International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 38 العدد
Part 1 المجلد
المجلة الدولية للعلوم والتقنية
International Science and Technology Journal
ISTJ

http://www.doi.org/10.62341/shlc2687

Received	**2026/02/09**	**تم استلام الورقة العلمية في**
Accepted	**2026/02/27**	**تم قبول الورقة العلمية في**
Published	**2026/02/28**	**تم نشر الورقة العلمية في**

Comparison of Different Machine Learning Techniques for Network Intrusion Detection

***Sadeg Ali Mohamed Alarbi[1], Lateefha Meeloud Amhimmid Almawaddi[2], Heba M. Ali Abouasha[3]**

The Higher Institute of Sciences and Technology, Alzhraa - Libya
Sadegaliarbi@gmail.com*
*ORCID No. 0009-0005-8626-8191

Abstract:
Network intrusion detection has become a critical component of modern cybersecurity systems due to the rapid growth of internet usage and the increasing volume and complexity of cyber-attacks. Cybersecurity protects computer systems and networks from unauthorized access [1]. This paper presents a comparative evaluation of three machine learning techniques namely Random Forest (RF), Decision Tree (DT), and K-Nearest Neighbors (KNN) for intrusion detection using the NSL-KDD benchmark dataset. The models were evaluated based on Accuracy, Precision, Recall, F1-score, Inference Time, and Confusion Matrix analysis. Experimental results show that Decision Tree achieved the fastest inference time (0.2397 sec) with balanced performance, while Random Forest achieved the highest precision (0.9677), indicating reliable attack detection with fewer false alarms. However, KNN, despite acceptable accuracy and precision, exhibited very high inference time (17.432366 sec), making it unsuitable for real-time deployment. The results highlight the importance of balancing detection performance and computational efficiency when selecting machine learning models for practical intrusion detection systems.
These results demonstrate that Decision Tree provides an effective trade-off between detection accuracy and computational efficiency for real-time intrusion detection systems.

Keywords: Machine learning, Intrusion Detection System, Decision Tree, Random Forest, KNN, NSL-KDD.

International Science and Technology Journal
المجلة الدولية للعلوم والتقنية
Volume 38 العدد
Part 1 المجلد
http://www.doi.org/10.62341/shlc2687

مقارنة تقنيات التعلم الآلي المختلفة لاكتشاف التسلل في الشبكات

*الصادق علي محمد العربي[1]، لطيفة ميلود امحمد المودي[2]
هبة مصطفي علي ابوعائشة[3]
قسم الحاسوب، المعهد العالي للعلوم والتقنية الزهراء – ليبيا
*Sadegaliarbi@gmail.com

الملخص:

أصبح كشف الأختراقات في الشبكات عنصراً أساسيًا في أنظمة الأمن السيبراني الحديثة نظرًا للنمو السريع في استخدام الإنترنت والزيادة المستمرة في حجم وتعقيد الهجمات السيبرانية. يعمل الأمن السيبراني على حماية أنظمة الحاسوب والشبكات من الوصول غير المصرح به [1]. وتقدم هذه الورقة تقييمًا مقارنًا لثلاث تقنيات من تقنيات التعلم الآلي، وهي: الغابة العشوائية(RF) ، وشجرة القرار(DT) ، وخوارزمية أقرب الجيران (KNN)، وذلك لكشف الأختراقات باستخدام مجموعة بيانات NSL–KDD المعيارية.

وقد تم تقييم النماذج بناءً على مقاييس: الدقة(Accuracy) ، والضبط(Precision) ، والاسترجاع(Recall) ، ومقياس F1 ، وزمن الاستدلال(Inference Time) ، وتحليل مصفوفة الالتباس(Confusion Matrix). أظهرت النتائج التجريبية أن نموذج شجرة القرار حقق أسرع زمن استدلال (0.2397 ثانية) مع أداء متوازن، بينما حقق نموذج الغابة العشوائية أعلى دقة (0.9677)، مما يشير إلى قدرة موثوقة على كشف الهجمات مع عدد أقل من الإنذارات الكاذبة. ورغم أن خوارزمية أقرب الجيران KNN حققت دقة وضبط مقبولين، إلا أنها أظهرت زمن استدلال مرتفعًا جدًا (17.432366 ثانية)، مما يجعلها غير مناسبة للتطبيق في الوقت الفعلي (الزمن الحقيقي).

تبرز النتائج أهمية تحقيق التوازن بين دقة الكشف والكفاءة الحاسوبية عند اختيار نماذج التعلم الآلي لأنظمة كشف الأختراقات العملية. وتُظهر هذه النتائج أن نموذج شجرة القرار يوفر مفاضلة فعّالة بين دقة الكشف والكفاءة الحاسوبية لأنظمة كشف الأختراق في الزمن الحقيقي.

الكلمات المفتاحية: التعلم الآلي، نظام كشف الأختراقات، شجرة القرار، الغابة العشوائية، خوارومية أقرب الجيران، مجموعة بيانات NSL–KDD المعيارية.

http://www.doi.org/10.62341/shlc2687

1. Introduction

The rapid development and growth of computer networks and internet-based services has
significantly increased the exposure of information systems to cyber-attacks. Network intrusions represent one of the most critical challenges to information security, as attackers continuously develop new techniques to compromise network confidentiality, integrity, and availability of network systems.

