Using Machine Learning Techniques to Detect Anomaly Progression in Common Network Traffic

Diana Terrazas-Lugo
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
terrazaslugod@arizona.edu

Cade Dacosta
Department of Management Information Systems
University of Arizona
Tucson, AZ, United States
cdacosta16@arizona.edu

Keran Jiang
Department of Management Information Systems &
Operations Supply Chain Management
University of Arizona
Tucson, AZ, United States
kj369@arizona.edu

Weiming Chen
Department of Management Information Systems &
Computer Science
University of Arizona
Tucson, AZ, United States
weimingchen@arizona.edu

***Abstract*— This research aims to investigate the accuracy of various machine learning algorithms in predicting cyber attacks using older datasets such as KDD-99 and NSL-KDD and newer datasets such as UNSW NB15. Four popular algorithms, including Naive Bayes, K-Nearest Neighbors, Random Forest, and Linear SVM, are compared to determine the significance of the features used in the prediction process and to detect the various attack types and how they change over time. The results of the study show that Random Forest achieves the highest accuracy among the tested algorithms, indicating that it is a suitable algorithm for predicting cyber attacks. Additionally, the study identifies the critical features that contribute to accurate predictions, which can be useful in developing effective cybersecurity systems. The importance of selecting appropriate features for successful prediction is emphasized throughout the study. This finding indicates that careful consideration should be given to the selection of features when developing cybersecurity systems. In summary, this research provides valuable insights into the application of machine learning algorithms for predicting cyber attacks. It highlights the importance of selecting appropriate features for successful prediction and emphasizes the significance of using accurate and relevant datasets in cybersecurity research. The results of this study can be used to guide the development of more effective cybersecurity systems and can serve as a foundation for future research in the field of cybersecurity.**

Keywords—Machine Learning Algorithms, Naive Bayes, Random Forest, Support Vector Machine, K-Nearest Neighbors, KDD-99, NSL-KDD, UNSW NB15, cybersecurity, cyberattacks

I. Introduction

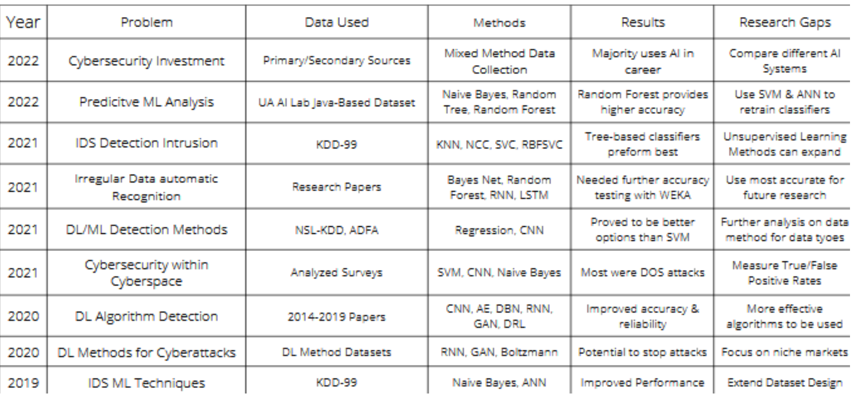
As the world becomes increasingly interconnected and data-driven, the need to handle cybersecurity within cyberspace and its data has become more critical than ever before. The different cybersecurity methods play a vital role in safeguarding sensitive data from various cyber threats, including hacking, identity theft, and data breaches [1]. In older and more recent years, several studies have investigated different cybersecurity methods to protect the thousands of datasets. This paper aims to provide a comparative analysis of these different datasets that have changed over time, drawing from a range of published research articles. By analyzing the findings of these studies, we aim to evaluate the strengths and weaknesses of each method and provide insights into their potential applicability and effectiveness in real-world scenarios. Essentially, using the Intrusion Detection Systems (IDS) within networks to help supervise suspicious activity to ensure full protection of data. By looking through each of the different methods that companies, organizations, and other entities decide to take, they will provide a better grasp of understanding their strengths, weaknesses, gaps, and other helpful insights. There are many different approaches to detecting breaches of data [1]. Using Machine Learning Algorithms to analyze the different datasets, it will give us further insight on which algorithm works best for the cyberattack datasets. By further expanding on these similar results and similar feature engineering, We can get a full performance view on each of the datasets with the use of Performance Matrix, Confusion Matrix, and the ROC Curve. It is important for people to understand which method is more accurate and effective for their dataset to fully extract the information within their data. Measuring the different cyberattacks that aggregate over time, we are able to see how these various attacks modify themselves and evolve to continue hacking into systems within cyberspace. Collecting these datasets over time will allow us to catch any patterns and possibly predict future cyber attacks that may potentially occur, helping us prepare ourselves to protect our systems. Through this analysis, we hope to contribute to the ongoing discourse on cybersecurity and provide guidance for individuals and organizations seeking to protect their data from cyber threats.

