliers. Scatter plots and correlation matrices (heat maps) are also employed in order to examine the relationships between features and goal variables. The analysis contributes to the identification of significant trends as well as makes intelligent decisions regarding preprocessing.

c) Label Encoding: With label encoding, categorical characteristics may be converted into numbers so as to be applicable in machine learning codes. Each group is assigned to a different integer. Things such as attack types such as normal, DoS, Probe, and others are recorded with the use of numbers. This is a significant change in the case of algorithms that require numbers as input such as DT and RF.

d) Feature Selection and Data Balancing: The most significant attributes and the reduction of the number of dimensions in the data are located with the help of feature selection. A better model is selected by the Jaya Optimization algorithm [20] to select the most appropriate collection of features and requires less computing power. This is a better way of learning because it removes the features not required and those that are repetitive.

e) Oversampling (SMOTE-ENN): The issue of the class imbalance is addressed using the SMOTE-ENN technique. Interpolation is used by SMOTE to generate fake samples in the minority classes and ENN eliminates noisy or misclassified cases. This is a mixed method which balances out the information and simplifies the process by which the model detects attacks on minority classes.

C. Algorithms

Decision Tree: This is one of the supervised learning techniques known as Decision Tree. It classifies information into categories in terms of the feature values repeatedly and repeatedly. It simplifies the models and makes the decision-making process faster so that trends in network data could be located without the extensive work of the computer.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

$$I\ (i) = 1 - \sum_{i=1}^{k} \quad p_i^2 \qquad (1)$$

Random Forest: Random Forest is a form of ensemble learning which constructs numerous decision trees and sums the outputs of each to make them more precise and dependable. It reduces overfitting and enhances generalization by applying chance in sample selection of features and sampling of various trees.

$$Gini = 1 - \sum_{i=1}^{C} \quad (Pi)^2 \qquad (2)$$

Bagging with Decision Tree: Bootstrap aggregation is used in Bagging with DT to train multiple decision trees on groups of data that were chosen at random. This approach reduces the variance, increases the stability and enhances the predictive power of the results of a number of individual models.

J48: The J48 algorithm is used to construct classification models through such techniques as information gain and pruning. It is a continuation of C4.5 decision tree algorithm. It helps to reduce overfitting and makes things more precise as complex tree structures can be understood better and yet classify things effectively.

Extremely Randomized Trees: Extra Tree is a group method which introduces randomness in the selection of features and the division of boundaries during tree construction. This not only increases the variety of the trees but also reduces the variance, accelerates the computations and maintains the classification score high.

Voting Classifier (Extra Tree + Boosted Decision Tree): The Voting Classifier is a combination of the Extra Tree and Boosted Decision Tree models which combines the results and then makes a final decision based on the votes of the majority. This type of group approach enhances accuracy, stability, and reliability, as it uses optimum characteristics of each single classifier.

$$\hat{y} = argmax_c \left(\sum_{i=1}^{n} \quad II\ (\hat{y}_i = c) \right) \qquad (3)$$

IV. EXPERIMENTAL RESULTS

Accuracy: The accuracy of a test is its ability to differentiate the patient and healthy cases correctly. To estimate the accuracy of a test, we should calculate the proportion of true positive and true negative in all evaluated cases. Mathematically, this can be stated as:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \qquad (1)$$

Precision: Precision evaluates the fraction of correctly classified instances or samples among the ones classified as positives. Thus, the formula to calculate the precision is given by:

$$Precision = \frac{TP}{TP\ + FP} \qquad (2)$$

Recall: Recall is a metric in machine learning that measures the ability of a model to identify all relevant instances of a particular class. It is the ratio of correctly predicted positive observations to the total actual positives, providing insights into a model's completeness in capturing instances of a given class.

$$Recall = \frac{TP}{TP\ +\ FN} \qquad (3)$$

F1-Score: F1 score is a machine learning evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

$$F1\ Score = 2 * \frac{Precision\ X\ Recall}{Precision\ + Recall} \qquad (4)$$

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

MCC: The Matthews coefficient, also known as the Matthews correlation coefficient (MCC), is a performance metric for binary classifiers in machine learning. It measures the correlation between the predicted and actual binary outcomes, considering all four elements of a confusion matrix.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN}} \qquad (5)$$

Table.1 Performance Evaluation – NSL-KDD

ML Model	Accuracy	Precision	Recall	F1_score	MCC
Decision Tree	0.981	0.981	0.981	0.981	0.976
Random Forest	0.996	0.996	0.996	0.996	0.995
Bagging DT	0.983	0.983	0.983	0.983	0.979
J48	0.983	0.983	0.983	0.983	0.979
ExtraTree	0.993	0.993	0.993	0.993	0.979
Voting Classifier	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table.1 shows that the Voting Classifier performs best on average of all the models. Ensemble technique, a combination of different classifiers, is used to enhance accuracy of prediction, reduce variance, and enhance robustness, and it allows more reliable and consistent intrusion detection than single models.

