

University of Cincinnati

Data Driven Cybersecurity

Project: Big Data Analysis for Threat Detection

Harivardhan Manapati

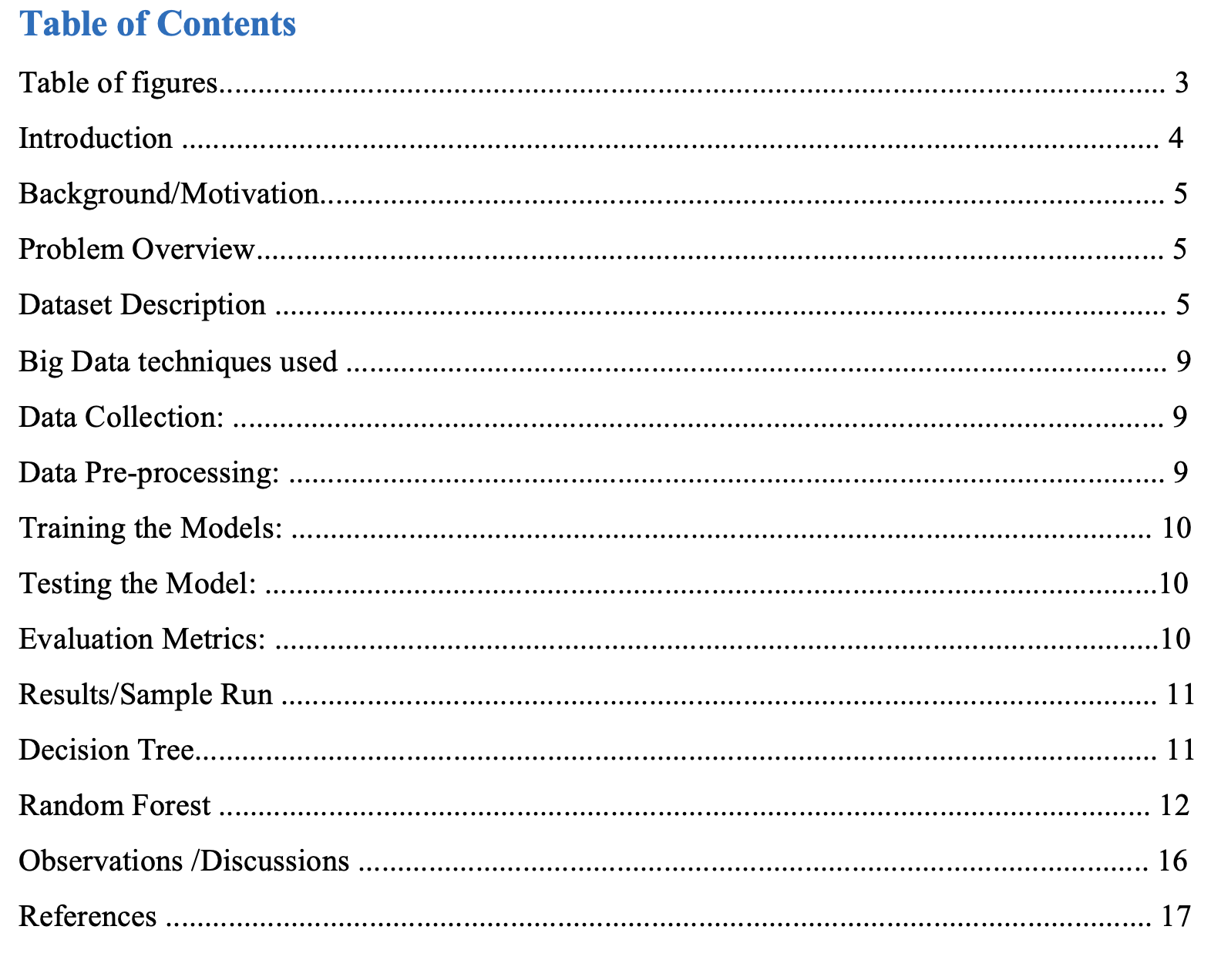
Amrutha Atluri

Sowmya Sree Pericherla

Ambika Nekkalapu

Vardhani Ramisetty

Harshitha Tallapally



**Table of Figures**

Figure:1 Number of instances for each attacks.................................................................... 6 Figure 2:Number of instances for Protocols..................................................................... 7 Figure 3: Attacking type columns after importing datasets.................................................... 11 Figure 4:Datasets after being imported.................................................................................... 11 Figure5:Counts of each attack…......................................................................................... 11 Figure 6:DecisionTreeEvaluationReports............................................................................... 12 Figure 7:Random Forest Evaluation Reports....................................................................... 13 Figure 8: Training accuracy for algorithms.............................................................................. 14 Figure 9:Testing accuracy for algorithms.......................................................................... 14 Figure10:Total time taken for Training algorithms................................................................. 14 Figure 11:Toatla time taken for testing algorithms. ............................................................... 16

**Introduction**  
Due to the ever-increasing amount, pace, and diversity of data created in the digital realm, the area of threat detection has seen substantial evolution in recent years. Advanced analytics approaches are urgently needed to efficiently identify and mitigate possible risks due to the sheer volume of data produced by multiple sources, including network logs, system logs, social media, and sensor data. Big data analytics become important in this situation.

In order to find patterns, trends, and insights that would be difficult or impossible to identify using conventional data processing techniques, big data analytics refers to the use of advanced analytics techniques on enormous datasets, like information mining, and artificial intelligence. collecting information using collection of data has become a potent method for identifying and thwarting insider threats, fraud, and other nefarious actions in the field of threat detection.

