Network Intrusion Detection

1st Amer Mohamed
Autonomous Systems A
Hochschule Hamm-Lippstadt
Lippstadt, Deutschland
mohamed-ahmed-mohamed-ali.amer@stud.hshl.de

***Abstract*—Since cybersecurity threats are increasing rapidly along with the rapid evolution of technology, network intrusion detection has become more important [1], [2]. In this study, we investigated the use of a Random Forest classifier for network intrusion detection, utilizing both the modified and original KDD Cup 1999 datasets [7], [13] to determine their practicality and efficiency to train a random forest classifier for this specific purpose. Our research produced a number of significant conclusions on the difficulties intrusion detection systems face. The initial dataset yielded models with a suspiciously high accuracy that would not convert to actual performance in real life scenarios. Further Research with the modified dataset produced serious over-fitting problems even after using extensive preparation methods, such as Principal Component Analysis and hyper-parameters tweaking on cleaner versions of the data. Our models performed poorly on validation data (about 79% accuracy), while they practically perfected training accuracy (around 99%).**

The metrics showed a trade-off in the behaviour of the system. the model performed very well at detecting actual network attacks, with a 97% recall rate against network attacks. on the other side, the system accurately identified only 67% of normal network connections as normal. The problem is that the model over-flagged lots of normal traffic as attacks, which would be a usability failure in real-world deployments where false positives could lead to low efficiency and a bad user experience. These findings point to the fundamental problem which is the need of high quality datasets that can be used to train machine learning models that it can identify accurately all types of attacks including novel ones.

I. Introduction

Networks safety is becoming a crucial topic in all types of enterprises and organizations. The importance of Network security and data confidentiality are growing rapidly due to the rapid evolution in technology and AI. One of the approaches taken to ensure network safety are network intrusion detection techniques [1]. Many techniques have been introduced through the evolution of technology. The goal of intrusion detection systems is to be able to detect unusual activities, unauthorized personnel, or misuse from insiders and external penetrators in a network, preferably in real-time. [2]. One of the emerging new technologies for network security is the use of machine learning and deep learning models to identify network attacks and intruders in a network. The abundance of big data has led to the ability to train machine learning and deep learning models efficiently. More research is being done on making those models extremely accurate and efficient for large deployment. [3]

II. Background

The main two types of Network detection system are Deployment method based and Detection method based. From the detection method perspective, it is further divided into two more categories "Signature-based intrusion detection (SIDS)" and "Anomaly detection-based intrusion detection (AIDS)".

The SIDS is based on the idea of defining a specific unique signature for network attacks. Those signatures are stored in a database. The system from there matches those signatures with the activity in the network and detects if there is a probable attack on the service. This type of approach lacks the ability to detect new types of attack as it lacks its signature and requires a huge carefully selected database which increases the computing resources needed for this algorithm [3].

The AIDS approach, also called the "behavior-based IDS," is based on the idea of defining a clear profile of normal users. Any deviation from this normal profile will be considered as an anomaly [4].The biggest advantages of using the AIDS approach are its ability to detect novel and new types of attacks. Though the only drawback of using this approach is the hard nature of classifying the difference between a normal and an abnormal profiles specially with the rising popularity of different IOT devices [5].In this paper, we aim to explore Random Forest machine learning approach along with data preprocessing techniques to improve the accuracy of network intrusion detection systems. The goal is to identify the most effective model parameters for detecting malicious activities within network traffic.

III. Dataset

The dataset used in this study is the KDD Cup 1999 Intrusion Detection Dataset, which was created by simulating attacks on a U.S. Air Force LAN to capture raw TCP/IP traffic. It contains 41 features (3 qualitative and 38 quantitative features) per connection and the target variable named class is labelled as normal and anomalous behaviour [6]. The features can be grouped by type into 4 categories.

A. Features

- **Basic Features of Individual TCP Connections:** These features are extracted from the basic TCP connection.

Many attacks can be determined just by analysing how the connection behaves [6], [7].

