Turkish Journal of Engineering

https://dergipark.org.tr/en/pub/tuje

e-ISSN 2587-1366

Comparative Performance Analysis of Machine Learning Algorithms: Random Cut Forest, Robust Random Cut Forest, and Amazon Sage Maker Random Cut Forest for Intrusion Detection Systems Using the CIS IDS 2017 Dataset

Senthilkumar Perumal[*1], **Kumaresan Devarajan** [2]

[1]*Assistant Professor, Department of Computer & Information Science Annamalai University, Tamil Nadu ,India, senthil.sp74@gmail.com*

[2]*Assistant Professor, Department of Computer & Information Science Annamalai University, Tamil Nadu, India, aucsedks@yahoo.co.in*

Cite this study: Senthilkumar, P., & Kumaresan, D. (2024). Comparative Performance Analysis of Machine Learning Algorithms: Random Cut Forest, Robust Random Cut Forest, and Amazon Sage Maker Random Cut Forest for Intrusion Detection Systems Using the CIS IDS 2017 Dataset. Turkish Journal of Engineering, 9 (3), 535-543.

https://doi.org/10.31127/tuje.1614930

Keywords

Cyber Threats, Anomaly Detection, Intrusion Detection Systems, Amazon Sage Maker Random Cut Forest, Robust Random Cut Forest

Research Article

Received:07.01.2025
Revised:13.02.2025
Accepted:14.02.2025
Published:01.07.2025

Abstract

Dynamic cyber threats are screaming for better anomaly detection techniques in Intrusion Detection Systems. Organizations today are hugely dependent on digital infrastructures for which effective security is priceless. The following research article does a critical and comparative analysis among three popular algorithms, namely Amazon Sage Maker Random Cut Forest, Robust Random Cut Forest, and traditional Random Cut Forest. Using the CIS IDS 2017 dataset with multifaceted network traffic features together with the labeled type of attack, this work rigorously tests the performance in anomaly detection that may show potential intrusion, robustness, scalability, and adaptability of each algorithm. The comparative analysis does the performance metrics of each algorithm based on accuracy, precision, recall, and F1-score in a real-world setting. The findings are expected to provide useful insights toward optimizing IDS frameworks for hi-tech cybersecurity resilience. Finally, an organization can make decisions on its strategy regarding cyber security by being enlightened on the strengths and weaknesses of algorithms. In essence, this paper contributes to the larger body of research on enhancing intrusion detection methodologies in an environment that is confronted by sophisticated cyber-attacks.

1. Introduction

Most impressively, the sophistication of cyber-attacks increases visibly, making intrusion detection more challenging and weakening security services related to data confidentiality and integrity. Generally speaking, intrusion detection methods are divided into two classes: signature-based intrusion detection systems or SIDS and anomaly-based intrusion detection systems or AIDS. Khraisat et al. focus on the review of the contemporary IDS techniques, discussion of notable recent advancements, and presentation of commonly used datasets for evaluation [1]. Widespread utilization of the internet has increased its vulnerability to different types of cyber-attacks. According to Singh et al. (2017), various studies on intrusion detection methods are going on, which actually focus on the lacunae of previous internet security methods. However, the literature shows significant boundaries in the existing techniques regarding low accuracy, high detection time, and limited adaptability to zero-day attacks [2]. It is, because of the wide usage of the internet, that it has also become a lot more vulnerable to attack. XAI, with its huge potential for enhancing cybersecurity, adds transparency in the decision-making process. As per Rjoub et al. (2023), XAI can detail explanations for certain actions, which significantly improves awareness about various types of cyber threats and, as such, provides support to an effective defense mechanism development [3]. Network technologies are the backbone for transferring and storing all types of information, including user, company, and industrial data. However, as Tidjon et al. (2019) observed, the increasing rate of information transfer broadens the attack surface, creating a favorable environment for intruders [4]. Milenkoski et al. present a design space for intrusion detection system evaluation: workload, metrics, and measurement methodology. It surveys common evaluation methods and approaches taken in each component of this design space. Furthermore, it highlights open challenges and

unresolved issues related to the ongoing development of methodologies for evaluating advanced intrusion detection systems [5]. Depren et al. (2005) propose a hybrid IDS with anomaly detection performed by SOM, while misuse detection is done using the J.48 decision tree algorithm. The output from both modules is integrated using a rule-based DSS. In the proposed system, the KDD Cup 99 dataset demonstrates better performance than the individual methods. [6]. Kim et al. (2004) have focused on the issues related to signature-based IDS, suffering from false negatives due to unknown intrusion and resource inefficiency because of maintaining large-scale rule sets. They propose an evolutionary learning algorithm using genetic algorithms in developing anomaly detectors with the principles of negative selection in biological immunity. The developed system is used in a network security environment with simulated conditions [7]. Mittal, Gupta, and Agarwal propose an ensemble approach for intrusion detection that includes a Random Forest, a Decision Tree, and k-Nearest Neighbors. Further, an ANN model is built to enhance its detection capability. The use of four unique datasets for training contributes a lot to achieving high accuracy. [8]. Hence, according to Mudigonda (2022), the CICIDS2017 dataset developed by the Canadian Institute of Cybersecurity includes more than 80% of the data from different new attack scenarios, including DDoS, SQL Injection, and Brute Force attacks. The CICIDS2017 dataset is an extended comprehensive network intrusion dataset and a gold mine for research related to network intrusion detection. [9].