Intrusion Detection Systems (IDS) have emerged as an essential security mechanism for monitoring network traffic and identifying malicious activities. Traditional IDS approaches, particularly signature-based methods, are limited in their ability to detect novel and unknown attacks.

In recent years, machine learning techniques have gained significant attention in IDS development due to their ability to learn complex patterns from data and improve detection accuracy. This paper presents a comparative evaluation of three widely used machine learning algorithms namely Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN) using the NSL-KDD benchmark dataset. Performance experiments are performed using the NSL-KDD dataset to determine whether the observed activity is malicious or innocuous [2].

2. Definitions and Performance Metrics

The performance of the proposed intrusion detection models is evaluated using standard classification metrics derived from the confusion matrix, which consists of four elements: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These represent correctly and incorrectly classified attack and normal instances.

Based on these elements, the following performance metrics are calculated:

Accuracy measures the overall correctness of the model:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Precision represents the proportion of correctly detected attacks:

$$Precision = \frac{TP}{TP+FP}$$

http://www.doi.org/10.62341/shlc2687

Recall (Sensitivity) indicates the ability of the model to detect actual attacks:

$$Recall = TP/(TP + FN)$$

F1-score is the harmonic mean of precision and recall:

$$F1 - score = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

These metrics provide a comprehensive evaluation of intrusion detection performance by balancing detection accuracy and the trade-off between false positives and false negatives.

3- Related Work

There are several previous studies that have investigated the application of machine learning techniques for network intrusion detection. Most existing research focuses on comparing a large number of algorithms to evaluate their overall detection accuracy. However, such studies often lack in-depth analysis of individual model behavior, particularly in terms of computational efficiency and real-time applicability.

Recent studies have explored intrusion detection in wireless and IoT environments using advanced machine learning and optimization techniques. For example, Kim and Chong (2017) [3] highlighted the effectiveness of machine learning-based IDS in improving detection accuracy. Kasim suggested using an Auto-Encoding (AE) method for selecting features and reducing dimensions to effectively classify traffic [4]. Budler and Ajoodha (2022) performed a comparative analysis of deep learning models for network intrusion detection systems, highlighting the potential of deep learning techniques to capture complex traffic patterns and improve detection accuracy compared to traditional approaches [5]. Richa & Adesh (2021) – Comparative evaluation of machine learning algorithms for intrusion detection using NSL‑KDD dataset, highlighting their effectiveness in detecting various attack types [6]. Chavan et al. studied the performance of four ML techniques KNN, SVM, DT, and LR for detecting DDoS attacks [7]. Also, Abu Al-Haija and Zein-Sabatto proposed CNN techniques for intrusion detection [8]. Performance experiments were conducted on the

http://www.doi.org/10.62341/shlc2687

NSL_KDD dataset. Logeswari et al. [9], a novel hybrid feature selection and LightGBM-based Intrusion Detection System (IDS) was proposed for SDN. They tested their proposed model on the NSL-KDD dataset and achieved notable results.

Unlike previous studies, this work focuses on a targeted comparative analysis of three widely used machine learning algorithms namely Decision Tree, Random Forest, and K-Nearest Neighbors using the NSL-KDD dataset. This approach enables a clearer evaluation of the trade-off between detection performance and inference time, which is critical for practical intrusion detection systems.

4- Theoretical background:

Wireless networks are among the most important modern technologies in the field of communications. They rely on electromagnetic waves to transmit data and are characterized by several features, the most important of which are flexibility in connection, ease of transmission, mobility, and wide coverage range. However, it faces many security challenges, as the transmitted data is vulnerable to hacking such as eavesdropping, unauthorized access, signal jamming, and some vulnerabilities in encryption protocols. Although NSL-KDD represents wired network traffic, the concepts and machine learning techniques discussed are equally applicable to wireless intrusion detection environments. Although this study is conducted on wired network traffic, wireless intrusion detection concepts are discussed to highlight the broader applicability of machine learning-based IDS.

4.1- Wireless network intrusion detection systems:

Wireless Intrusion Detection Systems (WIDS) are among the most important protection tools in wireless networks, as they monitor and analyze wireless data traffic in order to detect any abnormal activity or potential intrusion attempts. These systems rely on multiple technologies such as signature-based analysis, behavior-based analysis, or machine learning. This enables them to identify attacks in real time. Several commercial and open source WIDS solutions exist to monitor wireless network traffic.

http://www.doi.org/10.62341/shlc2687

4.2- Classification of intrusion detection systems in wireless networks:

Wireless intrusion detection systems (WIDS) can be classified according to several criteria, most notably the monitoring location, detection method, and system purpose. In terms of monitoring location, we can distinguish between network-based systems that monitor network traffic across the entire network, such as Air Magnet Enterprise and Cisco Meraki WIDS/WIPS, and host-based systems that focus on monitoring a specific device, such as OSSEC and Wazuh. As for the detection method, there are signature-based systems that rely on databases of known attacks, anomaly-based systems that detect deviations from normal activity, as well as hybrid systems that combine the two methods, such as Security Onion and Air Defense.