This research will examine how threat detection is changing as a result of big data analytics. We will explore the core ideas, techniques, and tools that make big data analytics a game-changer in the cybersecurity industry. We will also look at the benefits and challenges of using  stored information insights for threat detection and showcase real-world use examples where big data analytics has shown to be very successful in identifying and reducing a variety of risks.

With the use of prediction and classification algorithms like "Decision Tree, Random Forest, and Logistic Regression," we want to develop four machine learning models in this project that can identify risks. To categorize the data and identify risks when we train the data, our study has chosen a dataset that is a binary data set that comprises both threats and legitimate websites.

We evaluate the effectiveness of these algorithms using the criteria "Accuracy, Precision, Recall, and F-1 Score". Our study examines these measures among "Decision Tree, Random Forest, and Logistic Regression" to discover which approach delivers the most accurate and which algorithm provides inferior accuracy for the same training data.

**Background/Motivation**

Big Data Analysis for Threat Detection is an important project that has gained increasing attention due to the increasing threat of cyber-attacks and terrorism. With the proliferation of digital devices and the Internet, there has been a significant increase in the amount of data generated, making it difficult to analyse and identify potential threats in a timely manner.

The goal of this project is to develop a system that can effectively analyse and identify potential threats in large volumes of data, such as network traffic, social media feeds, and surveillance footage. This system will use advanced analytical techniques such as machine learning, data mining, and pattern recognition to identify anomalies and potential threats in the data.

The motivation for this project is to improve public safety and security by providing law enforcement agencies, security organizations, and government agencies with a powerful tool to detect and prevent potential threats. This system can also be used in commercial settings to detect fraud, identify potential security breaches, and improve overall security.

Overall, the Big Data Analysis for Threat Detection project has significant potential to improve public safety and security and is an essential research and development for both the public and private sectors.

**Problem Overview:**

The problem of threat detection is becoming increasingly challenging due to the ever-growing volume, velocity, and variety of data generated by organizations. Big data analytics for threat detection involves using advanced analytical techniques to detect potential security threats from massive volumes of structured and unstructured data.