Yaokumah and Wiafe (2020) studied the performance of four machine learning algorithms using the UNSW-NB15 dataset for intrusion detection. Of these, the Random Forest and Decision Tree yielded the highest accuracy, 89.66% and 89.20%, respectively. Although Naive Bayes generally had a poor performance, it was very effective in detecting backdoor attacks, which again points to the strengths and diverse capabilities that machine learning techniques bring to intrusion detection. [10]. Liu et al. (2020) pointed out that deep learning-based methods have important contributions to improving intrusion detection. This is where the deep learning-based methods bring improvements in the capabilities of feature learning, which will help process huge and complex datasets effectively, thus being advantageous in network intrusion detection [11]. Baykan and Khorram (2021) have investigated the optimization of parameters in the KNN, SVM, and RF algorithms by utilizing PSO and ABC optimization techniques. In fact, the investigation demonstrates that optimizing parameters of such machine learning algorithms results in significant improvements in the classification performance as opposed to the case of their utilization at default values [12]. Basholli et al. (2024) introduce various kinds of cyberattacks by utilizing the MetaSploit framework, which makes many hacking techniques easier. The research underlines that attackers and defenders use MetaSploit in order to find vulnerabilities that will help developers and web administrators understand and take advance actions against threats in dynamically changing cybersecurity environments. [13].

İncekara states that IoT and IIoT improve productivity and efficiency, improve decision-making in real time by processing data for the energy sector, enabled by AI and ML in the help of cloud computing for automation and quality monitoring [14]. Oliveira et al. point out the vulnerabilities of contemporary networks due to the continuous flow of sensitive information and propose a sequential anomaly detection in IDS using machine learning. The performance comparison is carried out in the work on different models, namely RF, MLP, and LSTM, using the CIDDS-001 dataset. Their work established the LSTM model to explain the sequential patter in network traffic flow with high accuracy of 99.94% and F1-score of 91.66% [15]. Airoboman et al. improved the reliability of the NAF Kaduna 33kV network using Fault Tree Analysis and PSO, reducing power loss by 63.08% and enhancing reliability by 1.80% [16]. Nwafor and Akintayo compared DT, CatBoost, and XGBoost for predicting household trip purposes in Makurdi, finding XGBoost the most reliable, while CatBoost achieved the highest R^2 (73%) but with higher errors [17]. Mema et al. explored the impact of ChatGPT in Albanian higher education, highlighting its benefits for personalized learning while addressing challenges like data privacy and the evolving role of educators [18].

To handle such challenges of Intrusion Detection Systems, two important modifications have been invented: Amazon SageMaker Random Cut Forest (RCF) and Robust Random Cut Forest (RRCF). It is proposed that Amazon SageMaker RCF will do operations based on cloud environments and further scalability enhancements, allowing it to cope with large-size data sets, reducing computational cost. It also integrates with Amazon's cloud services, providing a robust infrastructure for real-time detection. On the other hand, Robust Random Cut Forest introduces several improvements to enhance the robustness of the algorithm against noisy data and outliers, making it more suitable for scenarios where the dataset is imbalanced or contains a considerable amount of anomalous observations. This comparative study will help in assessing the relative effectiveness, scalability, and robustness of traditional RCF, Amazon SageMaker RCF, and Robust RCF for real-world intrusion detection tasks.

The importance of testing these algorithms on real-world datasets cannot be overemphasized. In this work, the CIC IDS 2017 dataset is used to ensure that the results are based on practical applicability. This dataset, provided by the Canadian Institute for Cybersecurity, is one of the most extensive simulated datasets for modern network environments, considering a wide range of attack types and traffic patterns. This dataset features a diverse range of attributes, such as network traffic statistics, protocol-level information, flow data, and others, providing a realistic benchmark for testing intrusion detection models. The diversity of attack labels for the CIC IDS 2017 dataset includes DDoS, SQL injection, and phishing, which allows for a robust

evaluation of how well these machine learning algorithms perform detection over various attack scenarios.