In terms of purpose, the systems include advanced commercial solutions that offer integrated spectral management and analysis, and open-source alternatives suitable for academic and experimental research, such as Kismet and Snort-Wireless. This classification is essential for understanding how to choose the optimal system to protect wireless networks from increasing security risks.

4.3- Key components of intrusion detection systems in wireless networks:

Wireless Intrusion Detection Systems (WIDS) consist of several key components that work together to ensure monitoring and detection of intrusions. The first of these is wireless sensors/probes, which collect data from the network, including wireless packets, access points, and connected devices.

Second, the Analysis Engine, which is responsible for analyzing collected data using signature, behavioral, or hybrid analysis techniques to detect anomalous activity and attacks. Third, the Management Console, which provides a centralized monitoring interface for displaying alerts, reports, and security performance analysis.

Finally, there are response mechanisms, including alerts, automatic isolation of suspicious devices, and integration with Wireless Intrusion Prevention Systems (WIPS) for immediate preventative action.

http://www.doi.org/10.62341/shlc2687

5- Machine learning and intrusion detection systems:

Machine learning has become an effective tool for enhancing the capabilities of intrusion detection systems (IDS) in wireless networks, due to its ability to learn from data and detect unusual patterns that traditional signature systems might miss. These systems rely on data classification and mining algorithms, such as decision trees.

Artificial neural networks and clustering mechanisms are used to identify anomalous behaviors and potential attacks in real time. Machine learning techniques are distinguished by their ability to detect new and unknown attacks and reduce the false alarm rate compared to traditional methods.

They can also be integrated with Intrusion Prevention Systems (IPS) to provide a proactive response aimed at mitigating the impact of attacks on wireless networks.

5.1- Decision Trees and Intrusion Detection Systems:

Decision trees (DTs) are one of the most common machine learning algorithms used in intrusion detection systems (IDS) in wireless networks, due to their simplicity and ability to interactively classify data. This method relies on splitting data into multiple branches based on their characteristics.

Each node represents a test of a specific characteristic, and each branch leads to a specific decision about the nature of the activity, whether it is legitimate or an attack. Decision trees are characterized by the ease with which results can be interpreted, their speed in classification, and their ability to handle large and diverse databases. DT is also frequently used to detect anomalous behaviors and new threats, and can be combined with other technologies such as Random Forests to enhance accuracy and reduce false alarms in wireless network environments.

5.2- Random Forests and Intrusion Detection Systems:

Random Forest (RF) is an advanced machine learning algorithm used in Intrusion Detection Systems (IDS) to enhance classification accuracy and reduce false alarms. RF works by creating a set of decision trees and generating a collective classification based on the majority vote of the trees to determine the nature of network activity, whether it is normal or a security threat.

This technology excels at handling large and complex datasets, tolerating noise or missing data better than individual decision trees,

http://www.doi.org/10.62341/shlc2687

and contributing to the detection of new and unknown attacks in wireless networks. RF is often integrated with WIDS systems to enhance security performance and improve network responsiveness to constantly evolving threats.

5.3- Nearest Neighbor (KNN) algorithm and intrusion detection systems:

The K-Nearest Neighbors (KNN) algorithm is a simple and efficient machine learning algorithm used in intrusion detection systems (IDS) on wireless networks. KNN classifies new data based on its similarity to pre-existing data points, determining the nature of network activity (normal or attack) according to the majority of the K-neighbor classifications in the property space.

This algorithm is characterized by its ease of implementation and the fact that it does not assume any prior data distribution. It can also detect anomalous patterns and unknown attacks.

However, KNN may face challenges in dealing with very large or high-dimensional databases, which calls for the use of performance optimization techniques such as dimensionality reduction or combining them with other algorithms to enhance the efficiency of WIDS systems.

6- Methodology

The overall methodology adopted in this paper is illustrated in Figure 1. which presents the complete workflow of the proposed intrusion detection system.

The experiments were conducted using Python and the Scikit-learn library in a Jupyter Notebook environment. The NSL-KDD dataset was utilized as a benchmark dataset and divided into training and testing sets using the KDDTrain+.txt and KDDTest+.txt files, containing 11,232 and 22,544 instances, respectively, including multiple categories of network traffic such as normal and various attack types.

First, the dataset was loaded and prepared for analysis. The selected machine learning algorithms Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN), were then trained independently using the training dataset.

After training, each model was evaluated on the same testing dataset to ensure a fair comparison. The performance evaluation was carried out using standard metrics, including Accuracy, Precision,

http://www.doi.org/10.62341/shlc2687

Recall, F1-score, and Inference Time. The F1-score was specifically considered to provide a balanced evaluation between precision and recall, which is particularly important due to the imbalanced nature of intrusion detection datasets.