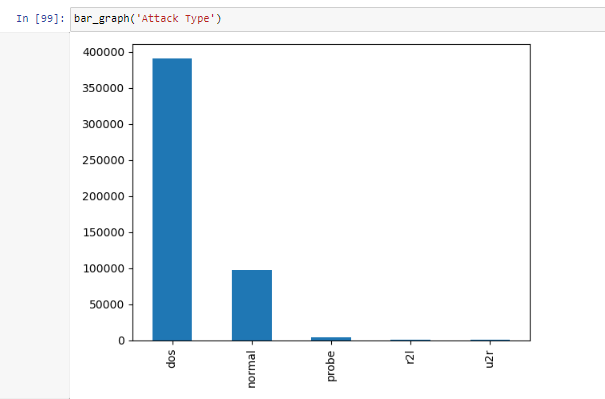
**Dataset Description:**

Around The uncompressed data utilized for training consisted of a maximum of   compressed distinct The Transmission Control Protocol (As a result, there were approximately a billion connected data. Comparatively, from over a period of time of test data, around billion connection records were produced.

When data generally transferred from one network handle to another, from a particular Internet Protocol  range to another using a predefined protocol, it does so via a connection, which is a series of transmission control protocol   information with fixed beginning and ending periods. Each link is categorized as either usual or a target, and there is only one form of approach. connection record typically takes roughly one hundred.

**Types of Attacks:**

The information presented above shows the many kinds of network assaults and how frequently they occur. The information is displayed in a tabular manner, with the “attacks” column listing the various assaults, such as “dos,” “normal,” “Probe,” “r2L,” and “u2R,” and the “dtype” column listing the frequency with which each attack was identified in the network traffic.

*Figure:1 number of instances for each attack*

The results show that “dos” attacks have been identified the most frequently, occurring 379,532 times per year. Attacks classified as “normal” were then identified 96,326 times after this. Only 1,034, 234, and 224 instances of each of the other assault kinds, such as “Probe”, “r2l”, and “u2r”, were found.

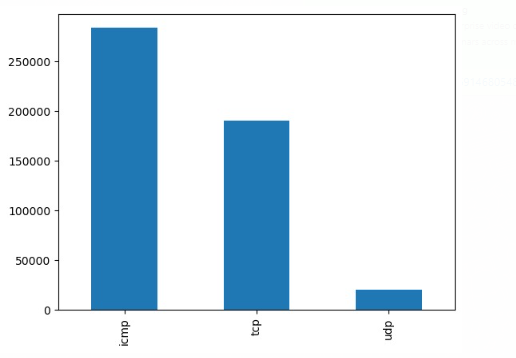
Graphical user interface, text

Description automatically generated

We can use the information in this data to determine the types of attacks that are frequently conducted against their networks and to take the necessary precautions to prevent them. They can also spot any irregularities or potential security issues by keeping track of the frequency of various attack types and acting quickly to minimize them. This information can aid in creating strong network security measures and ensuring that the network is safe and operating at its best.

**Protocol\_type**

The information below shows how frequently various network protocol types occur. The information is displayed as a graph, with the “protocol\_type” X-axis showing the different sorts of protocols, such as “ICMP,” “TCP,” and “udp,” and the “dtype” Y axis showing the frequency with which each protocol was found in the network traffic.

  
***Figure:2 number of instances of protocols***

The data show that “ICMP” is the protocol that is most frequently found, occurring 283,602 times per second. The “TCP” protocol was identified 190,065 times after this, but the “udp” protocol was only detected 20,354 times.

Network administrators and security analysts can learn more about the prevalent protocols used in their networks and spot any irregularities or potential security risks by using the information supplied in this report. They can make sure that their network is operating at its best and take the necessary action to solve any potential problems by keeping track of the frequency with which certain protocol types occur.

The test data comprises attack types that weren't present in the training data and does not come from the same distribution of chances as the data used to train, which is an important distinction to make. This makes the work more realistic. Some penetration professionals contend that since most novel assaults are modified copies of well-known ones, it is possible to identify new variants by looking for the a "mark" of well-known attacks assault, with another fourteen distinct kinds present solely in the validation data..

Table

Description automatically generated

**Big Data techniques used:**

As a team we decided to use ML algorithms and processes analysis of our data in terms of big data methods. The six stages which are typical in creating a machine learning model have been used.