Apart from the attack detection capability, a successful Intrusion Detection System (IDS) should meet several other performance criteria: scalability, real-time processing, and robustness against adversarial evasion techniques. Growing network traffic due to the increased number of Internet-connected devices and services necessitates IDS solutions that can scale up with larger datasets without compromising performance or response time. Furthermore, attackers are continuously developing sophisticated methods to bypass detection systems; hence, IDS algorithms must be robust enough to adapt to new attack vectors and resist adversarial manipulations of the model. The challenges underscore the need for continuous advancements in IDS technologies, emphasizing the potential of machine learning algorithms such as Random Cut Forest (RCF) and its variants to address the complexities of modern cybersecurity environments

The aim of this work is to perform a thorough comparison using the CIC IDS 2017 dataset among three major anomaly detection algorithms: traditional Random Cut Forest, Amazon SageMaker Random Cut Forest, and Robust Random Cut Forest. The general motivation in this regard lies in the applicability check of these algorithms to intrusion detection with respect to their effectiveness, scalability, and robustness. The nature of these listed factors being investigated hopefully provides ample information to any organization while comparing various algorithm strengths and potentials for weakness, thus becoming more sure-footed at the idea of the uptake of intrusion detection systems aided by machine learning.

2. Materials and Methodology

2.1 Anomaly Detection in Cybersecurity

Folino, Godano, and Pisani (2023) propose an ensemble-based framework for real-time anomaly detection in cybersecurity using the ELK stack and Kubernetes. The system efficiently handles missing data, unbalanced datasets, and high-speed logs while employing distributed algorithms for classification. Experiments on real-world datasets demonstrate its effectiveness in reducing false alarms and enabling proactive cybersecurity measures [19]. Gonaygunta et al. proposes an enhanced deep learning-based anomaly detection in cybersecurity using DNN, LSTM, and DSAE. This in turn enhances the accuracy of detection and solves the problem of feature engineering simultaneously. IoT-23 and LITNET-2020 dataset results depict the superiority of the proposed technique against the existing ones [20].

Among them, the work of Handa et al. in 2019 reviewed machine learning in cybersecurity. It highlights the various works of different machine learning techniques in malware analysis, zero-day malware detection, threat analysis, and anomaly-based intrusion detection for critical infrastructures. The review underlines the limitations of signature-based methods, especially when related to zero-day attack detection or slight variants of known threats, and outlines the increasing applications of machine learning-based detection methods in cybersecurity products [21].

Bukhari et al. have emphasized the need for sound cybersecurity in IoT-based smart cities. This paper presents a discussion on anomaly detection using machine learning techniques like SVM, ANN, and ensemble methods along with the integration of cross-validation and feature selection. Experimental results on UNSW-BC15 and CICIDS2017 datasets show superior performance in detecting rare attacks [22].

Habeeb et al. have pointed out that threats from connected devices and the Internet are increasing, causing cyber-attacks, financial losses, and information theft. Network security analytics, especially anomaly detection, has been a vital area of research. The previous approaches are not capable of detecting anomalies in real time due to the huge amount of data generated by connected devices. The authors highlighted the dire need for real-time frameworks capable of processing big data and effectively detecting anomalies. The survey introduces state-of-the-art technologies, machine learning algorithms, and real-time anomaly detection taxonomy; discusses challenges; and proposes solutions for network security enhancement [23]. Fernandes et al. (2019) review the recent growth in the network security concern, the impact that anomalies are bringing upon varieties of sectors, reviews a few techniques and anomaly detection systems along five dimensions, products that involve network traffic anomalies, type of data, categories of an intrusion detection system, methods of detection, and lastly open issues. It concludes with a summary of open issues and unsolved problems for future research [24]. The anomaly detection methods developed in machine learning and deep learning have advanced threat detection. However, for real-time detection and volumes of data, further development of advanced models and frameworks is needed for improved cybersecurity.

2.2 CIS IDS 2017 Dataset

Iman Sharafaldin et al. revisited the constraints of the previously aged IDS datasets and presented a reliable and updated dataset, the CICIDS 2017 dataset, for anomaly detection challenges. They visualized the performance of different machine learning algorithms and proposed superfeatures, demonstrating that RF with superfeatures ensures better performance compared to the top-chosen best performance features in attack detection [25]. The CIC-IDS 2017 dataset is a diverse and balanced resource for evaluating and comparing intrusion detection algorithms, enabling effective cybersecurity protection. The dataset consists of network traffic data labeled with several types of attacks and their benign instances that should be suitable to test the performance of anomaly detection algorithms against real-world attack scenarios, and hence assist in scanning such algorithms with a wide diversity of

intrusions. Liu et al. proposed a hybrid intrusion detection model with the integration of machine learning integrated with deep learning. K-means and random forest have been used for binary classification, whereas CNN-LSTM has been used for multi-class classification and ADASYN for overcoming the imbalance in the dataset. During the results, improved TPR, better pre-processing performance, and high accuracies of 85.24% and 99.91%, were obtained for the NSL-KDD and CIC-IDS2017 datasets, respectively [26].