Finally, the results obtained from the three models were analyzed and compared in terms of detection performance and computational efficiency, highlighting the strengths and limitations of each algorithm for real-time intrusion detection systems.

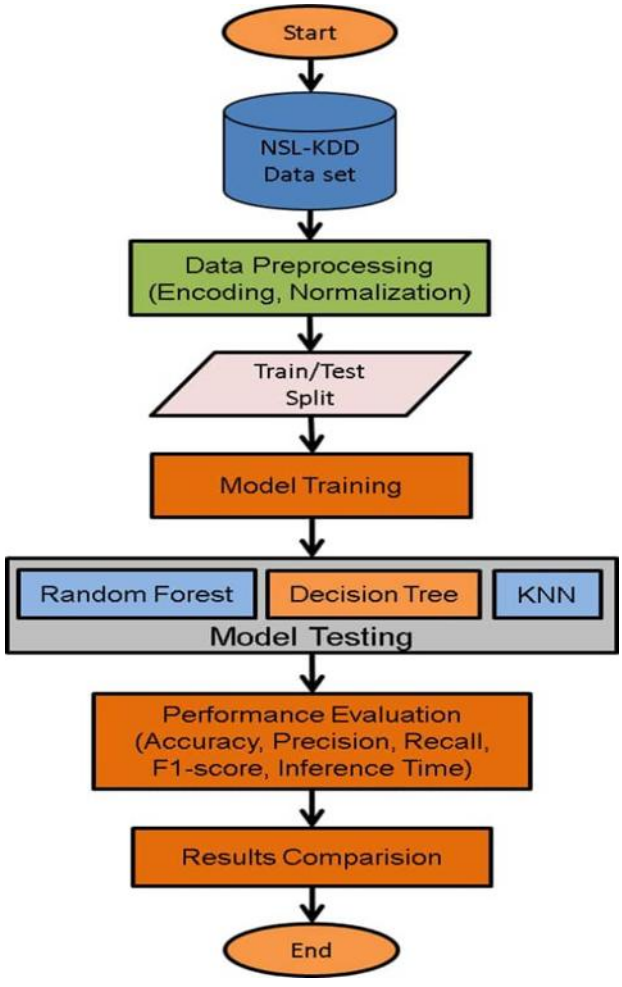

Figure 1. Overall methodology flowchart for intrusion detection model evaluation

http://www.doi.org/10.62341/shlc2687

7- Results and Discussion

7.1- Results

7.1.1- Decision Tree (DT):

A tree-based classifier that uses hierarchical decision rules for classification.

After applying the algorithm, the results were as shown in the following table :

Table 1. Decision Tree (DT)

Evaluation Results	
Accuracy	0.8100
Precision	0.9670
Recall	0.6882
F1-score	0.8048
Inference time	0.2397
Test samples	22544

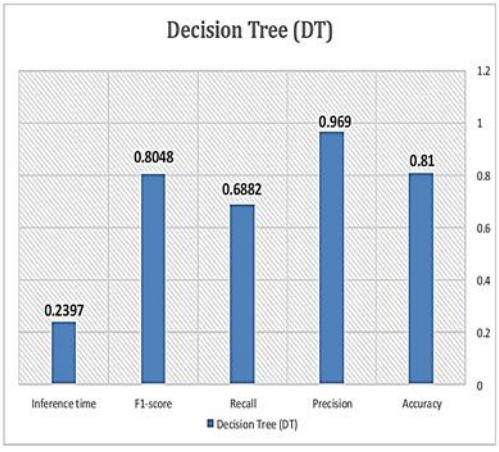

Figure 2. Performance metrics of the (DT

Table 1. shows that the inference time is fast and Accuracy and Precision are good although the Recall value is relatively low performance, the achieved F1-score (0.8048) indicates a good balance between precision and recall, this confirming the robustness of the Decision tree model.

Decision Tree

TP: 8832
TN: 9428
FP: 283
FN: 4001

---

http://www.doi.org/10.62341/shlc2687

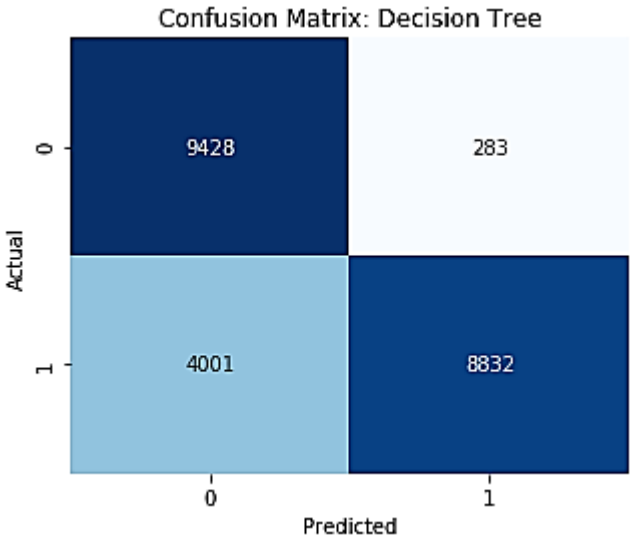

Figure 3. Decision Tree Confusion Matrix

7.1.2- Random Forest (RF):

