International Journal of Informatics and Computation (IJICOM)
Vol. 8, No. 1, 2026
ISSN: 2685-8711, E-ISSN: 2714-5263 DOI: 10.35842/ijicom

Performance Evaluation of Decision Tree and Random Forest Algorithms for IDS

Sarah Anjani[1], Encik Yoega Renaldi[2], Feby Charlos[3], Maisan Dewi Puspa Khairani[4]

Abstract
Increasing complexity and frequency of cyberattacks pose a serious threat to modern network infrastructure, making intrusion detection systems (IDS) crucial for maintaining cybersecurity. Conventional IDSs often struggle to identify novel and sophisticated attack patterns, necessitating an adaptive machine learning approach. This study evaluates and compares the performance of Random Forest and Decision Tree algorithms for network intrusion detection using the KDD Cup 99 dataset. This dataset contains both normal network traffic and various categories of cyberattacks, making it suitable for IDS evaluation. The proposed methodology consists of three stages: data preprocessing, model training, and performance evaluation. Model performance is assessed using accuracy, precision, recall, and F1-score metrics. Experimental results show that RF outperforms DT in most evaluation measures. RF achieves an accuracy of 0.88, a precision of 0.98, a recall of 0.74, and an F1-score of 0.84, while DT achieves an accuracy of 0.82, a precision of 0.80, a recall of 0.80, and an F1-score of 0.80. Furthermore, RF demonstrated better generalization capabilities when handling imbalanced data. These findings demonstrate that ensemble-based methods provide a robust and reliable solution for developing accurate IDSs and improving overall network security performance.

Keywords:
Intrusion Detection System, RF, DT, Network Security

1. Introduction

The quick development of computer networking technology has raised cybersecurity concerns considerably in a number of industries. Attacks like Denial of Service (DoS), probing, and illegal access have become more common and sophisticated as a result of increased network connectivity. These attacks can disrupt network systems and take advantage of system vulnerabilities[1], [2].

Implementing effective IDS has become a crucial tactic for preserving the security of computer systems and networks due to the growing threats to network security [3]. Conventional intrusion detection techniques mostly rely on known attack patterns and behaviors that are generated from historical data or the expertise of practitioners. As a result, their capacity to recognize novel and unidentified attack techniques is restricted. As a result, an intrusion detection system (IDS) that can automatically identify suspicious behavior is required [1], [4].

An Intrusion Detection System (IDS) is a system that integrates network traffic and identifies unusual activity that indicates an attack. Over time, machine learning-based methods have become increasingly popular because they can improve detection accuracy and adaptively recognize new attack patterns. Machine learning provides extensive and complex data analysis capabilities, making it highly efficient in modern network security

Corresponding Author: Sarah Anjani (dosen03345@unpam.ac.id)
1 Sarah Anjani 1, Universitas Pamulang, dosen03345@unpam.ac.id
2 Encik Yoega Renaldi 2, Universitas Pamulang, dosen03347@unpam.ac.id
3 Feby Charlos 3, Universitas Pamulang, dosen03351@unpam.ac.id
4 Maisan Dewi Puspa Khairani, Universitas Pamulang, dosen03348@unpam.ac.id

systems [5], [6], [7]. An IDS is an essential component of modern network security systems. It is intended to keep an eye on network traffic and instantly spot questionable conduct. Signature-based and anomaly-based are the two main types of IDS. Zero-day attacks cannot be detected by signature-based intrusion detection systems, but they are successful in identifying known attacks. IDS based on machine learning and anomaly detection, on the other hand, can identify new assaults by identifying departures from typical patterns. Signature-based and anomaly-based IDS are combined to create hybrid IDS [8].

In the era of deep learning and generative AI, classical machine learning algorithms such as DT and RF retain significant relevance for IDS applications. Although deep learning models may achieve marginally higher accuracy on large datasets, they require extensive computational resources, substantial labeled training data, and long training times that are often impractical for real-time deployment in resource-constrained network environments [9], [10]. In contrast, DT and RF offer a compelling balance of interpretability, computational efficiency, and robust performance. DT provides transparent, rule-based classification that security analysts can directly audit and understand a critical requirement in regulated environments where explainability is mandated. Random Forest, as an ensemble of DTs, addresses the overfitting limitation of individual trees while maintaining low computational overhead compared to deep neural networks. Recent studies confirm that tree-based ensemble models remain competitive with deep learning on structured tabular network traffic data [11],[12]. Furthermore, in the context of generative AI-augmented cyber threats, where attack patterns evolve rapidly, the fast-retraining capability of RF provides a practical advantage over computationally expensive deep learning pipelines.