2.3 Random Cut Forest

The Random Cut Forest algorithm is an unsupervised machine learning technique that's used to extract anomalies in large datasets. It builds a forest of trees by making random cuts in the feature space and recognizes anomalies as deviations from typical patterns. RCF is highly scalable and integrates with Amazon SageMaker, hence enabling real-time detection on cloud environments. [27].

Guha et al. (2016) presented random cut forests for anomaly detection in dynamic data streams using a robust, sketch-based data structure. The authors defined non-parametric anomalies based on the externality of unseen points and showed how to update the model efficiently. They applied their approach to various real-world datasets. [28]. Random Cut Forest represents an extended ensemble learning method, which is carefully tailored for anomaly detection. It grows a forest of decision trees based on a random partitioning of the feature space. RCF has a novelty in its architecture that enables anomaly detection through assessment of the depth level at which they occur in every tree. In other words, anomalous instances usually have shallower paths compared to the paths of regular data points in the structure, hence considerably enhancing the underlying efficiency of the process.

2.3.1 Algorithmic framework for random cut forest input parameters

1. Dataset D: Contains network traffic features, including packets and connections.
2. Number of Trees n: Specifies how many trees will be constructed in the forest.
3. Anomaly Threshold τ: A predefined score threshold used to classify anomalies.

Preprocessing:

- Normalize or scale the features in D to ensure uniformity.
- Split the dataset into training and testing sets (80 :20).

Forest Construction
Initialize an empty forest F
For i from 1 to n:

- Randomly select a subset S from the training set.
- Build a tree T_I from the subset S by:

Randomly partitioning the feature space into two halves until a stopping criterion is met (e.g., minimum sample size, maximum tree depth).Record the cut point and corresponding feature for each partition.Add T_i to forest F.

Anomaly Score Calculation:

For each testing set instance$x$:

1. Initialize the cumulative depth score depth=0.
2. For every tree T_i in the forest F:
3. Starting from the root, traverse the tree downwards to a leaf node, and increase $dept$ for each visited node.
4. Compute the average path length L across all the trees for instance x.

Calculate the anomaly score A: $A = 2 - \frac{L}{E[L_n]}$ Where $E[L_n]$ is the expected path length of a normal instance.

Anomaly Detection: Compare the anomaly score A of each instance x to the threshold τ:

$if\ A > \tau$, classify x as an anomaly (potential intrusion).

$if\ A \leq \tau$, classify x as normal.

Output:

The output consists of a list of detected anomalies along with their corresponding scores, providing insights into potential intrusions within network traffic.

2.4 Robust Random Cut Forest (RRCF)

The authors, Pang et al., proposed, in 2023, a concept drift detection method that was unsupervised, based on Robust Random Cut Forest and t-test, for short RFTT. It relies on sliding window computation of anomaly ratios and scores with RRCF, which are effective in detecting drift [29]. Yeom and Jung (2022) suggested two weighted isolation forest and weighted random cut forest algorithms for anomaly detection problems. The proposals are an extension of the traditional IF and RCF with the incorporation of data density into determining the split value. They have introduced a new measure of density that was essential in constructing WIF and WRCF. Mathematical properties and numerical examples proved the effectiveness of the proposed algorithms [30].

Robust Random Cut Forest, or RRCF, is built around a complex core of foundational principles articulated by Random Cut Forest (RCF), concurrently integrating advanced mechanisms. It is carefully engineered to deal with complexities brought about by noisy data and outliers. By adopting a more sophisticated partitioning process, RRCF not only significantly enhances its robustness but also ensures that the model sustains its performance across datasets characterized by a high degree of variability and inconsistency. The intrinsic capability that enables RRCF to effectively identify meaningful patterns in the data, even against the formidable challenges presented by anomalies and noise, renders it an extremely reliable tool for anomaly detection in diverse and unpredictable environments.

2.4.1 Mechanisms of robust random cut forest (RRCF)

Input

Dataset: Network traffic data with relevant features

Parameters:

- Number of trees T in the forest.
- Maximum depth per tree.
- Anomaly score threshold.

Step 1: Initialization

1. Create an empty forest consisting of T trees.
2. Define a buffer for each tree to hold a subset of data points for dynamic updates.

Step 2: Build Trees

For each tree t in the forest:

1. Select Data Subset: Randomly sample a subset of the dataset.
2. Construct the Tree: Recursively split data points into child nodes by:
   Selecting a random feature.