An ensemble learning method that combines multiple decision trees to improve classification accuracy and robustness. The decisions of all the decision trees generated within the forest are aggregated, and a vote makes the classifying decision of most of the trees [10].
After applying the algorithm, the results were as shown in the following table:

Table 2. Random Forest (RF)

Evaluation Results	
Accuracy	0.7636
Precision	0.9677
Recall	0.6049
F1-score	0.7445
Inference time	2.8336
Test samples	22544

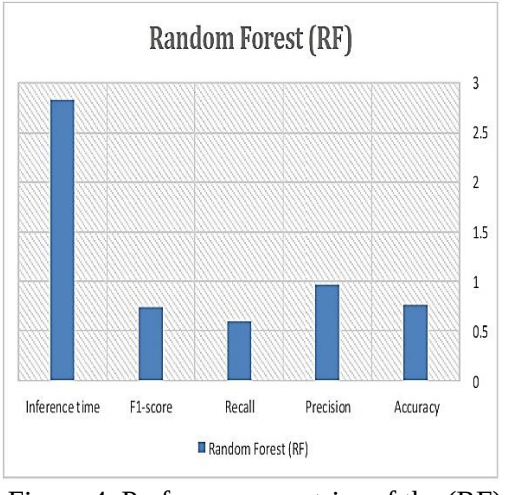

Figure 4. Performance metrics of the (RF)

http://www.doi.org/10.62341/shlc2687

Table 2. shows that the Accuracy and Precision have good performance, but Recall has low performance with reasonable inference time.

Despite achieving the highest precision, the lower recall value resulted in a reduced F1-score (0.7445), indicating that some attack instances were missed.

Random Forest

TP: 7763
TN: 9452
FP: 259
FN: 5070

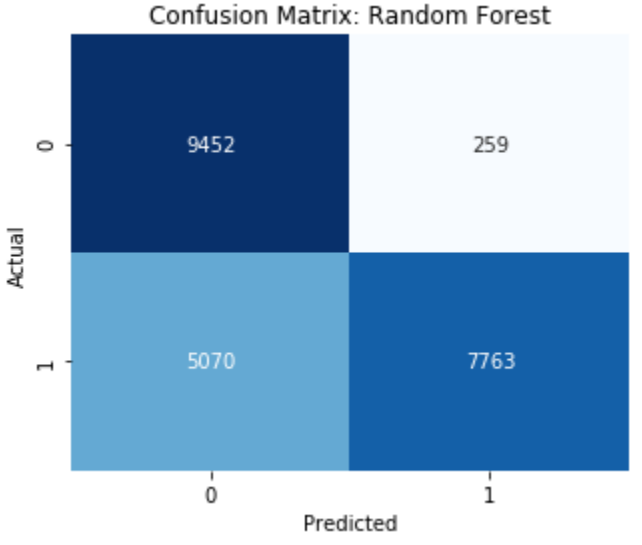

Figure 5. Random Forest Confusion Matrix

7.1.3- K-Nearest Neighbors (KNN):

A distance-based classifier that assigns labels based on the majority class of neighboring samples.

After applying the algorithm, the results were as shown in the table 3.

Table 3. shows that the Accuracy and Precision have good performance, but Recall has low performance with longer inference time.

http://www.doi.org/10.62341/shlc2687

Table 3. K-Nearest Neighbors(KNN)

Evaluation Results	
Accuracy	0.7652
Precision	0.9204
Recall	0.6431
F1-score	0.7572
Inference time	17.432366
Test samples	22544

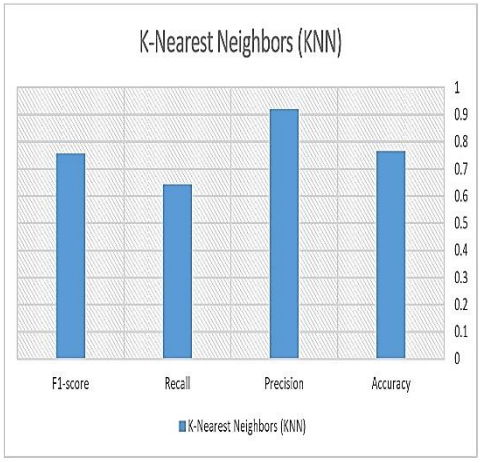

Figure 6. Performance metrics of the (KNN)

The F1-score of (0.7572) reflects moderate detection performance.