IDS uses a variety of classification techniques, such as RF and DT. DT offers the advantage of fast computation and easy model interpretation, making it a transparent and interpretable baseline classifier that is widely adopted in network security research. However, DTs frequently suffer from overfitting issues, particularly when dealing with high-dimensional and imbalanced datasets. Random Forest, as an ensemble technique, addresses this limitation by integrating multiple DTs, thereby enhancing generalization performance and producing a more precise and reliable model [13]. The selection of these two algorithms in this study is intentional: DT serves as a strong interpretable baseline, while RF represents an advanced ensemble approach, allowing a meaningful and systematic comparison of individual versus ensemble learning strategies in the context of network intrusion detection [14]. Because it covers a wide range of assaults, including DoS, Probe, R2L, and U2R, the KDD Cup 99 dataset is frequently used as a benchmark dataset in Intrusion Detection System (IDS) research. The efficacy of classification algorithms may be impacted by this dataset's large data dimensionality and unequal class distribution, necessitating a thorough and precise assessment [1].

Recent research has shown that ensemble algorithms such as RF demonstrate superior performance compared to individual algorithms in detecting network intrusions. This is because the ensemble approach can reduce overfitting on complex data and enhance model generalization [15]. This study aims to evaluate and compare the performance of RF and DT algorithms in detecting network intrusions using the KDD Cup 99 dataset, in order to provide recommendations for selecting the best algorithm in developing a more reliable and efficient IDS. A systematic comparative evaluation is conducted under consistent experimental conditions as a fair and reproducible performance benchmark, while analyzing ensemble versus individual approaches on imbalanced data to reveal the superiority of ensemble methods in real intrusion detection scenarios.

2. Related Works

Numerous prior studies have been carried out to enhance the performance of IDS [16], [17], [18], [19], specifically with regard to anomaly detection accuracy. These studies have employed a variety of techniques, including the use of a single classifier, ensemble learning, and a combination of machine learning and feature extraction. Experiments are carried out in IDS research to increase anomaly detection systems' accuracy. In order to compare ensemble learning and feature selection techniques, a baseline accuracy value is first obtained using a single classifier. The goal of using ensemble learning is to outperform individual approaches. Evaluation is done by comparing the outcomes of the two approaches and consulting the confusion matrix value. According to the study's findings, an ensemble learning strategy can boost accuracy to 96.8%, but a single classifier (Naive Bayes) can only achieve 77.4% [20].

To improve IDS detection accuracy, one study suggested an IDS model that combines CNN and Random Forest. CNN automatically extracts network features to reduce dimensionality and noise, followed by classification using Random Forest. Testing on the KDD Cup 1999 and UNSW-NB15 datasets showed that this model achieved 97% accuracy and more than 98% precision, and was better than conventional machine learning methods [21]. Using the UNSW-NB15 dataset, a different study compared the effectiveness of IDS with feature selection using the RF Classifier method. This project aimed to use machine learning to speed up IDS processing time. Two phases of the investigation were carried out: one without feature selection and the other with Extra Trees Classifier feature selection. The findings demonstrated that while there is a minor drop in accuracy, utilizing features can speed up the RF Classifier's detection time [4].

Experiments were carried out in a different IDS study to increase the anomaly detection system's accuracy. A baseline accuracy value was obtained in the first phase using a single classifier, and it was subsequently compared with feature selection and ensemble learning techniques. The goal of using ensemble learning is to outperform individual approaches. The numbers in the confusion matrix and a comparison of the outcomes of the two methods served as the basis for the evaluation. According to the study's findings, the accuracy of a single classifier (Naïve Bayes) was 77.4%, whereas the accuracy of the ensemble learning strategy was 96.8% [20].

A more sophisticated approach was proposed that integrates K-Means, Random Forest, and Deep Learning (CNN-LSTM) on the CIC-IDS2017 dataset. This hybrid method achieved 99.91% accuracy, outperforming conventional methods [9]. However, additional research has demonstrated that the use of Deep Learning techniques is frequently hindered by lengthy training periods and the requirement for high-end hardware, which might not be appropriate for real-time deployment in networked systems. A recent study reinforced this efficiency aspect by demonstrating that inference speed is an essential component to mitigate real-time attacks, frequently outweighing minor accuracy increases [22]. In order to assess machine learning models in terms of model performance and the computational time needed to train them, this study will carry out experiments.

3. Proposed Method

The purpose of this study is to evaluate an IDS's performance by comparing the classification model performance of RF and DT methods. The research steps will be explained in this section. The research flow used in this work is schematically shown below, as seen in Fig. 1:

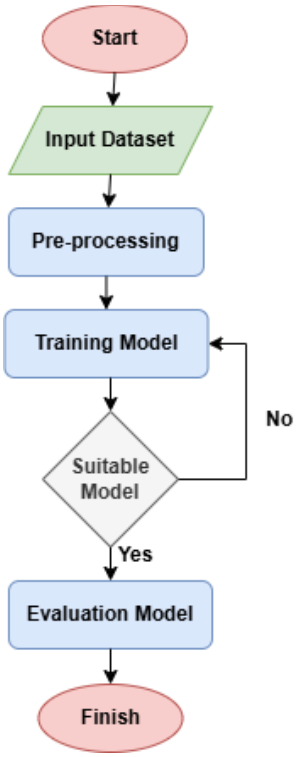

Fig. 1. Research flow

In this study, the collected data is subsequently subjected to preprocessing, including cleaning, transformation, and normalization. A model that can recognize patterns or make predictions is then created using the data during the model training stage. We also conduct a model suitability evaluation is conducted to determine whether the resulting model meets the expected performance standards. If the model does not achieve optimality, further iterations are conducted in the training phase for improvement. On the other hand, if the model is judged appropriate, the following stage is the model assessment phase, which uses particular evaluation metrics to evaluate the model's performance in further detail. This phase serves as a baseline for determining model effectiveness before the process is declared complete.