.

II. Literature Review

There is a difference in the overall landscape of potential attacks as the Web has evolved over time. Findings provide valuable insights into the application of machine learning algorithms to predicting various types of cyber attacks and how attack types are starting to vary over time based on datasets. Measuring different datasets that start from 1999 and end in 2015 will help offer us an insight to how cyberattacks started and proceeded to evolve over time with the everchanging IoT. Allowing us to gather information on previous, current, and potential cyberattacks overall and find the best ML algorithm to detect the anomalies.

A. Taxonomy



Year	Problem	Data Used	Methods	Results	Research Gaps
2022	Cybersecurity Investment	Primary/Secondary Sources	Mixed Method Data Collection	Majority uses AI in career	Compare different AI Systems
2022	Predicitve ML Analysis	UA AI Lab Java-Based Dataset	Naive Bayes, Random Tree, Random Forest	Random Forest provides higher accuracy	Use SVM & ANN to retrain classifiers
2021	IDS Detection Intrusion	KDD-99	KNN, NCC, SVC, RBFSVC	Tree-based classifiers preform best	Unsupervised Learning Methods can expand
2021	Irregular Data automatic Recognition	Research Papers	Bayes Net, Random Forest, RNN, LSTM	Needed further accuracy testing with WEKA	Use most accurate for future research
2021	DL/ML Detection Methods	NSL-KDD, ADFA	Regression, CNN	Proved to be better options than SVM	Further analysis on data method for data tyoes
2021	Cybersecurity within Cyberspace	Analyzed Surveys	SVM, CNN, Naive Bayes	Most were DOS attacks	Measure True/False Positive Rates
2020	DL Algorithm Detection	2014-2019 Papers	CNN, AE, DBN, RNN, GAN, DRL	Improved accuracy & reliability	More effective algorithms to be used
2020	DL Methods for Cyberattacks	DL Method Datasets	RNN, GAN, Boltzmann	Potential to stop attacks	Focus on niche markets
2019	IDS ML Techniques	KDD-99	Naive Bayes, ANN	Improved Performance	Extend Dataset Design

Figure 1 - Taxonomy Table

Our team got inspiration for using the KDD-99 and NSL-KDD datasets through previous articles that we read. These articles studied the different datasets and measured their various attack types using Machine Learning Algorithms. Figure 1 is a small snapshot of some of the literature we read to expand our own knowledge on. A lot of the results within the literature stated that the Random Forest classification proved to have more accuracy and overall better performance when compared to the other Machine Learning algorithms. We were curious if this would remain consistent with our own findings.