- **Content Features within a Connection:** These are features that inspect the payload or command content of a connection to detect any suspicious behaviour. They go beyond the basic properties of a connection and look deeper into what is actually being transmitted during the session. [6], [7]
- **Time based Traffic features:** These features consider connections to the same host in the past two seconds. [6], [7]
- **Host-based Traffic Features:** It is similar to Time based traffic features but these use a larger time window to detect patterns in connections. [6], [7]

B. Types of attacks

- **Denial of Service Attack (DOS):** It is a type of attack where the attacker overloads computing and memory resource. That makes the service too busy to fully handle legitimate requests and denies legitimate users access to the machine [7], [8]
- **User to Root Attack (U2R):** This type of attack occurs when the attacker starts out with access to a normal legitimate account on the system and then becoming able to exploit vulnerability to gain root access to the system [7], [8]
- **Remote to Local Attack (R2L):** It occurs when an attacker can send packets to a machine over a network where the attacker does not have access as a user to that machine [7], [8]
- **Probing Attack:** It is an attempt to gather information about a network of computers for the purpose of breaching through their security and gaining root access [7], [8]

C. Potential Issues in the Dataset

Previous research shows that this specific dataset has some issues. They experimented with various machine learning models all of which showed a very high accuracy of approximately 98% on the training dataset while having inconsistent accuracy fluctuations through epochs for the validation dataset. In other terms, it will not translate well into real life deployment. The first important deficiency in the KDD dataset is the huge number of redundant records. Analysing KDD train and test sets, it was found that about 78% and 75% of the records are duplicated in the train and test set respectively. This will cause the model to be biased towards the more frequent records and prevent it from learning the novel records. [8], [9]. The previous research made on the original KDD dataset introduced a newer version dataset called NSL-KDD. The researchers who made this addressed all the current issues observed in KDD dataset and made a new dataset that performed better with machine learning models in binary classification and multi-class classification models [9]. Table 1. shows the number of records for each attack category in the NSL-KDD training and validation datasets. It is also noticed that some types of attacks like U2R abd R2L has a really low count of records in the training data set and it is 4 times more tested in the testing dataset. This might lead to decreasing the model accuracy. This is implemented in the modified dataset to test if the model is actually able to generalize and detect attacks that the model have not seen quite often [9], [13].

TABLE I
NUMBER OF RECORDS IN KDDTRAIN+ AND KDDTEST+ DATASETS [9].

	Number of records					
	Normal	**DoS**	**Probe**	**U2R**	**R2L**	**TOTAL**
KDDTrain+	67 343	45 927	11 656	52	995	125 973
KDDTest+	9 711	7 458	2 421	200	2 654	22 544

D. Dataset Preprocessing

By analysing the features provided in the KDD dataset, two features had an always zero value which were "num_outbound_cmds, is_host_login" features. Those two features were dropped in the data preprocessing because they did not provide any importance in the model training. Moreover, three categorical columns were included in the dataset. After experimenting with encoding algorithms. "Hot-Encoding" the features proved the best metrics in model evaluation. When testing the model on the modified dataset, it showed extreme over-fitting of the model. Hence, More data preprocessing techniques were used like Principal component analysis (PCA) and data scaling.

IV. MACHINE LEARNING MODEL

A. Overview

For the network intrusion detection dataset, A random forest classifier algorithm was used. A random forest classifier is based on classification tree algorithm. A classification tree splits data recursively based on feature thresholds to create a tree structure that predicts class labels. Its aim is to divide the dataset into pure regions where almost all samples belong to a single class [10]. The algorithm works by partitioning the feature space into rectangles and then fitting a simple constant in each [11].

B. The mathematical model

The classification tree which is the parent method of Random forest classification tree works by having a root node that represents the first input and the entire data to be used, then it keeps on branching. each internal node represents decisions made depending on the input at an instance. Each leaf of the tree represents class labels or the final prediction . The splitting of nodes are usually based on a mathematical relation like "Recursive binary splitting" shown in equation (1) [9]–[11].

At each node of a classification tree, the dataset is split into two regions using a feature x_j and threshold s, as follows:

$$\begin{aligned} R_1(j,s) &= \{\mathbf{x} \mid x_j < s\}, \\ R_2(j,s) &= \{\mathbf{x} \mid x_j \geq s\} \end{aligned} \tag{1}$$

where:

- $\mathbf{x} \in \mathbb{R}^p$ is the feature vector
- x_j is its j-th component, and s is the splitting threshold.
- x_j is the j-th feature
- s is the threshold that defines the split.

In other words the algorithm tries to find the best pair of (j,s) that result in the purest split [10], [11].