3.1 Algorithm Justification in the Deep Learning Era

Even though deep learning models are widely used in cybersecurity research, traditional machine learning methods are still crucial for deploying intrusion detection systems. For this study, RF and DT were chosen for the reasons listed below. First, interpretability: Security analysts can immediately comprehend and audit classification judgments because to DT's transparent, human-readable decision rules, which is essential in compliance-driven settings [12]. Second, computational efficiency: both algorithms are appropriate for real-time IDS deployment on conventional hardware since they require substantially less training time and processing resources than deep neural networks [23], [24]. Third, resilience on tabular data: studies consistently show that on structured tabular network traffic data, tree-based ensemble models are on par with or better than deep learning [24]. Fourth, resilience against idea drift: While deep learning retraining is computationally costly, RF can be effectively retrained when attack patterns change. Fifth,

explainability alignment with XAI requirements: RF supports Explainable AI (XAI) frameworks like SHAP and LIME, which are becoming more and more necessary in production IDS systems in the age of generative AI, where black-box models are under examination [12].

3.2 RF

The An ensemble learning technique called the RF algorithm generates several separate DT models [10]. To produce forecasts that are more accurate and reliable, this technique combines the results of multiple DTs. The bootstrap sampling approach, which uses sampling with replacement, is used to randomly select training data during the creation process so that each tree is trained with a distinct subset of the data. Additionally, at each tree building, random feature selection is carried out to boost model variety and improve generalization skills [25].

Through this mechanism, RF can perform efficient classification, even on data with incomplete attributes and datasets with a large number of samples. During the classification process, the data is randomly divided into several DTs, which then form a tree structure with the root node as the starting point, internal nodes as decision branches, and leaf nodes as the final classification results. The ultimate decision is made by combining the forecast results from each tree, strengthening and standardizing the final model [26]. The following is the RF architecture which can be seen in Fig. 2:

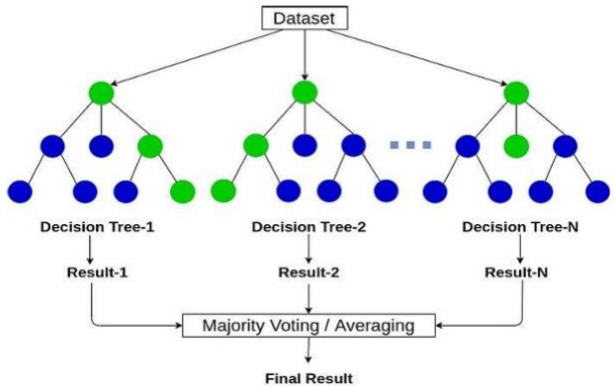

Fig. 2. Architecture RF

In this study, RF is employed to classify network traffic into normal or intrusion categories. Given a training dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where x_i denotes the feature vector and y_i denotes the corresponding class label, a total of T DTs is constructed using bootstrap samples drawn randomly with replacement from the training dataset. At each node, only a random subset of features is considered as split candidates, which is commonly set to $m = \sqrt{p}$ for classification tasks, where p denotes the total number of features. This technique aims to increase diversity among the trees, thereby producing a more robust model.

The quality of data splitting at each node is evaluated using the Gini Impurity index, formulated as:

$$Gini\ (D) = 1 - \sum_{k=1}^{K} p_k^2 \quad (1)$$

Where K denotes the number of classes and p_k represents the proportion of samples belonging to class K. A lower Gini value indicates that the data at that node is more homogeneous or pure. For each split candidate, the impurity after splitting is computed as:

$$Gini_{split} = \frac{|D_{left}|}{|D|} Gini\left(D_{left}\right) + \frac{|D_{right}|}{|D|} Gini\left(D_{right}\right) \quad (2)$$

where D_{left} and D_{right}represent the data in the left and right branches resulting from the split, respectively. The optimal split is selected based on the smallest $Gini_{split}$ value. Once all trees are constructed, classification is performed by aggregating the predictions from all trees through majority voting. The final prediction of the RF is formulated as:

$$H(x) = argmax_y \sum_{t=1}^{T} I(h_t(x) = y) \quad (3)$$

where $H(x)$is the final classification output, $h_t(x)$ is the prediction of the t-th tree, I(·) is an indicator function that equals 1 when the condition is satisfied and 0 otherwise, and T denotes the total number of trees in the forest. The class that receives the highest number of votes is assigned as the final prediction.