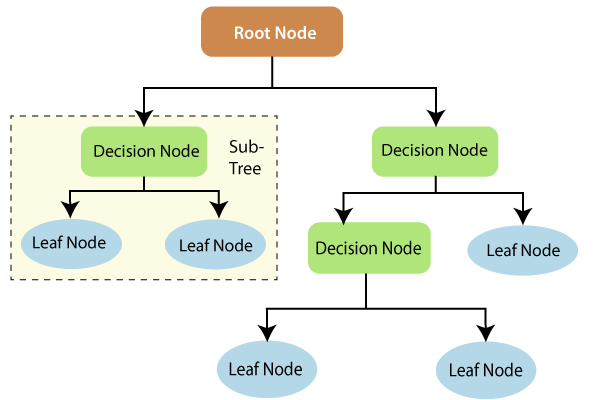
Gathering of Data We used the data sets from an open source Kaggle, which has real-time data regarding websites that are readily available, to train our model in the beginning and evaluate its performance because the goal of our study is to develop an intelligent system for detecting Threat that can result in Cyber-attacks.

Data cleaning is necessary after data collection in order to provide accurate findings and improve performance; otherwise, model training may result in mistakes. The choice of features is one of the key steps in this procedure. The Index and Age of Domain columns, for instance, may not be necessary when cleaning the data, so we may remove them and use the revised data set to extract more precise characteristics. Cleaning out NaN values and missing values is important part in the process of collecting the data. To obtain correct results, we may either replace them with the column's mean or with "0."

**Training Models:**

**Decision tree:**

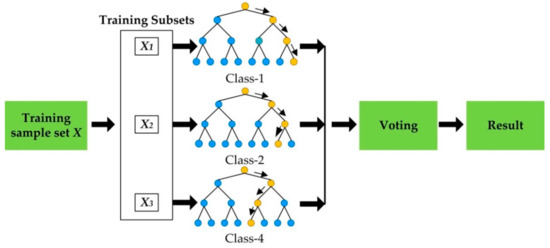
Decision trees, a popular machine learning approach, are included to solve, It is best option to beginners in the field of machine learning since they are simple to comprehend, interpret, and use. This thorough book will cover every facet of the decision tree algorithm, including its basic operating principles, various decision tree kinds, how to design a decision tree, and how to assess and improve a decision tree.



One of the algorithms for forecasting and categorization that is utilized the most is this one. It is organized in a tree-like fashion, with core nodes designating test attributes, branches designating test results or outputs, and leaf nodes designating class labels. Recursive partitioning is the process of repeatedly separating each derived sub-node in the source nodes to represent the attribute of a test in order to learn the tree. When all of the leaf nodes had the exact identical information as the target contingent, the repeated partitioning stops or it cannot be done because it is not adding any value. It is appropriate for exploratory investigations and can handle high-dimensional data. In general, decision trees offer decent accuracy.

**Random forest:**

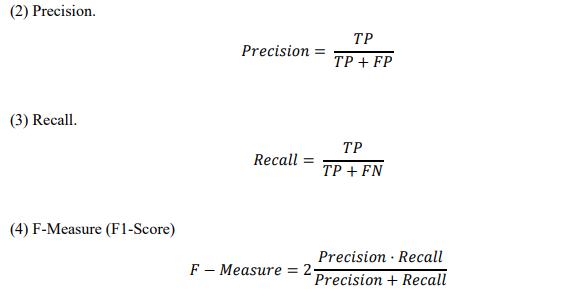
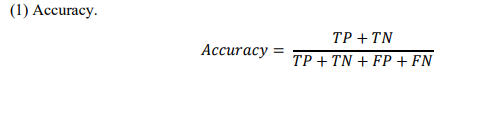
The supervised learning algorithm Random Forest uses the idea of bagging. In the technique of bagging a number of models are trained using various dataset subsets, and the final output is produced by combining the results of all the models. A decision tree serves as the foundational paradigm in the case of random forests.