B. Research Gaps & Questions

Previously, literature has compared KDD-99 and NSL-KDD with different machine learning algorithms. These include the use of Naive Bayes, Support Vector Machine, and other Supervised Learning algorithms [3]. However, some gaps our group has recognized include the use of comparing to more recent datasets like UNSW NB15, and focusing on any new varieties of cyberattacks. The Naive Bayes, Support Vector Machine, Random Forest, and K-Nearest Neighbors methods help us expand more and calculate the results for all of the machine learning algorithms that will allow us to compare the results. By comparing, we can measure the consistency within our results with others and focus on any gaps that might have occurred with previous literature. We are able to expand further with similar results and similar feature engineering to expand the results using Performance Matrix, Confusion Matrix, and ROC Curve to have the full performance view of the datasets. Other research used ML algorithms to classify the various types of cyberattacks in common network traffic. We expanded further by identifying hyperparameters that point to attack types and then retrained and returned the strongest classifier which ended up being Random Forest. Some questions we want to answer are:

1. How do attack types differ within older datasets like KDD-99 and NSL-KDD compared to a more recent dataset like UNSW NB15?
2. What Machine Learning Algorithm offers the best performance when testing the datasets?

III. Research Testbed

A. KDD-99

We've decided to use previous literature that tested this dataset and compare results. This dataset contains 4.8 million instances, with 42 attributes, and holds categorical and integer characteristics within said attributes. One factor that we will be looking through the dataset is the variety of attacks they show the user. There are five different classes being Normal, Denial of Service (DoS), User to Root (U2R), Remote to Local (R2L), and Probe (Probing Attack). To ensure full understanding, I've included a small definition for each of the attacks:

Normal: No intrusion detected.

Denial of Service: Depleting the resources of the victims causing them to be unable to handle requests, i.e. flooding.

Remote to Local: Unauthorized access from a remote machine. An attacker intrudes into a remote machine and gains local access to said machine, i.e. password guessing.

User to Root: An attacker uses a normal account to login into a victim system to gain administrator privileges by exploiting some vulnerability in the victim, i.e. overflow attacks.

Probing: Surveillance to gain information about the victim, i.e. port scanning [5].

These attack types of what we focus on to compare the results of the literature to ensure consistency with our own testing.

B. NSL-KDD

A revised version of KDD-99 that came out in 2009, this dataset solves the issues where only selected records are chosen. Created to reduce redundancy of duplicate records, adding new attack types, and less computationally expensive to train [3]. This dataset is beneficial to use for further expansion of testing the data. Containing 1.1 million instances and 43 attributes, it has less data and makes it easier to test. Using this dataset can help compare the results from testing both of them to compare accuracy of the percentage of types of attacks. Also giving us insight of how cyberattacks looked during these times.

C. *UNSW NB15*

This dataset from 2015 published by the University of New South Wales contains 9 different types of attacks and 49 attributes. The increase in types of cyberattacks further proves our hypothesis of the expansion of cyberattacks over time. UNSW NB15 contains both binary and multi-class attack scenarios while also providing a realistic & challenging set of data to test and compare different approaches to detecting & mitigating network attacks [2]. Expanding on the 9 types of attacks, they are:

Fuzzers: The attacker attempts to discover gaps within the security of a network by feeding it with the massive inputting of random data to make it crash.

Analysis: Variety intrusions that try to enter web applications through port scans, emails, and web scripts.

Backdoors: Technique of sneaking through stealthy normal authentication, to secure unauthorized remote access to a device, and locating the entry to plain text while struggling to continue unobserved.

DoS: Intrusion which disrupts the computer resources to be extremely busy in order to prevent the authorized requests from accessing a device.

Exploits: A sequence of instructions that takes advantage of a vulnerability that is caused by an unintentional or unsuspected behavior on a host or network.

Generic: Technique that establishes against every block cipher using a hash function collision without respect to the configuration of the block cipher.

Reconnaissance: Similar to a probe attack, gathers information about a network to evade its security controls.

Shellcode: Attack in which the attacker penetrates a slight piece of code starting from a shell to control the compromised machine.

Worm: Attack whereby the attacker replicates itself in order to spread on other computers. Often, it uses a computer network to spread itself, depending on the security failures on the target computer to access it [4].