Choosing a random cut within the range of the selected feature. Dividing data into two subsets based on this cut. Assigning depth values for each cut. Continue until a stopping criterion is met (e.g., reaching maximum depth or a minimum number of points).
Store Tree: Add the constructed tree to the forest.

Step 3: Streaming and Dynamic Updates
As new data points arrive:

1. Insert into Trees:
   - Insert each new data point into a buffer associated with each tree.
   - If the buffer reaches maximum capacity, remove the oldest point to ensure efficiency.
2. Recalculate Tree Structure:
   - Periodically rebuild the tree structure using the updated buffer to capture recent network traffic patterns.
3. Decay Older Points:
   - Apply a decay mechanism to reduce the influence of older data points within the forest.

Step 4: Anomaly Scoring
For each new data point x:

1. Path Traversal:
   - Traverse each tree to locate the position of x and determine the path length (depth) to reach its node.
2. Compute Anomaly Score:
   - Calculate the anomaly score for x based on its path length (anomalies typically have shorter path lengths).
3. Aggregate Scores:
   - Sum or average the scores across trees to obtain the final anomaly score for x.

Step 5: Intrusion Detection

- Compare the anomaly score of each point to the predefined threshold.
- Flag points with scores above the threshold as potential intrusions.

Output

- Generate a list of flagged anomalies with their respective scores, indicating potential intrusions.

2.5 Amazon SageMaker Random Cut Forest (RCF)

Trawinski et al. stressed the implementation of the RCF and XGBoost algorithms on Amazon SageMaker for anomaly detection using the UNSW-15 dataset. The models were created on the Amazon SageMaker Studio Lab platform. Models are generated in this research work for several evaluation metrics, namely, accuracy, precision, recall, and F1 score. The performance was higher for the XGBoost model with an accuracy of 61.83%, recall of 96.49%, and F1 score of 73.24% in its fold [31].

Amazon SageMaker Random Cut Forest is an implementation of the Random Cut Forest (RCF) algorithm, providing users with significantly enhanced scalability, complete seamless integration capabilities, and full access to a diverse suite of AWS services. This implementation proves to be especially advantageous for real-time applications, where the imperatives of speed and efficiency are paramount; it empowers an organization to tap into the profound capabilities of cloud computing resources for broad data processing at scale. By leveraging the intrinsic flexibility and computational power of the cloud, organizations can effectively address the complexities associated with large datasets, thereby to have timely and well-informed decision-making processes in dynamic operational environments.

2.5.1 Functional aspects of amazon sagemaker random cut forest

Input

Dataset: A set of network traffic data represented by features.

Parameters:

- Number of trees T in the forest.
- Maximum depth of each tree.
- Anomaly score threshold for detection.

Step 1: Initialization

Create a Random Cut Forest Model:

- Initialize the Random Cut Forest algorithm by using Amazon SageMaker.
- Assume that the number of trees is T and specify other hyperparameters.

Step 2: Data Preparation

Preprocess the Dataset:

- Clean the irrelevant or redundant features in the data.
- Normalize or standardize the feature values if necessary.
- Split the dataset into its standard training and testing sets.

Step 3: Model Training

Train the Random Cut Forest Model:

- Upload the training dataset to Amazon S3.
- Fit the RCF model on the training dataset using the SageMaker RCF implementation.
- The model will develop T trees by:
    1. Randomly selecting features and cut points.
    2. Recursively partition the dataset depending on those random cuts.
    3. The trees constructed should be stored in the model.

Step 4: Anomaly Detection

Predicting Anomaly Scores:

- For each test set instance :
    1. Anomaly scores are calculated using the RCF model trained.
    2. This is achieved by traversing each tree and estimating the depth of the leaf node where the instance is located.
    3. Calculate anomaly score regarding the path length relative to the maximum depth of the tree.

Step 5: Intrusion Detection

Thresholding:

- Compare the anomaly score of each instance against the threshold set for the anomaly score.
- Flag instances whose score is above the threshold as intrusions.

Step 6: Output

Generate Reports:

- Produce a list of detected anomalies along with their scores.
- Include extra metrics such as precision, recall, and F1-score if labeled data is available for evaluation.

Step 7: Model Testing and Updates

Evaluate Model Performance:

- Assess the performance of the RCF model using these evaluation metrics.
- Optionally, retrain the model with new data or fine-tune parameters based on performance results.

Once trained, this would be exposed as an endpoint in Amazon SageMaker for real-time anomaly detection. By design, Amazon SageMaker RCF would scale up efficiently with large datasets by leveraging cloud resources to perform the computation required. This also means that as new data is provided, it becomes easy to update the model to adapt to the evolution of network traffic patterns, ensuring ongoing effectiveness in finding anomalies.