K-Nearest Neighbors

TP: 8253
TN: 8997
FP: 714
FN: 4580

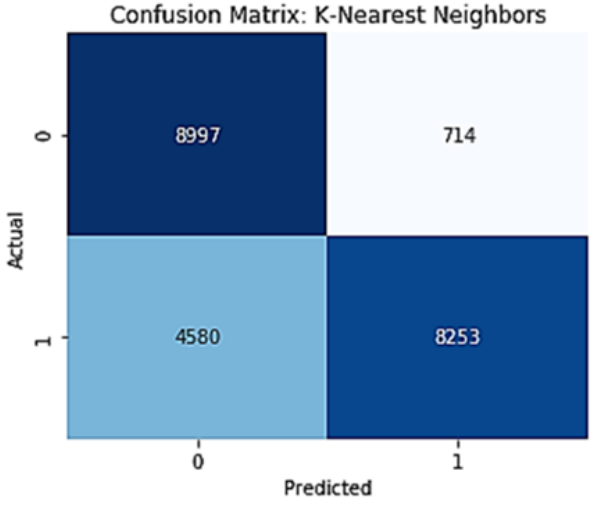

Figure 7. K-Nearest Neighbors Confusion Matrix

http://www.doi.org/10.62341/shlc2687

The inference time of the KNN model was not clearly visible in the figure 6. due to its significantly larger value compared to the other metrics; however it is reported explicitly in the corresponding table and does not clearly appear as in the figure 8.

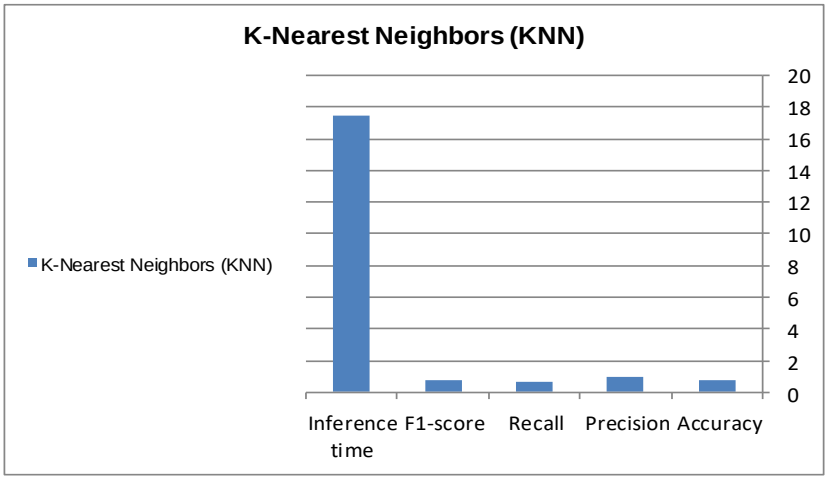

Figure 8. Performance metrics of the (KNN) with inference time

We have summarized the performance metrics of the three models in the following table.

Table 4. Comparison of DT, RF and KNN performance metrics

Model	Accuracy(%)	Precision	Recall	F1-score	Inference Time (sec)
Decision Tree	**0.8100**	**0.9670**	**0.6882**	**0.8048**	**0.2397**
Random Forest	**0.7636**	**0.9677**	**0.6049**	**0.7445**	**2.8336**
K-Nearest Neighbours	**0.7652**	**0.9204**	**0.6431**	**0.7572**	**17.432366**

Table 4. summarize a comparison of the three evaluation models based on the metrics mentioned above, allowing the identification of the best and most appropriate model. The three models were tested on the same dataset.

http://www.doi.org/10.62341/shlc2687

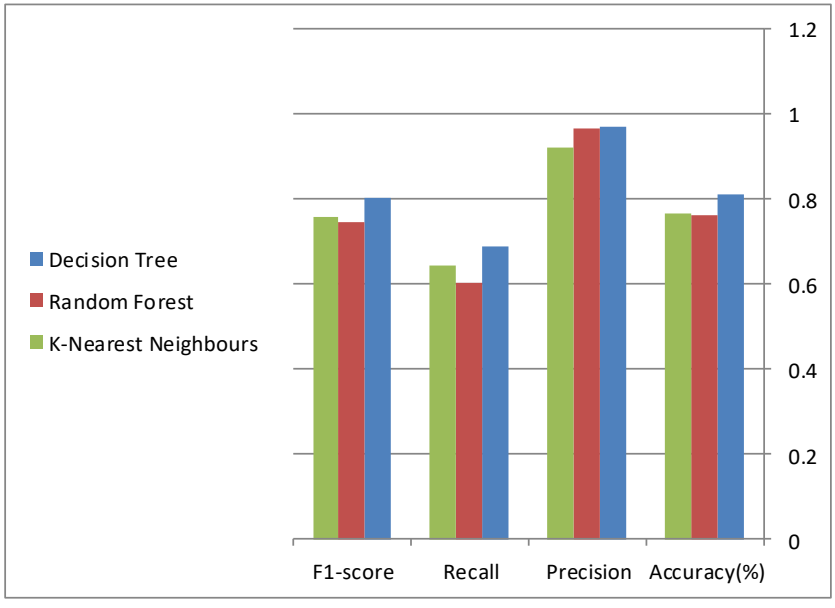

Figure 9. Performance metrics of the (DT, RF, KNN) without Inference time

The inference time was not included in figure 9 because the inference time for the k-Nearest Neighbor model is large compared to other models and does not clearly appear as in the Figures 10.