3.3 DT

A supervised classification approach that resembles a tree and can be used for both classification and regression problems is called a DT. This method works by dividing a dataset into more uniform groups based on the independent variables or attributes deemed most influential, resulting in the best possible separation of the data. In a DT framework, each node represents a condition or test for a specific attribute, while branches represent decision rules that direct the data to the next subset. This procedure keeps going until the last node (leaf), which stands for the data's classification or prediction outcome, is achieved [27], [28].

The mathematical formulation for DT construction is based on information gain, which measures the reduction in entropy after a dataset is split on attribute A. The entropy of dataset D is defined as:

$$Entropy\ (D) = -\sum_{k=1}^{K} p_k log_2(p_k) \quad (4)$$

where $p_k$is the proportion of samples belonging to class k in dataset D. The information gain for splitting on attribute A is then computed as:

$$Gain\ (D, A) = Entropy\ (D) - \sum_{v \in Values(A)} \frac{D_v}{D} . Entropy\ (D_v) \quad (5)$$

where $D_v$is the subset of $D$for which attribute Atakes value v. At each internal node, the attribute yielding the highest information gain is selected as the splitting criterion. This recursive partitioning process continues until all leaf nodes are pure that is, $Entropy = 0$ or a predefined stopping criterion is met, such as a maximum tree depth or a minimum number of samples per node. Although DT is highly interpretable and computationally efficient, its tendency to overfit training data particularly on high-dimensional and imbalanced datasets remains a well-documented limitation. This shortcoming is addressed by ensemble methods such as Random Forest, which aggregates multiple DTs to improve generalization performance.

3.4 Evaluation Metrics

In this study, the goal of using these measures is to evaluate how well the model's features work and identify the approach that yields the best results. A confusion matrix, which displays the classification findings as four primary components including True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [10], [11]. The confusion matrix shown in Fig. 3 is depicted in the following image:

Actual

Predicted Value	True	False
True	TP	FP
False	FN	TN

Fig. 3. Confusion matrix

Fig. 3 explains the elements or components of a confusion matrix that can be used to evaluate IDS models using RF algorithms and DTs.

4. Experimental Setup

In this study, we conduct the preprocessed KDD Cup 99 dataset to drive the experiments. This paper implements two classifiers, a RF and a DT, and train these models on the available training data. We then evaluate the trained models on a held-out test set to demonstrate their ability to detect both normal activity and a range of network intrusions. It is to quantify performance using the confusion matrix and standard metrics such as accuracy, precision, recall, and F1-score. In addition, this paper compares the computational efficiency of the two algorithms by reporting training and testing times. This study aims to deliver a comprehensive assessment of the model's efficacy and efficiency in detecting network intrusions.

4.1 Dataset

In this paper, we collected the KDD Cup 99 dataset, one of the most widely used data sources in anomaly detection research, especially for IDS. Stolfo and colleagues developed the dataset from information obtained during the DARPA-98 IDS program evaluation, and each record is labeled as normal or attack based on attributes that capture the features of network traffic. The dataset identifies four primary attack types, including DoS, U2R, R2L, and Probe, to reflect common patterns observed in real network environments. This study relies on the KDD Cup 99 data as a benchmark for IDS development. The data consist of 11 attributes, including a class attribute that serves as the target label, and a total of 9,537 records, with normal and attack instances displaying different class distributions. Figure 4 shows the distribution of classes in the target set.

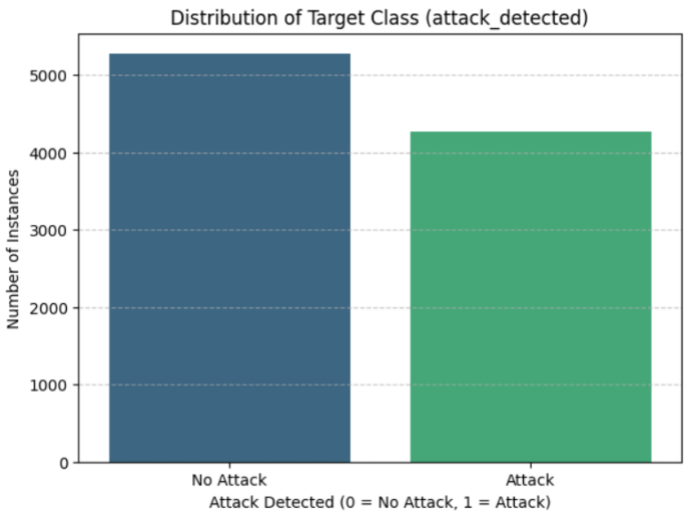

Fig. 4. Class distribution

Fig. 4 explains the distribution of the target classes, namely normal (0) and attack (1). The total amount of data in the normal class is 5,273, while in the attack class there are 4,267 data.