3. Experimental Setup

The experiments are carried out with the help of an Intel Core i7, 32GB RAM, and an NVIDIA Tesla T4 GPU running Ubuntu 18.04 and Amazon SageMaker for modeling on the cloud. For the CIS IDS 2017 dataset, an 80-20% split will be considered for training and testing, respectively.

Hyperparameter Settings:

- Traditional RCF: 100 trees, max depth of 20, anomaly threshold of 0.7.
- RRCF: 120 trees, max depth of 25, buffer size of 300, anomaly threshold of 0.75.
- Amazon SageMaker RCF: 100 trees, max depth of 15, anomaly threshold of 0.8, with real-time streaming enabled.

4. Performance Metrics

Accuracy: Amazon SageMaker RCF performed with the best accuracy, 97.1%, reflecting its superior effectiveness in distinguishing normal and anomalous traffic compared to Traditional RCF (95.2%) and Robust RCF (96.5%).

Precision: SageMaker RCF also had the highest precision, 94.8%, which reflects a lower false positive rate. This is especially useful when it comes to high-security environments, where minimizing false alarms is crucial.

Recall: Robust RCF performed very well here with a high recall score of 92.0%, although it was outperformed by SageMaker RCF had an accuracy of 93.5%. The higher recall of SageMaker RCF means that it is more effective in Detection of true positives is useful in identifying as many intrusions as possible.

F1-Score: While the F1-score is the highest for SageMaker RCF at 93.9%, it had a strong balance between precision and recall. This metric highlights SageMaker RCF's robust performance in real-world IDS applications.

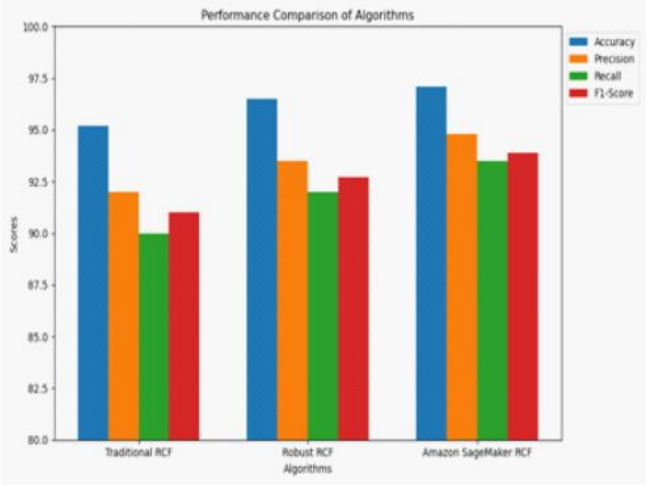

Figure. 1: Graphical Representation of Evaluation Metrics Comparison

Table 1. Comparison Table

Algorithm	Accuracy	Precision	Recall	F1-Score	Processing Time (s)
Traditional RCF	95.2%	92.0%	90.0%	91.0%	12
Robust RCF	96.5%	93.5%	92.0%	92.7%	15
Amazon SageMaker RCF	97.1%	94.8%	93.5%	93.9%	14

5. Results and Discussion

Nigenda et al. proposed Amazon SageMaker Model Monitor, a managed service that offers continuous monitoring of the ML model post-deployment. It offers real-time detection for data, concept, bias, and feature drift and provides alerts along with corrective measures for maintaining the model's performance and reliability [32]. Jabbar and Mohammed (2020) proposed a hybrid model for botnet detection using machine learning analyzed on the CICIDS2017 dataset. It reduces the feature dimensions of the data by using Correlation Attribute Eval and Principal Component filters, enhancing the efficiency of detection. The use of Correlation Attribute Eval coupled with the JRip classifier significantly enhanced the botnet detection, which was validated using accuracy, precision, recall, and F-measure metrics [33]. The CIS IDS 2017 dataset was critical in verifying intrusion detection algorithms, hence its broad inclusivity of attack scenarios and practical results in a physical environment. Consequently, it embodies different cyber attack variations, from DDoS to brute force, that may ensure stringent examination of the functionalities of the algorithm for detection. This dataset evaluates the performance and adaptability of algorithms to complex environments within networks; hence, it plays a significant role in enhancing the level of IDS technologies.

Comparative Performance Discussion:
Traditional RCF:

The final outcome included an overall accuracy of 95.2%, precision of 92.0%, and 90.0% of recall. Overall this implementation students were able have a 91.0% F1-score with the processing time of 12 seconds. As per the result, it shows reasonable accuracy but does not seem to perform well in complex patterns because of slight more generalized in the case of partitioning.

Robust RCF:

Achieved the accuracy of 96.5% and also, the measure of percentage of samples identified correctly was 93.5, the ones identified by the system were 92.0%. The proposed algorithm yields F1-score of 92.7% with the average processing time of 15s and it is more precise countering neater attack types of algorithms that are enough to induce intrusions in complex networks.