The decision tree's output is based on a composite of numerous decision trees with low variance, not just one tree. The algorithm known as Random Forest uses several decision trees and a method known as Bagging, which combines the concepts of Aggregation and Bootstrap, to perform both Regression and Classification approaches. This concept revolves around utilizing the results of numerous decision trees rather than relying solely on one.

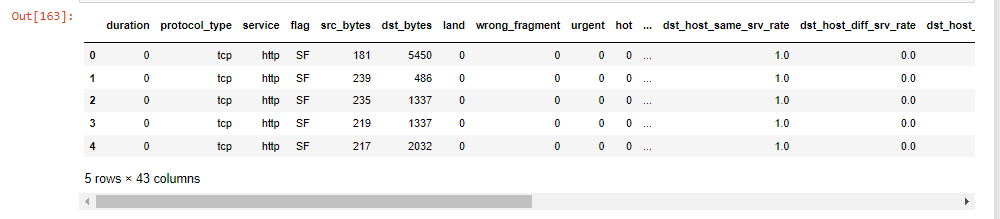
**Testing the Model**:   
We will test the model using 20% of the dataset after training it using 80% of the data. Where we may check these models' performance measures and determine if they can accurately anticipate the desired outcome.

**Evaluation Criteria:** The results of the information collection is primarily evaluated using metrics listed below.



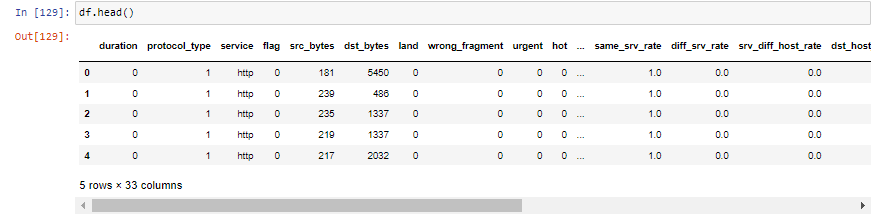
**Results**

Results shown following below,The Attacking type columns

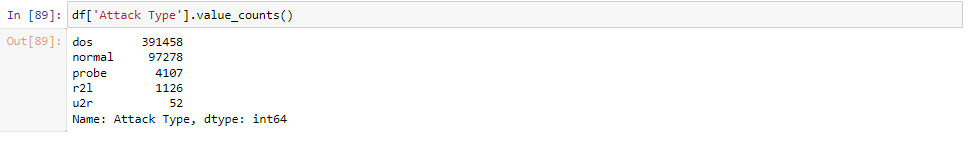


*Figure:3 Attacking type columns after importing datasets*

The next graphic, which shows the dataset after it has been imported, shows that all of the column identifiers are binary numbers:

  
*figure:4 data set after being imported*

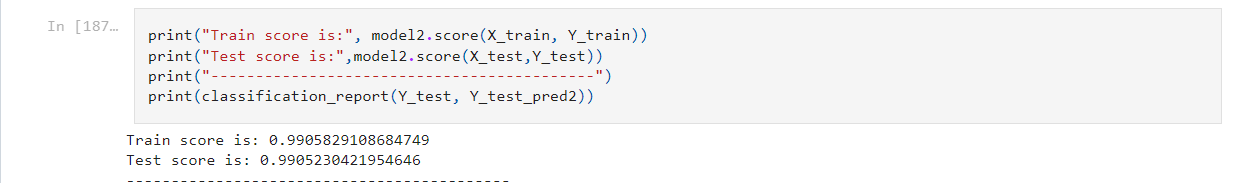
Based on the sorts of threats, the table shows how many there are in a system. A total of 494,021 threats were noted. A total of 391,458 of these threats were categorized as "dos," which make up the bulk. With 97,278 incidents, "normal" threats were the second most frequent category. Attacks known as "probes" were the third most frequent, with 4,107 cases. There were 1,126 "R2l" threats, or "remote to local," in all. A last point is that there were only 52 "u2r" threats, which stands for "user to root." The system was subjected to several attacks overall, the "dos" kind being the most prevalent.