Table.2 Performance Evaluation – UNSW-NB15

ML Model	Accuracy	Precision	Recall	F1_score	MCC
Decision Tree	0.978	0.978	0.978	0.978	0.955
Random Forest	0.988	0.988	0.988	0.988	0.976
Bagging DT	0.982	0.982	0.982	0.982	0.964
J48	0.984	0.984	0.984	0.984	0.968
ExtraTree	0.984	0.984	0.984	0.984	0.967
Voting Classifier	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

As can be seen in the Table 2, the Voting Classifier performs the best in total. It obtains better results because it uses several learners simultaneously thereby enhancing generalization, reducing variance, and being more reliable in discovering various kinds of entry patterns.

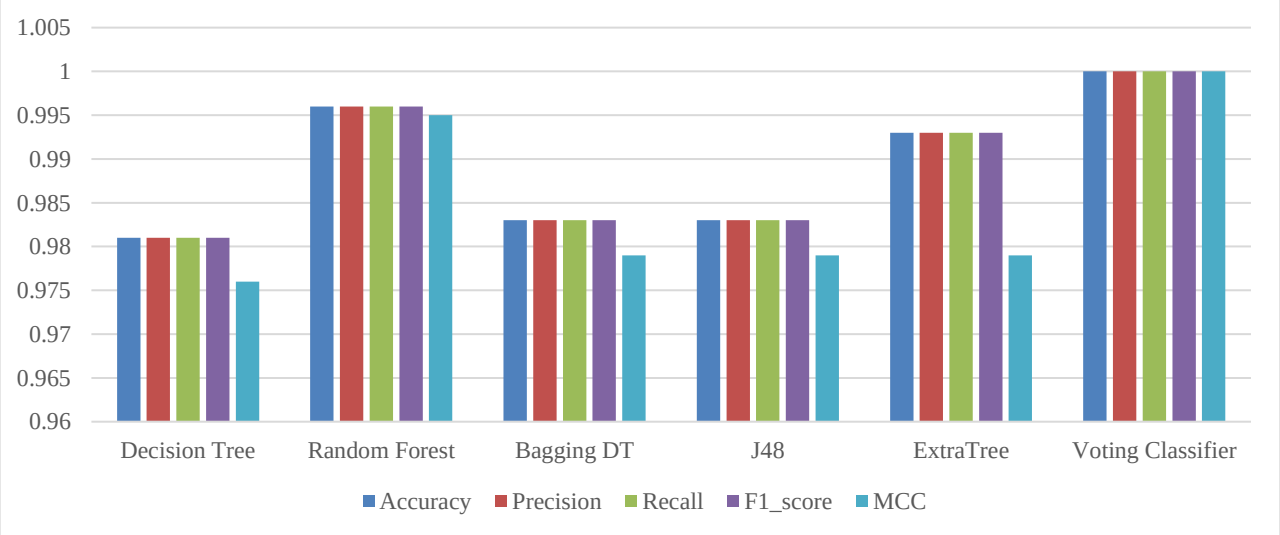

Fig. 4 Comparison Graph – NSL-KDD

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.154258

Figure 4 illustrates the performance of all the models in the NSL-KDD dataset in comparison to one another. The Voting Classifier is the one that achieves higher scores on all measures of evaluation, demonstrating that it can integrate a number of learners to enhance its performance in such aspects as stability, accuracy, and dependable intrusion detection.