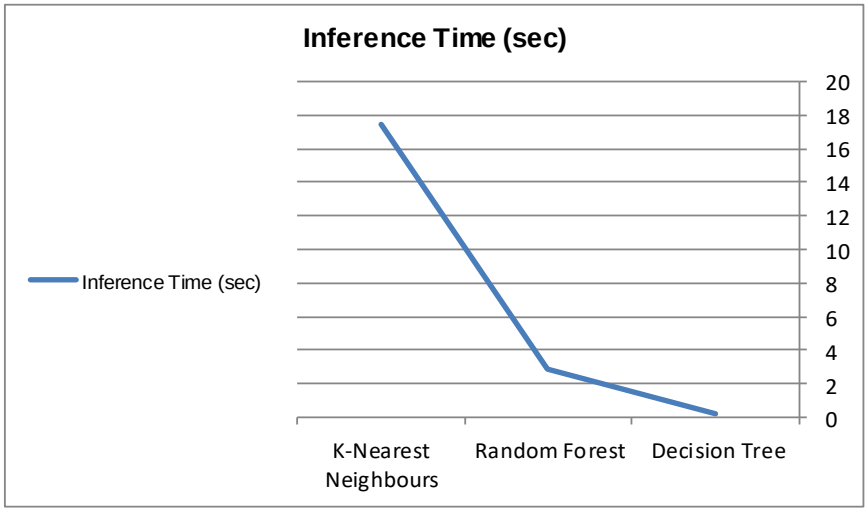

Figure 10. Inference time metric of the (DT, RF, KNN)

Figure 11. Contains all models with their metrics.

http://www.doi.org/10.62341/shlc2687

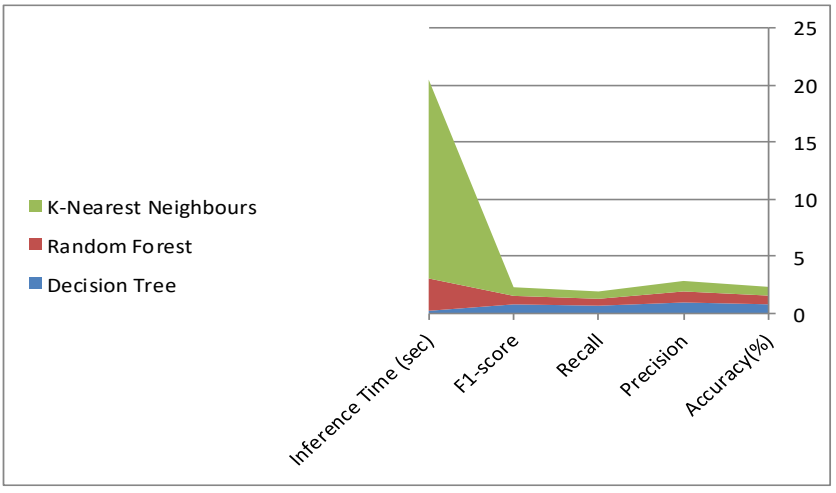

Figure 11. Performance metrics of the (DT, RF, KNN)

7.2- Analysis

The F1-score provides a more reliable comparison between models among all evaluation metrics, based on this metric the Decision Tree achieved the highest F1-score (0.8048) which demonstrating the most balanced performance, also achieved the highest recall (0.6882) and extremely low inference time (0.2397 sec), demonstrating suitability for real-time applications. Its accuracy (0.81) and precision (0.967) are also strong, providing a balanced performance.

Random Forest achieved the highest precision (0.9677), indicating reliable positive predictions with fewer false alarms. However, recall is lower (0.6049), meaning some attacks may be missed. Its inference time (2.8336 sec) is longer than Decision Tree but acceptable for offline or near real-time analysis.

K-Nearest Neighbours (KNN) showed moderate accuracy and recall (0.7652, 0.6431) with high precision (0.9204). The extremely high inference time (17.432366 sec) makes it impractical for real-time deployment.

7.3- Discussion

The experimental results demonstrate clear differences in performance among the three supervised machine learning algorithms, Decision Tree (DT), Random Forest (RF), and K-Nearest Neighbors (KNN) when applied to the NSL-KDD dataset.

Decision Tree achieved the fastest inference time, making it highly suitable for real-time intrusion detection systems where rapid

http://www.doi.org/10.62341/shlc2687

response is critical. Its balanced accuracy, precision, and recall indicate reliable overall performance with low computational cost.
Random Forest achieved the highest precision, which reflects its strong ability to correctly identify attack instances with minimal false alarms. However, its lower recall indicates that some attack instances may not be detected. The higher inference time compared to Decision Tree suggests that RF is more suitable for offline analysis or environments where detection accuracy is prioritized over speed.
KNN demonstrated acceptable accuracy and high precision; however, its extremely high inference time makes it impractical for real-time intrusion detection systems. This limitation is due to its instance-based learning nature, which requires distance computation with all training samples during classification.
The including of F1-score strengthens the evaluation.
Overall, the results confirm that there is a trade-off between detection performance and computational efficiency. Decision Tree is more suitable for real-time deployment, Random Forest is preferable for high-precision detection scenarios, and KNN is more appropriate for offline evaluation and benchmarking purposes, but not suitable for real-time systems due to high inference time.