*Figure:5 Counts of each attack*

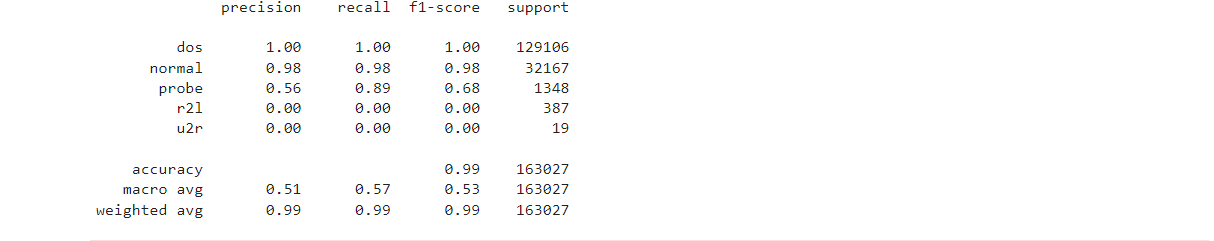
**Decision Tree**

The decision tree algorithm's outcomes are delivered in the form of several metrics. The model took 1.002018928527832 seconds to train, and it took 0.0874471664428711 seconds to test. According to the training score of 0.9905829108684749, the model has successfully mastered the training set of data. As indicated by the testing score of 0.9905230421954646, the model may generalize effectively to new, unforeseen data







  
*Figure 6:Decision Tree Evaluation Reports*

The model's effectiveness was evaluated in relation to each of the attack types, including "dos," "normal," "probe," "r2l," and "u2r," precision, and retention criteria. According to its perfect precision and recall scores, the model correctly detected every instance of the "dos" and "normal" attack types without producing any false positives or false negatives. However, the model's performance for the "probe," "r2l," and "u2r" attack types ranged from 0 to 0.56 in terms of precision and recall. The macro average F1-score of 0.53 for all attack kinds indicates a modest overall performance for the model.

In conclusion, it seems that the decision tree method does a decent job of accurately and quickly distinguishing between "dos" and "normal" attacks. The fact that it performs less well against various attack types shows that additional work has to be done to create a model that is more reliable and accurate.

**Random Forest**

A random forest model has been trained and evaluated using the provided data set on a dataset of network intrusion detection data. The model's training period lasted 6.85 seconds, while the testing period was 0.36 seconds. The model's An excellent precision score (“0.99999”) on the instruction set and a somewhat smaller precision score on the actual test set set, “0.9996”, show that it can generalize well to unseen data.





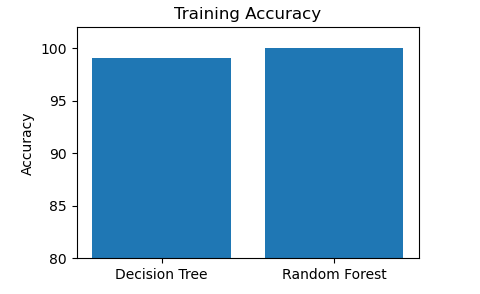
The model works well for all sorts of network attacks, including were also calculated reported. For DOS and regular traffic, the model specifically earned good precision and recall scores. It also achieved high precision scores for R2L and U2R attacks. Overall, it appears that the random forest model is a reliable and precise method for identifying network intrusions in real-world circumstances.



*Figure 7:Random Forest Evaluation Reports*

An essential indicator that shows how well a machine learning algorithm has learned from the data it was trained on is the training accuracy. We have two models in this situation, a decision tree and a random forest, each with a training accuracy score. The decision tree model successfully classified 99.05% of the training data it was exposed to, earning an exceptional training accuracy score of 99.5. The random forest model, on the other hand, was able to attain an even higher training accuracy score of 99.99%, suggesting a remarkable degree of classification accuracy.