4.2 Data Pre-processing

Before employing the dataset to train the IDS model, we performed several preprocessing steps to ensure data quality. We first executed a data-type transformation by converting the session n_d duration column from an object type to a numeric type, specifically int64, by removing the delimiter (dot). We then addressed missing values through data cleaning, which entails identifying, correcting, or removing erroneous, incomplete, redundant, or irrelevant observations from the dataset. Collectively, these procedures enhance data consistency and reliability, thereby improving the robustness of subsequent modeling.

Table 1 below shows the distribution of the dataset used to build the model:

Table. 1. Dataset distribution

Data	Percentage	number
Data training	80%	7.629
Data testing	20%	1.908
Total	**100%**	**9.537**

5. Result and Analysis

The evaluation outcomes of models built using the RF and DT algorithms are shown in this section. Based on the used dataset, testing was done to determine how well each model identified network intrusions. The performance and robustness of the model were then evaluated by analyzing the data using a variety of assessment criteria, including accuracy, precision, recall, and F1 score. The best-performing algorithm in an IDS was identified by comparing the evaluation outcomes. The confusion matrices employed for the RF and DT models yielded the following results:

5.1 Random Forest

The RF model's evaluation results, which are based on the confusion matrix to examine the model's capacity to fully categorize attack and normal data, are as follows:

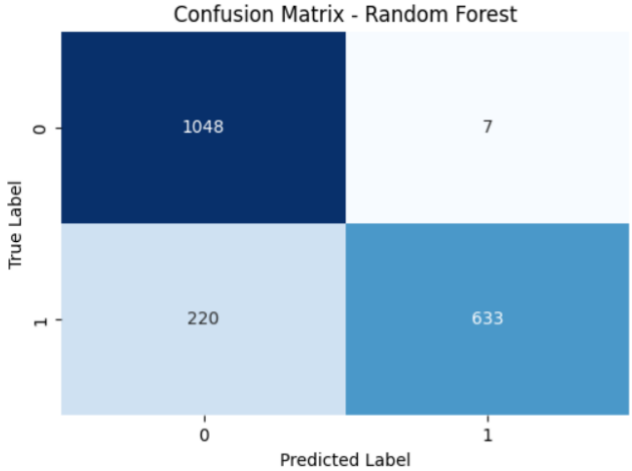

Fig. 5. Confusion matrix RF

Fig. 5 shows the confusion matrix results of the RF algorithm. The RF model demonstrated strong overall classification performance on the test dataset. Analysis of the confusion matrix reveals that the model correctly classified 1,048 normal records as normal (True Negative) and 633 attack records as attacks (True Positive). The False Positive count of 7 indicates that the model produces very few false alarms, correctly identifying normal traffic in the vast majority of cases. However, the model recorded 220 False Negatives, meaning that a portion of actual attack instances were misclassified as normal. This pattern reflects the well-known precision-recall trade-off characteristic of RF on imbalanced datasets: the model is highly conservative in labeling instances as attacks, resulting in very high precision (0.98) but moderate recall (0.74).

The high precision of 0.98 is particularly significant for operational IDS deployment, as it indicates that when RF raises an attack alert, it is correct 98% of the time. This dramatically reduces the burden on security analysts from investigating false alarms, which is a major operational challenge in real-world IDS environments. The F1-score of 0.84 represents a strong balance between precision and recall, and the overall accuracy of 0.88 confirms that RF correctly classifies 88% of all network traffic records in the test set.

5.2 Decision Tree

The evaluation results utilizing the confusion matrix for the DT model, which are shown in Fig. 6, are as follows:

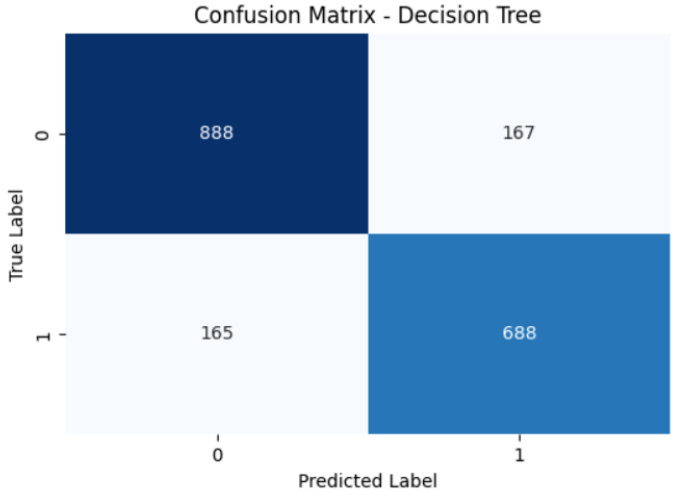

Fig. 6. Confusion matrix DT

Fig. 6 shows the results of the confusion matrix from the DT algorithm The DT model achieved a more balanced distribution between True Positives (688) and True Negatives (888), with False Positives of 167 and False Negatives of 165. This relatively symmetric error distribution explains the equal precision and recall values of 0.80, resulting in an F1-score of 0.80 and overall accuracy of 0.82. Compared to Random Forest, DT produces substantially more false alarms (167 versus 7), reflecting a more aggressive classification boundary that is less conservative in labeling instances as attacks.