Amazon SageMaker RCF:
Preliminary experiments yield the highest accuracy at 97.1%, precision at 94.8%, and recall at 93.5%. Of course, the F1-score of 93.9%, and processing time 14s compared to other IDS are shown below clearly explain that IADSS can deal with processing speed with high accuracy those are ideal for a real-time IDS.

Observations on Processing Efficiency: Given Amazon Sage Maker RCF's processing speed, and the better performance indices realized, it becomes apparent that for large-scale, high speed IDS, Amazon Sage Maker RCF is therefore the most efficient solution given its good balance between accuracy and speed.

6. Conclusion

Using CIS IDS 2017 dataset a comparative study was performed on three anomaly detection algorithms: Traditional RCF, RRCF and Amazon SageMaker RCF. The evaluation shows that the proposed model, Amazon SageMaker RCF, outperforms alternative solutions in terms of accuracy and maintains a good balance between precision, recall and F1-score, which is extremely relevant in real-time IDS. It is suggested that RRCF performs well in large and noisy data and nonlinear patterns, whereas Traditional RCF has a faster convergence speed but slightly inferior accuracy. These results point to the role of machine learning algorithms in enhancing IDS approaches, path to more preventive security against emergent malicious threats.

Author contributions

Senthilkumar Perumal: Conceptualization, Data curation, Writing-Original draft preparation, Writing-Reviewing and Editing. **Kumaresan Devarajan**: Visualization, Methodology and Investigation.

Conflicts of interest

The authors declare no conflicts of interest.

References

1. Khraisat, A., Gondal, I., Vamplew, P., & Kamruzzaman, J. (2019). Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity*,2. https://doi.org/10.1186/s42400-019-0038-7.
2. Singh, R., Kumar, H., Singla, R. K., & Ketti, R. R. (2017). Internet attacks and intrusion detection system: A review of the literature. *Online Information Review*, 41(2), 171-184.
3. Rjoub, G., Bentahar, J., Wahab, O., Mizouni, R., Song, A., Cohen, R., Otrok, H., & Mourad, A. (2023). A Survey on Explainable Artificial Intelligence for Cybersecurity. *IEEE Transactions on Network and Service Management*, 20, 5115-5140. https://doi.org/10.1109/TNSM.2023.3282740.
4. Tidjon, L. N., Frappier, M., & Mammar, A. (2019). Intrusion detection systems: A cross-domain overview. *IEEE Communications Surveys & Tutorials*, 21(4), 3639-3681.
5. Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A., & Payne, B. D. (2015). Evaluating computer intrusion detection systems: A survey of common practices. *ACM Computing Surveys* (CSUR), 48(1), 1-41.
6. Depren, Ö., Topallar, M., Anarim, E., & Ciliz, M. K. (2005). An intelligent intrusion detection system (IDS) for anomaly and misuse detection in computer networks. *Expert Systems with Applications*, *29*(4), 713–722. https://doi.org/10.1016/j.eswa.2005.05.002
7. Kim, D., Yang, J., & Sim, K. (2004). Adaptive intrusion detection algorithm based on learning algorithm. *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, 3, 2229-2233
8. Mittal, A., Gupta, A., & Agarwal, K. (2024, May). Anomaly Detection in Cybersecurity: Leveraging Machine Learning for Intrusion Detection. In 2024 International Conference on *Communication, Computer Sciences and Engineering* (*IC3SE*) (pp. 1-5). IEEE.
9. Mudigonda, N. (2022). A Method for Network Intrusion Detection Using Deep Learning, *Journal of Student Research*, 11(3).
10. Yaokumah, W., & Wiafe, I. (2020). Analysis of machine learning techniques for anomaly-based intrusion detection, *International Journal of Distributed Artificial Intelligence* (IJDAI), 12(1), 20-38.
11. Liu, Z., Su, N., Qin, Y., Lu, J., & Li, X. (2020). A deep random forest model on spark for network intrusion detection, *Mobile Information Systems*, 2020(1), 6633252.
12. Baykan, N. A., & Khorram, T. (2021). Network Intrusion Detection using Optimized Machine Learning Algorithms, *Avrupa Bilim ve Teknoloji Dergisi*, (25), 463-474.
13. Basholli, F., Mema, B., & Basholli, A. (2024). Training of information technology personnel through simulations for protection against cyber attacks. *Engineering Applications*, 3(1), 45-58.
14. İncekara, Çetin Önder . (2023). Industrial internet of things (IIoT) in energy sector. *Advanced Engineering Science*, 3, 21–30. Retrieved from https://publish.mersin.edu.tr/index.php/ades/article/view/839
15. Oliveira, N., Praça, I., Maia, E., & Sousa, O. (2021). Intelligent cyber attack detection and classification for network-based intrusion detection systems. *Applied Sciences*, *11*(4), 1674. https://doi.org/10.3390/app11041674
16. A. Airoboman, I. Araga, and J. Mohammad-Ashafa, "Reliability Improvement of Distribution System Network using Network Reconfiguration," *Engineering Applications*, vol. 3, no. 3, pp. 214–225, 2024. [Online]. Available: https://publish.mersin.edu.tr/index.php/enap/article/view/1581.
17. Nwafor, E. O., & Akintayo, F. O. (2024). Predicting trip purposes of households in Makurdi using machine learning: A comparative analysis of decision tree, CatBoost, and XGBoost algorithms. *Engineering Applications, 3(3),* 260–274. Retrieved from https://publish.mersin.edu.tr/index.php/enap/article/view/1605.
18. Mema, B., Basholli, F., & Hyka, D. (2024). Learning transformation and virtual interaction through ChatGPT in Albanian higher education. *Advanced Engineering Science, 4*, 130–140. Retrieved from https://publish.mersin.edu.tr/index.php/ades/article/view/1509.
19. Folino, G., Otranto Godano, C., & Pisani, F. S. (2023). An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity. *Journal of Supercomputing, 79*(9), 11660–11683. https://doi.org/10.1007/s11227-023-05049-x
20. Gonaygunta, H., Nadella, G. S., Pawar, P. P., & Kumar, D. (2024). Enhancing cybersecurity: The development of a flexible deep learning model for enhanced anomaly detection. *Systems and Information Engineering Design Symposium (SIEDS)*, Charlottesville, VA, USA, 79–84. https://doi.org/10.1109/SIEDS61124.2024.10534661
21. Handa, R., Kumar, S., & Kumar, S. (2019). Machine learning in cybersecurity: A review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(1), e1306. https://doi.org/10.1002/widm.1306
22. Bukhari, O., Agarwal, P., Koundal, D., & Zafar, S. (2023). Anomaly detection using ensemble techniques for boosting the security of intrusion detection system. *Procedia Computer Science*, *218*, 1003-1013. https://doi.org/10.1016/j.procs.2023.01.080
23. Habeeb, R. A. A., Nasaruddin, F., Gani, A., Hashem, I. A. T., Ahmed, E., & Imran, M. (2019). Real-time big data processing for anomaly detection: *A survey. International Journal of Information Management*, 45, 289-307.
24. Fernandes, G., Rodrigues, J. J., Carvalho, L. F., Al-Muhtadi, J. F., & Proença, M. L. (2019). A comprehensive survey on network anomaly detection. *Telecommunication Systems*, 70, 447-489.
25. Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A.A. (2019). A Detailed Analysis of the CICIDS2017 Data Set. In: Mori, P., Furnell, S., Camp, O. (eds) Information Systems Security and Privacy. ICISSP 2018. *Communications in Computer and Information Science*, vol 977. Springer