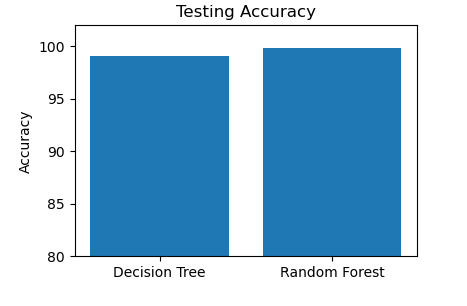
**Training Accuracy:**

  
*Figure:8 Training accuracy for algorithms*

The fact that both models achieved such high training accuracy ratings suggests that they were able to successfully acquire the training data that was provided to them. It's crucial to remember, though, that good performance on data that hasn't been seen before does not necessarily follow from high training accuracy. To assure these models' generalization performance, it is crucial to assess them using test data and other appropriate metrics. However, these training accuracy scores give a clear idea of how well these models are able to learn from their training data.

**Testing Accuracy:**

A crucial indicator for assessing how well machine learning models perform on untried data is testing accuracy. In this instance, we have two models—a decision tree and a random forest—each with a testing accuracy score that corresponds to it. A 99.05% testing accuracy score means that the decision tree model properly classified 99.05% of the unobserved data it was exposed to. The random forest model, on the other hand, was able to obtain a testing accuracy score of 99.86%, showing a higher degree of classification accuracy.

  
*Figure:9 Testing accuracy for algorithms*

However, these testing accuracy scores give a reasonable idea of how well these models perform in general on untested data. To verify these models' overall efficacy and robustness, it is crucial to assess their performance using a variety of indicators and test datasets.

**Total time for Training**

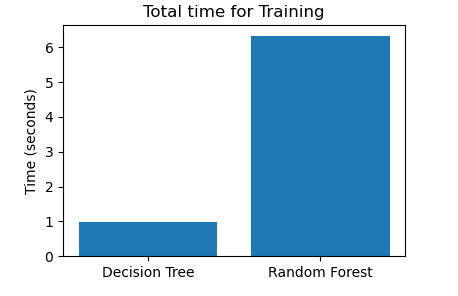
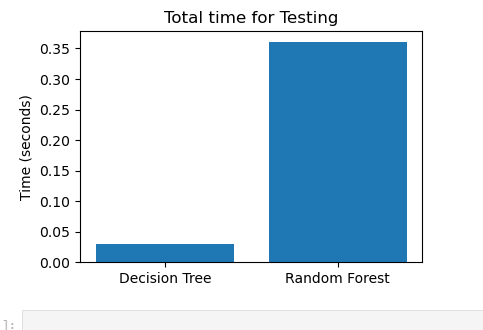
When working with huge datasets or complicated algorithms, the overall training time of a machine learning model is a significant consideration. We have two models in this situation, a decision tree and a random forest, each with a matching training period. The training of the decision tree model took a total of 99.98 seconds, but the training of the random forest model only required 6.31 seconds.

Figure: 10 Total times taken for Training algorithms

**Total time for Testing**

When working with huge datasets or real-time applications, the overall amount of time spent testing a machine learning model is a crucial issue to take into account. We have two models in this situation, a decision tree and a random forest, along with the corresponding testing times for each. The testing of the decision tree model took a total of 0.03 seconds, but the testing of the random forest model took a total of 0.36 seconds.

  
*Figure:11 Total time taken for testing algorithms*

The varying complexity of the algorithms and quantity of the testing datasets may be the cause of these variations in testing times. Different decision trees are used in random forests, which are more complex algorithms. The numerous decision trees that need to be reviewed when testing random forests may result in a longer processing time for the data. As a result, testing for the decision tree model may be finished significantly quicker than for the RF model.

**Observations /Discussions**

"Decision Tree, Random Forest" classifiers were used in this study to examine how the dataset was used. Results are shown in Table 2 along with "accuracy, weighted-precision, weighted-recall, and weighted-f1-score scores."

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **“Classifier”** | **“Weighted Accuracy”** | **“Weighted Precision”** | **“Weighted-Recall”** | **“Weighted-F1- Score”** |
| **Decision Tree** | “0.99” | “0.99” | “0.99” | “0.99” |
| **Random Forest** | “1” | “1” | “1” | “1” |

A classifier recognized 99% of the cases in the dataset correctly, according to its weighted accuracy score of 0.99. Similar to this, it obtained a weighted accuracy of 0.99, indicating that it correctly identified a certain class 99% of the time while making predictions about it. The weighted F1 score of 0.99 indicates that accuracy and recall are generally balanced, and the weighted recall of 0.99 demonstrates that the classifier correctly recognized 99% of cases belonging to a certain class.