While the balanced precision and recall of DT may appear advantageous from a recall perspective, the high False Positive rate of 167 represents a significant operational disadvantage. In a real-world IDS, each false positive requires analyst investigation, consuming time and resources. The DT's tendency toward false alarms is a direct consequence of its single-tree structure and susceptibility to overfitting, which causes it to learn overly specific decision boundaries that do not generalize as well as the ensemble approach.

5.3 Comparison of Model Results

The following is a comparison table of machine learning models, which can be seen in Table 2:

Table. 2. Comparison of IDS model performance

Model	Accuracy	Precision	Recall	F1-Score	Training Time (s)
RF	0.88	0.98	0.74	0.84	1.212
DT	0.82	0.80	0.80	0.80	0.077

The comparative analysis reveals that RF outperforms DT on three of the four primary evaluation metrics. In terms of accuracy, RF achieves 0.88 compared to DT's 0.82, representing a 6-percentage point improvement that reflects better overall classification capability. The precision difference is most dramatic: RF achieves 0.98 while DT achieves only 0.80, an 18-percentage point gap demonstrating that RF is far more effective at reducing false positive alerts. This is the most critical metric for operational IDS deployment, as false alarms degrade analyst trust and increase operational costs.

In the recall metric, DT slightly outperforms Random Forest, achieving 0.80 compared to 0.74. This 6-percentage point advantage indicates that DT detects a slightly higher proportion of actual attack instances, which could be advantageous in high-security environments where missing any attack is unacceptable. However, this advantage must be weighed against DT's substantially higher false positive rate. The F1-score, which harmonically balances precision and recall, favors RF at 0.84 versus 0.80, confirming that RF achieves a better overall balance between detection completeness and alert reliability.

Regarding training efficiency, DT is dramatically faster, requiring only 0.077 seconds compared to Random Forest's 1.212 seconds — approximately 16 times faster. This computational advantage is relevant for time-sensitive deployments or environments with constrained processing resources. However, the 1.212-second training time of RF remains entirely practical for batch model retraining scenarios. The results are consistent with prior comparative studies showing Random Forest's superiority on structured network traffic data [1], [14], confirming that ensemble-based approaches offer a more robust and reliable foundation for IDS development. The following is an illustration of the model comparison metrics, which can be seen in Fig. 7:

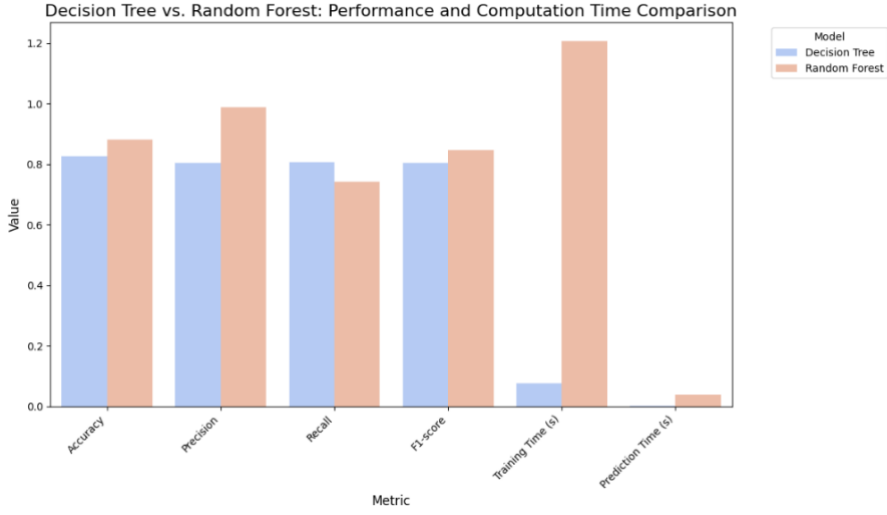

Fig. 7. Model comparison chart

The results of this study carry particular significance in the context of the current deep learning and generative AI era. While deep learning models such as LSTM and CNN-LSTM have reported higher accuracy rates (99%+) on IDS benchmarks, they require substantially larger datasets, longer training times, and dedicated GPU hardware. The performance achieved by RF in this study (0.88 accuracy, 0.98 precision) demonstrates that high-quality intrusion detection is achievable with classical machine learning at a fraction of the computational cost, making it accessible for deployment in resource-constrained environments such as small enterprise networks, edge computing, and IoT security infrastructure [10], [12]. Furthermore, the explain ability of RF through feature importance rankings and SHAP values aligns with emerging regulatory requirements for transparent AI decision-making in security applications

6. Conclusion

This study compared the performance of DT and RF algorithms for intrusion detection using the KDD Cup 99 dataset. The results showed that RF achieved better overall performance, with higher accuracy (0.88), precision (0.98), and F1-score (0.84) than DT. Its significantly higher precision indicates fewer false positive alerts, making it more suitable for practical IDS deployment. Meanwhile, DT achieved higher recall (0.80) and faster training time, making it useful for environments requiring rapid retraining and higher attack detection sensitivity. These findings indicate that RF is generally more reliable and robust for intrusion detection, while DT offers advantages in speed and simplicity. Both algorithms remain relevant in resource-constrained environments due to their interpretability and computational efficiency.