8- Conclusion

This paper presented a comparative evaluation of Decision Tree, Random Forest, and K-Nearest Neighbors algorithms for network intrusion detection using the NSL-KDD dataset. The models were assessed based on accuracy, precision, recall, and inference time.
The results indicate that Decision Tree provides the fastest inference time with balanced detection performance, making it highly suitable for real-time intrusion detection systems. Random Forest achieved the highest precision, making it effective for minimizing false positives, although at the cost of increased inference time. KNN, despite acceptable accuracy and precision, proved impractical for real-time deployment due to its high computational cost.
Considering both F1-score and inference time, Decision tree demonstrated the most balanced and practical performance for real-time intrusion detection systems.

http://www.doi.org/10.62341/shlc2687

Overall, the study highlights the importance of balancing detection performance and computational efficiency when selecting machine learning models for practical intrusion detection applications.

9- Future Work

Future work may include evaluating additional machine learning and deep learning algorithms, applying feature selection and dimensionality reduction techniques, and testing the models on real-world and contemporary network traffic datasets.

Extend the study to include additional machine learning algorithms such as Support Vector Machines or neural networks for a more comprehensive analysis.

Explore deep learning or adaptive learning techniques for enhanced

Extend the study to include additional algorithms such as SVM and neural networks.

Improve performance using Hyper parameter Tuning and Feature Selection.

Evaluate the models on real and contemporary network data.

Explore deep learning and adaptive learning for detecting complex intrusions.

10- References

[1] Zeebaree, S. R. et al.: Impact analysis of SYN flood DDoS attack on HAProxy and NBL cluster-based web servers, Indones, J. Electr. Eng. Comput. Comput. Sci, 2020, Vol. 19, No. 1, pp. 510-517

[2] Kaggle, Accessed Dec. 16, 2019 [Online] Available: https://www.kaggle.com/datasets/hassan06/nslkdd

[3] K. D. Kim and P. K. Chong, “Intrusion detection based on machine learning techniques,” Journal of Network and Computer Applications, vol. 93, pp. 1–10, 2017.

[4] Kasim, : An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks. Computer Networks, 2020, Vol. 180, pp. 107390

[5] B. Budler and R. Ajoodha, “Comparative Analysis of Deep Learning Models for Network Intrusion Detection Systems,” in Proc. IEEE 2nd Conference on Information Technology and Data Science (CITDS), Debrecen, Hungary, 2022, pp. 45–50.

http://www.doi.org/10.62341/shlc2687

[6] V. Pai, D. & N. D. Adesh, (2021), "Comparative analysis of Machine Learning algorithms for Intrusion Detection," IOP Conference Series: Materials Science and Engineering, vol. 1013, no. 1, p. 012038, 2021, doi:10.1088/1757-899X/1013/1/012038.

[7] Chavan, N., et al.: "DDoS Attack Detection and Botnet Prevention using Machine Learning," 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2022, pp. 1159-1163

[8] Abu Al-Haija, Q., and Zein-Sabatto, S.: An efficient deep-learning-based detection and classification system for cyber-attacks in IoT communication networks, Electronics, 2020, Vol. 9, No. 12, pp. 2152

[9] Logeswari G, Bose S, Anitha T. An intrusion detection system for SDN using machine learning. Intelligent Automation & Soft Computing 2023; 35 (1): 867-880. https://doi.org/10.32604/iasc.2023.026769

[10] Nhat-Duc, H. and Van-Duc, T.: Comparison of histogram-based gradient boosting classification machine, random Forest, and deep convolutional neural network for pavement raveling severity classification. Automation in Construction, 2023, Vol. 148, pp. 104767

[11] Kaur, G., & Singh, V. (2021). "Machine learning-based anomaly intrusion detection for cybersecurity: A contemporary survey." International Journal of Information Security.

[12] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). "A survey of network-based intrusion detection data sets." Computers & Security.

[13] Alshamrani, T., & Alwan, J. K. (2020). "Network intrusion detection using deep and classical machine learning methods: A comparative study." Journal of Cyber Security and Mobility.

[14] Moustafa, N., & Slay, J. (2019). "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." Military Communications and Information Systems Conference (MilCIS).

http://www.doi.org/10.62341/shlc2687

[15] Das, S. et al.: DDoS Intrusion Detection Through Machine Learning Ensemble, 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Sofia, Bulgaria, 2019, pp. 471-477

[16] Ghaniyyat B. Balogun and others : Comparative Analysis of Various Machine learning Techniques Applied Towards Intrusion Detection in Computer Networks, Journal of Computer and Communication, Vol. 3, No. 2, pp. 31-54 , 2024

[17] Isa Aver and Murat Koca: Cybersecurity Attack Detection Model, Using Machine Learning Techniques, Vol. 20, No. 7, 2023

[18] Tavallaee, M. et al: A detailed analysis of the KDD CUP 99 data set, 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, 2009, pp. 1-6