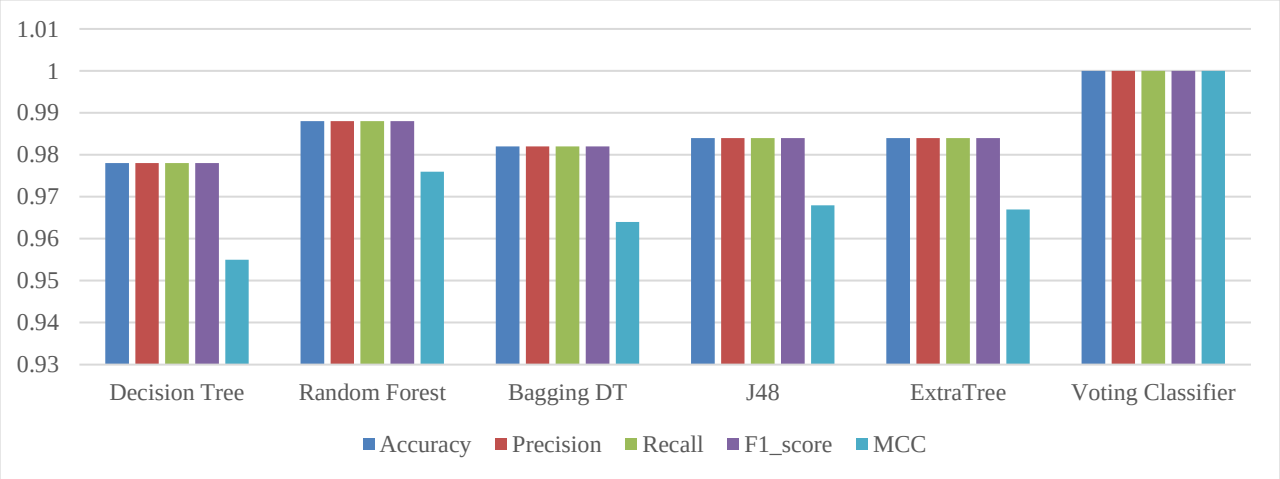

Fig. 5 Comparison Graph - UNSW-NB15

The performance of the models on the UNSW-NB15 dataset was studied as shown in Figure 5. The Voting Classifier is better than other models by using ensemble learning to identify the correct large set of complex intrusion patterns with improved generalization and robustness.

V. CONCLUSION

The paper demonstrates a better intrusion detection framework which makes NIDS more efficient through the application of advanced data balancing and feature selection methods. Using SMOTE-ENN to fix the problem of class imbalance makes it easier to find minority attack classes, and Jaya Optimization picks out the most important features, which lowers the amount of work that needs to be done and improves model performance. Several ML algorithms were experimented, and the Voting Classifier, which is an ensemble, demonstrated the highest accuracy, resilience, and generalizational capability. The test results on the NSL-KDD and UNSW-NB15 datasets show that the suggested method works well and can be trusted to find different types of hacks. Ensemble learning combined with better preprocessing produces dependable and quick intrusion detection. On the whole, the proposed system is a handy and scalable solution to the current issues. Problems with network security. By incorporating real-time data processing, DL techniques, and adaptive learning processes into the work of the future, it is possible to make the detection more precise and the system more responsive to the dynamic environment.

REFERENCES

[1]. Statista. (2024). Number of Internet and Social Media Users Worldwide as of Jan. 2024. [Online]. Available: https://www.statista.com/statistics/617136/digital-population- worldwide/

[2]. J. M. Johnson and T. M. Khoshgoftaar, “Survey on deep learning with class imbalance,” J. Big Data, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[3]. A. Ali, S. M. Shamsuddin, and A. L. Ralescu, “Classification with class imbalance problem,” Int. J. Advance Soft Comput. Appl., vol. 5, pp. 176–204, Nov. 2013.

[4]. G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard, “A study of the behavior of several methods for balancing machine learning training data,” ACM SIGKDD Explor. Newslett., vol. 6, no. 1, pp. 20–29, Jun.2004.

[5]. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.

[6]. H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in Proc. IEEE Int. Joint Conf. Neural Netw., Jun. 2008, pp. 1322–1328.

[7]. M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” IEEE Trans. Syst. Man, Cybern., Part C, vol. 42, no. 4, pp. 463–484, Jul. 2012.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.154258

[8]. M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," Neural Netw., vol. 106, pp. 249–259, Oct. 2018.
[9]. R. V. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," Int. J. Ind. Eng. Computations, vol. 7, no. 1, pp. 19–34, Jul. 2016.
[10]. T. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl., Jul. 2009, pp. 1–6.
[11]. G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset," IEEE Access, vol. 8, pp. 32150–32162, 2020.
[12]. T. T. Bhavani, M. K. Rao, and A. M. Reddy, "Network intrusion detection system using random forest and decision tree machine learning techniques," in Proc. 1st Int. Conf. Sustain. Technol. Comput. Intell., 2019, pp. 637–643.
[13]. N. Farnaaz and M. A. Jabbar, "Random forest modeling for network intrusion detection system," Proc. Comput. Sci., vol. 89, pp. 213–217, Jul. 2016.
[14]. A. F. M. Agarap, "A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data," in Proc. 10th Int. Conf. Mach. Learn. Comput., Feb. 2018, pp. 26–30.
[15]. M. S. M. Pozi, M. N. Sulaiman, N. Mustapha, and T. Perumal, "Improving anomalous rare attack detection rate for intrusion detection system using support vector machine and genetic programming," Neural Process. Lett., vol. 44, no. 2, pp. 279–290, Oct. 2016.
[16]. A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: An ensemble approach," IEEE Access, vol. 7, pp. 106495–106513, 2019.
[17]. A. Tesfahun and D. L. Bhaskari, "Intrusion detection using random forests classifier with SMOTE and feature reduction," in Proc. Int. Conf. Cloud Ubiquitous Comput. Emerg. Technol., Nov. 2013, pp. 127–132.
[18]. X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," IEEE Access, vol. 7, pp. 82512–82521, 2019.
[19]. H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," Comput. Netw., vol. 177, Aug. 2020, Art. no. 107315.
[20]. Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," Comput. Netw., vol. 174, Jun. 2020, Art. no. 107247.