For future work, further evaluation should be conducted using more recent and realistic intrusion detection datasets, such as CICIDS2017, UNSW-NB15, and CIC-IDS2018, to assess performance against modern cyber threats. Future studies may also incorporate hyperparameter optimization techniques, including grid search and Bayesian optimization, to improve model effectiveness. Additionally, hybrid approaches that combine RF with deep learning models could be explored to achieve higher detection performance while maintaining interpretability. Real-time deployment experiments in live network environments are also recommended to validate practical applicability. Furthermore, integrating explainable AI techniques such as SHAP can enhance transparency and trust in IDS decision-making. Finally, investigating federated learning-based IDS architectures may support privacy-preserving and distributed cybersecurity solutions across multiple organizations.

Acknowledgment

The authors express their gratitude to our university for providing institutional and financial assistance through an Internal Research Grant. We also thank the Kaggle platform, the KDD Cup 99 dataset source, and everyone else who helped with this research.

References

[1] C. Lu, Y. Cao, and Z. Wang, “Research on Intrusion Detection Based on an Enhanced RF Algorithm,” *Applied Sciences (Switzerland)*, vol. 14, no. 2, Jan. 2024, doi: 10.3390/app14020714.

[2] M. Politeknik, N. J. Devita, and A. Larasati, “Penerapan Algoritma Rf Dan K-nearest Neighbor Untuk Deteksi Intrusi Pada Dataset CICIDS2017,” *Jurnal Multidisiplin Ilmu Akademik*, vol. 2, no. 6, pp. 632–640, 2025, doi: 10.61722/jmia.v2i6.7213.

[3] E. M. Maseno, Z. Wang, and H. Xing, “A Systematic Review on Hybrid Intrusion Detection System,” 2022, *Hindawi Limited*. doi: 10.1155/2022/9663052.

[4] S. Budiman, A. Sunyoto, and A. Nasiri, “SISTEMASI: Jurnal Sistem Informasi Analisa Performa Penggunaan Feature Selection untuk Mendeteksi Intrusion Detection Systems dengan Algoritma RF Classifier.” [Online]. Available: http://sistemasi.ftik.unisi.ac.id

[5] Sudhanshu Sekhar Tripathy and Bichitrananda Behera, “Performance Evaluation Of Machine Learning Algorithms For Intrusion Detection System,” *Journal of Biomechanical Science and Engineering Japan Society of Mechanical Engineers*, no. April, 2023, doi: 10.48175/ijarsct-24434.

[6] M. T. Abdelaziz *et al.*, “Enhancing Network Threat Detection with Random Forest-Based NIDS and Permutation Feature Importance,” *Journal of Network and Systems Management*, vol. 33, no. 1, Mar. 2025, doi: 10.1007/s10922-024-09874-0.

[7] M. A. Rachmatullah *et al.*, “Analisis Efektivitas Intrusion Detection System (IDS) Untuk Serangan DDoS Menggunakan Komparasi Random Forest Dan Decision Tree,” vol. 10, no. 1, pp. 1536–1541, 2026.

[8] S. V. N. Santhosh Kumar, M. Selvi, and A. Kannan, “A Comprehensive Survey on Machine Learning-Based Intrusion Detection Systems for Secure Communication in Internet of Things,” *Comput. Intell. Neurosci.*, vol. 2023, no. 1, Jan. 2023, doi: 10.1155/2023/8981988.

[9] C. Liu, Z. Gu, and J. Wang, “A Hybrid Intrusion Detection System Based on Scalable K-Means+ RF and Deep Learning,” *IEEE Access*, vol. 9, pp. 75729–75740, 2021, doi: 10.1109/ACCESS.2021.3082147.

[10] N. Kayambu and S. Kumar, “Performance Comparison of Feature Selection Methods for Machine Learning Models on DDoS Attack Dataset,” in *2025 IEEE 6th Annual World AI IoT Congress, AIIoT 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 595–602. doi: 10.1109/AIIoT65859.2025.11105263.

[11] D. P. Hostiadi, Y. P. Atmojo, R. R. Huizen, I. M. D. Susila, G. A. Pradipta, and I. M. Liandana, “A New Approach Feature Selection for Intrusion Detection System Using Correlation Analysis,” in *2022 4th International Conference on Cybernetics and Intelligent System, ICORIS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/ICORIS56080.2022.10031468.

[12] M. A. Talukder *et al.*, “Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction,” *J. Big Data*, vol. 11, no. 1, Dec. 2024, doi: 10.1186/s40537-024-00886-w.