The purity of the split is calculated by "Gini Index Equation" shown in equation (2) [11].

$$G(t) = 1 - \sum_{k=1}^{K} p_k^2 \tag{2}$$

where:

- K is the number of classes,
- p_k is the proportion of samples in node t that belong to class k.

The lower the value of $G(t)$ indicate purer nodes thus, better splits.

More algorithms for getting the purest split are sometimes used like "Cross-entropy" and "Misclassification error" [11].

Random forest is built upon the classification trees. It is a group of classification decision trees that merges to get more accurate values. In simpler terms, it is the same as using multiple decision trees machine learning models and getting the best output out of all of them [12]. In Fig. 1 is a small visualization of the random forest algorithm with two decision trees [12].

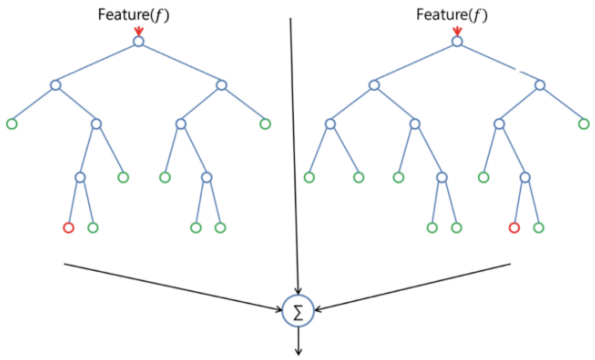

Fig. 1. Random Forest visualization with 2 decision trees

One of the downsides of the random forest algorithm is that it has more tendency to over-fit the data when the tree is deep. That is why it is important to utilize hyper parameters to ensure not having a too deep tree structure [12].

V. Implementation of Random Forest Classifier

A. Original Dataset

The supplied dataset was used to train and test the model due to the lack of labelling on the validation dataset. 80% of the data was used for the training and the rest for validation. the metrics captured showed a very high suspicious accuracy. Although both validation and training accuracy difference was too low, which shows that the model was working great. a very high accuracy of approximately 0.99 seems too high to generalize on real life. This shows that the dataset had issues as discussed in section 3. One of the causes of a very high accuracy is also having a very similar training and validation data that were captured in the same way that made the model even memorizing the noise of the data and performing well on both. The training and validation accuracy are shown in Fig. 2.

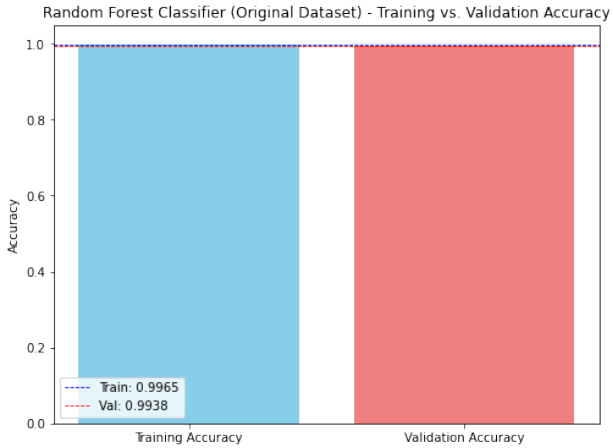

Fig. 2. Training and validation accuracy on KDD1999 dataset using random forest classifier machine learning algorithm.

B. Modified Dataset

Redoing the same process with the modified dataset showed different metrics. Due to the fact that the modified dataset had a labelled validation dataset. When splitting the training dataset and using 80% for the training and the rest for validation. The model did well in terms of accuracy, F1 score, recall and precision. The performance metrics of the model was really similar to the one used with the original dataset.Furthermore, when using the entire training dataset for training the model and taking the validation data provided to test the model. the model performed worse, where the training accuracy was close to 99% while the validation accuracy at 75%. To address this issue, an extensive search grid was used to tune the hyper parameters and more aggressive data pre-processing techniques were used on the training dataset such as Principal component analysis (PCA) reduction methods and Scaling of the dataset. This resulted in a slight increase in the accuracy of the validation to approximately 79.9%.

Further exploring the reason of this apparent over-fitting, the training dataset and the test dataset that includes 4 types of attacks as well as the normal traffic category mentioned earlier in Section 3. have a multi-class imbalance. Some attack types have a low count in the training dataset while having many occurrences in the test dataset. This was part of the refining

of the dataset to ensure that the model is able to generalize to unseen or barely seen types of attacks [9], [13].