26. Liu, C., Gu, Z., & Wang, J. (2021). A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning. *IEEE Access*,*9*,74745–74756.
27. Amazon Web Services. (n.d.). *Amazon SageMaker developer guide: Random Cut Forest algorithm* (pp. 3567-3577). https://docs.aws.amazon.com/pdfs/sagemaker/latest/dg/sagemaker-dg.pdf#randomcutforest
28. Guha, S., Mishra, N., Roy, G., & Schrijvers, O. (2016, June). Robust random cut forest based anomaly detection on streams. In *International conference on machine learning* (pp. 2712-2721). PMLR.
29. Pang, Z., Cen, J., & Yi, M. (2023). Unsupervised concept drift detection method based on robust random cut forest. *International Journal of Machine Learning and Cybernetics*, 14(12), 4207-4222.
30. Yeom, S., & Jung, J. H. (2022). Weighted Isolation and Random Cut Forest Algorithms for Anomaly Detection. *arXiv preprint arXiv*:2202.01891.
31. Trawinski, I., Wimmer, H., & Kim, J. (2023). Anomaly detection in intrusion detection system using Amazon SageMaker. *2023 IEEE/ACIS 21st International Conference on Software Engineering Research, Management and Applications (SERA)*, 210–217. https://doi.org/10.1109/SERA57763.2023.10197735
32. Nigenda, D., Karnin, Z., Zafar, M. B., Ramesha, R., Tan, A., Donini, M., & Kenthapadi, K. (2022, August). Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In Proceedings of the *28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (pp. 3671-3681).
33. Jabbar, A. F., & Mohammed, I. J. (2020, November). Development of an optimized botnet detection framework based on filters of features and machine learning classifiers using CICIDS2017 dataset. In IOP Conference Series: Materials Science and Engineering (Vol. 928, No. 3, p. **032027)**. IOP Publishing.