[13] M. Azhar, S. Perveen, A. Iqbal, and B. Lee, “IDRandom-Forest: Advanced RF for Real-time Intrusion Detection,” *IEEE Acces*, 2024.

[14] Z. Azam, M. M. Islam, and M. N. Huda, “Comparative Analysis of Intrusion Detection Systems and Machine Learning-Based Model Analysis Through DT,” *IEEE Access*, vol. 11, pp. 80348–80391, 2023, doi: 10.1109/ACCESS.2023.3296444.

[15] H. M. R. U. Rehman *et al.*, “A systematic literature study of machine learning techniques based intrusion detection: datasets, models, challenges, and future directions,” *J. Big Data*, vol. 12, no. 1, Dec. 2025, doi: 10.1186/s40537-025-01323-2.

[16] C. Author, H. Y. Fan, and S. Anjani, “IDS-GAN: Stepping up Intrusion Detection Method using GAN Algorithm,” *International Journal of Informatics and Computation (IJICOM)*, vol. 5, no. 1, 2023, doi: 10.35842/ijicom.

[17] K. Anitha and K. R. Gandhi, “Ensemble of Machine Learning Model with Tuna Swarm Optimization-Driven Feature Selection for Cybersecurity Threat Detection and Classification Approach,” *Journal of Intelligent Systems and Internet of Things*, vol. 15, no. 2, pp. 76–90, 2025, doi: 10.54216/JISIoT.150206.

[18] M. H. Wathan, Indra Irwan, Better Swengky, M Syafrizal Zain, Ardi Ramadani, and Selamet Riadi, “131-Article Text-638-1-10-20250628,” *International Journal of Informatics and Computation (IJICOM)*, vol. 7, 2025.

[19] C. H. Huang J, “Effective Ransomware Attacks Detection Using CNN Algorithm,” *International Journal of Informatics and Computation (IJICOM)*, vol. 5, no. 2, p. 2023, doi: 10.35842/ijicom.

[20] R. Sudiyarno, A. Setyanto, and E. T. Luthfi, “Peningkatan Performa Pendeteksian Anomali Menggunakan Ensemble Learning dan Feature Selection Anomaly Detection Performance Improvement Using Ensemble Learning and Feature Selection,” *Citec Journal*, vol. 7, no. 1, 2020.

[21] P. A. doost, S. S. Moghadam, E. Khezri, A. Basem, and M. Trik, “A new intrusion detection method using ensemble classification and feature selection,” *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-98604-w.

[22] M. Haggag, M. M. Tantawy, and M. M. S. El-Soudani, “Implementing a deep learning model for intrusion detection on apache spark platform,” *IEEE Access*, vol. 8, no. 1, pp. 163660–163672, 2020, doi: 10.1109/ACCESS.2020.3019931.

[23] C. Lu, Y. Cao, and Z. Wang, “Research on Intrusion Detection Based on an Enhanced RF Algorithm,” *Applied Sciences (Switzerland)*, vol. 14, no. 2, Jan. 2024, doi: 10.3390/app14020714.

[24] N. Kayambu and S. Kumar, “Performance Comparison of Feature Selection Methods for Machine Learning Models on DDoS Attack Dataset,” in *2025 IEEE 6th Annual World AI IoT Congress, AIIoT 2025*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 595–602. doi: 10.1109/AIIoT65859.2025.11105263.

[25] L. Lei, S. Shao, and L. Liang, “An evolutionary deep learning model based on EWKM, RF algorithm, SSA and BiLSTM for building energy consumption prediction,” *Energy*, vol. 288, Feb. 2024, doi: 10.1016/j.energy.2023.129795.

[26] E. A. Winanto, Y. Novianto, S. Sharipuddin, I. S. Wijaya, and P. A. Jusia, “Peningkatan Performa Deteksi Serangan Menggunakan Metode Pca Dan Random Forest,” *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, pp. 285–290, Apr. 2024, doi: 10.25126/jtiik.20241127678.

[27] E. Emirmahmutoğlu and Y. Atay, “A feature selection-driven machine learning framework for anomaly-based intrusion detection systems,” *Peer. Peer. Netw. Appl.*, vol. 18, no. 3, Jun. 2025, doi: 10.1007/s12083-025-01947-4.

[28] S. Seniaray and R. Jindal, “Performance Analysis of Anomaly-Based Network Intrusion Detection Using Feature Selection and Machine Learning Techniques,” *Wirel. Pers. Commun.*, vol. 138, no. 4, pp. 2321–2351, Oct. 2024, doi: 10.1007/s11277-024-11602-5.

[29] H. Haeruddin, E. Erick, and H. W. Aripradono, “Perbandingan Support Vector Machine, RF Classifier, dan K-Nearest Neighbour dalam Pendeteksian Anomali pada Jaringan DDos,” *JTIM : Jurnal Teknologi Informasi dan Multimedia*, vol. 7, no. 1, pp. 23–33, Jan. 2025, doi: 10.35746/jtim.v7i1.628.