The accuracy achieved might be okay for real-life implementation due to the fact that the model metrics for precision and recall are quite decent for a model for network intrusion detection. The goal is to have the model flagging all the real attacks barely missing any of them while also being able to flag most of the normal connections as normal.

The current state of the model trained percision-recall curve found in Fig. 3. shows that the model was able to detect 97%

Fig. 3. Percision-recall curve of the Random forest classifier on the modified NSL-KDD dataset

of the real attacks while the precision of 0.69 indicates that among all instances predicted as Class 1, only 69% are truly attacks, showing that the classifier prioritizes detecting attacks and sometimes flagging normal connections as attacks too.

VI. Conclusion

Network Intrusion have been always a big worry due to the fast paced technologies evolving over the years and the need to feel secure while using technology is crucial for humans. That is why developing a model for network intrusion detection that would perform perfectly is really needed. In the approaches used in this paper, It was deducted that more up to date problem-free datasets is still needed for a machine learning or deep learning model to perform better in real life scenarios of network attacks. More research and experimenting with more types of algorithms are still needed to provide the system with the needed safety and reliability.

References

[1] S. Kumar, Survey of current network intrusion detection techniques, Washington Univ. in St. Louis, pp. 1–18, 2007.

[2] B. Mukherjee, L. T. Heberlein and K. N. Levitt, "Network intrusion detection," in IEEE Network, vol. 8, no. 3, pp. 26-41, May-June 1994, doi: 10.1109/65.283931. keywords: Intrusion detection;Computer networks;Protection;Computer security;Data security;Computer science;Computer crime;Information security;Real time systems;Prototypes,

[3] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah, and F. Ahmad, "Network intrusion detection system: A systematic study of machine learning and deep learning approaches," Trans. Emerging Telecommun. Technol., vol. 32, no. 1, e4150, 2021. [Online]. Available: https://doi.org/10.1002/ett.4150

[4] W. Ma, "Analysis of anomaly detection method for Internet of Things based on deep learning," Trans. Emerg. Telecommun. Technol., vol. 31, no. 6, e3893, 2020. [Online]. Available: https://doi.org/10.1002/ett.3893

[5] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things-based smart cities: Recent advances and challenges," IEEE Commun. Mag., vol. 55, no. 9, pp. 16–24, Sep. 2017. [Online]. Available: https://doi.org/10.1109/MCOM.2017.1600514

[6] KDD Cup 1999, "KDD Cup 1999 Data," 1999. [Online]. Available: [Accessed: May 13, 2025].

[7] KDD Cup, "KDD Cup 1999 Data," [Online]. Available: http://kdd.ics.uci.edu/databases/kddcup99/task.html, [Accessed: May 13, 2025].

[8] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "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, doi: 10.1109/CISDA.2009.5356528. keywords: Testing;Intrusion detection;Data security;Statistical analysis;Computer security;Computer aided manufacturing;Learning systems;Computational intelligence;Computer networks;Application software,

[9] A. R. Tapsoba and T. Frédéric OUEDRAOGO, "Evaluation of supervised learning algorithms in binary and multi-class network anomalies detection," 2021 IEEE AFRICON, Arusha, Tanzania, United Republic of, 2021, pp. 1-6, doi: 10.1109/AFRICON51333.2021.9570886. keywords: Training;Supervised learning;Support vector machine classification;Predictive models;Prediction algorithms;Feature extraction;Classification algorithms;Intrusion Detection System (IDS);Supervised Learning Algorithms (SLA);Recursive Feature Elimination (RFE);AUC - ROC Curve;NSL-KDD,

[10] F. Pedregosa et al., "Tree-based models," Scikit-learn: Machine Learning in Python, [Online]. Available: https://scikit-learn.org/stable/modules/tree.html [Accessed: May 20, 2025].

[11] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York, NY, USA: Springer, 2009.

[12] M. S. Brown, "Random Forest Algorithm: A Complete Guide," Built In, 2023. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm

[13] T. H. Divyasree and K. K. Sherly, "A Network Intrusion Detection System Based On Ensemble CVM Using Efficient Feature Selection Approach," *Procedia Computer Science*, vol. 143, pp. 442-449, 2018, doi: 10.1016/j.procs.2018.10.416.