Even better results were obtained by the Random Forest classifier, which received 1.00s for each of the four criteria. This indicates that the model had a flawless F1 score, excellent accuracy and recall, and correctly categorized every event in the dataset.

**As a result of our research, we have identified four angles for further work on threat detection:**

**Integration of Machine Learning and Human Knowledge**: Increasing the effectiveness of threat detection systems by combining the advantages of machine learning and human knowledge.

**Multi-Modal Data Fusion:** By combining data from several sources, such as video, audio, and sensor data, threat detection systems may be able to identify threats with more accuracy.

**Artificial Intelligence with an Explanation:** To help people comprehend how machine learning models make judgments, future research may concentrate on creating explainable artificial intelligence (XAI) tools.

**Defending Against Adversarial Attacks:** Future studies could concentrate on creating novel methods for identifying and thwarting aggressive assaults.

**References:**

**1. Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis https://bdataanalytics.biomedcentral.com/articles/10.1186/s41044-016-0006-0**

**2. Sharath Chandra Guntuku, Big data analysis frame work using random forest https://scholar.google.com/citations?user=76\_hrfUAAAAJ&hl=en&oi=sra**

**3. Big Data Analysis Techniques for Cyber-threat Detection in Critical Infrastructures https://ieeexplore.ieee.org/document/6844756**

**https://www.theguardian.com/technology/2016/oct/26/ddos-**

**attack-dyn-mirai-botnet**

**https://www.theguardian.com/technology/2016/oct/26/ddos-**

**attack-dyn-mirai-botnet**

**4. Use of machine learning in big data analytics for threat detection https://scholar.google.com/scholar?as\_q=Use+of+Machine+Learning+in+Big+Data+Analytics+for+Insider+Threat+Detection&as\_occt=title&hl=en&as\_sdt=0%2C31**

**5. Insider threat detection model for the cloud, Lucky Nkosi; Paul Tarwireyi; Matthew O Adigun**

**https://scholar.google.com/scholar?as\_q=Insider+Threat+Detection+Model+for+the+Cloud&as\_occt=title&hl=en&as\_sdt=0%2C31**

**6. Big data analytics for security and privacy challenges Aditya Dev Mishra; Youddha Beer Singh**

**https://scholar.google.com/scholar?as\_q=big+Data+Analytics+for+Security+and+Privacy+Challenges&as\_occt=title&hl=en&as\_sdt=0%2C31**

**7. Big data analytics with applications in threat detection https://www.researchgate.net/publication/327185823\_Big\_Data\_Analytics\_with\_Applications\_in\_Insider\_Threat\_Detection**

**8.Bhavani Thuraisingham, Mohammad Mehedy Masud, Pallabi Parveen, and Latifur Khan. 2017. Big Data Analytics for Malware Detection. Big Data Analytics with Applications in Insider Threat Detection (2017), 339-354. DOI :https://doi.org/10.1201/9781315119458-31**

**9.Bhavani Thuraisingham, Mohammad Mehedy Masud, Pallabi Parveen, and Latifur Khan. 2017. Insider Threat Detection for Sequence Data. Big Data Analytics with Applications in Insider Threat Detection (2017), 217-226. DOI:https://doi.org/10.1201/9781315119458-22**

**10.2020. Data Set Descriptions. Statistics Using Stata (2020), 673-685. DOI:https://doi.org/10.1017/97811087701630.021**

**11.2018. Classification: Theory - Decision Trees and Random Forest. (2018). DOI:https://doi.org/10.4135/9781526469168**

**12.Yassine Maleh, Mamoun Alazab, Loai Tawalbeh, and Imed Romdhani. 2023. Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence. CRC Press.**