Labeling this dataset, two attributes were provided: attack_catrepresents the nine categories of the attack and the normal within a binary classification being either 0 for normal and 1 otherwise.

IV. Scikit Learning Methods

A. *Feature Engineering*

Feature engineering is an important step of developing any effective and strong machine learning model. It mostly consists of identifying and tweaking features that are relevant towards predicting outcomes of interest.In this study, we propose a feature engineering process first starting with the KDD-99 and NSl-KDD datasets. Specifically, we're going to emphasize the discrete and continuous features that will be used to train our upcoming various machine learning models.

First, let's take a look at the discrete features we used for our study that are found in the KDD-99 and NSl-KDD datasets. These are some of the features that can be mapped to values:

protocol_type: Feature shows the type of protocol used in the connection, ex. TCP, UDP, or ICMP. It has been mapped to numerical values, {TCP: 0, UDP:1, and ICMP:2, and others.}

service: Feature shows the type of service running on the destination host, ex. HTTP, SSH, or telnet. It has been mapped to numerical values, {HTTP:0, SSH:1, and telnet: 2, and others.}

flag: Feature shows whether there is a normal or error status of the connection.It has been mapped to numerical values, {Normal:0, Error:1.}

Next, let's move on to our continuous features. These are features that take on numerical values, and include the following:

duration: Feature shows the length of the connection in the number of seconds.

srv_diff_host_rate: This feature indicates the percentage of various connections to different hosts that has the same service.

dst_bytes: This feature indicates the number or count of data bytes flowing from the destination to the source.

These are just a few of the features included in both the KDD-99 and NSL-KDD datasets. By understanding and analyzing these various continuous and discrete features, we can gain further insights into the characteristics of network traffic and identify potential attack types.

Jumping forward in time in terms of our dataset collection we saw that our UNSW dataset mostly differs in features from the KDD-99 and NSL-KDD datasets but kept some minor features to the latter. For this study, we utilize a set of these discrete and continuous features from the UNSW-NB15 dataset:

proto: Feature shows the type of protocol used in the process, ex. TCP, UDP, or ICMP. It has been mapped to numerical values, {TCP: 0, UDP:1, and ICMP:2, and others.}

service: Feature shows the type of service running on the destination host, ex. HTTP, SSH, SMTP, DNS, etc.. It has been mapped to numerical values, {HTTP:0, SSH:1, and SMTP: 2, DNS: 3, and others.}

state: Feature shows and represents the state of the connection with flags.It has been mapped to numerical values, {Normal:0, Error:1, etc..}

Next, let's move on to our continuous features found in the UNSW-NB15 dataset. These are features that take on numerical values, and include the following:

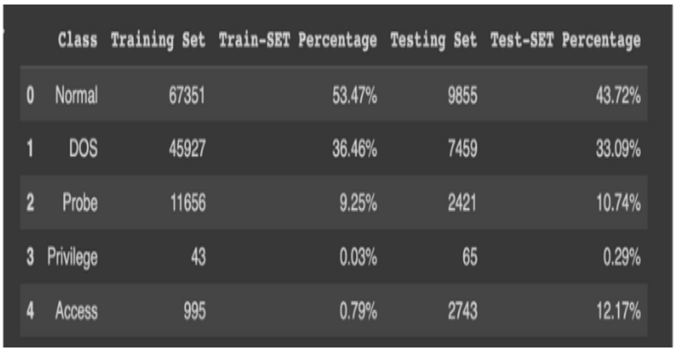
dur: Feature shows the length of the connection in the number of seconds.

dloss: This feature represents the number of data packets that were dropped or had to be retransmitted in communications.

dbytes: This feature indicates the number destination to source bytes send in the communications process [5].

B. Multi-Class Classification

To be able to identify how attack methods have changed over time we needed to identify various attack types through a multi-class classification approach. As we saw in the last section of this paper, because of similar feature engineering, the KDD-99 and NSl-KDD datasets have the same attack labels. The KDD99 and NSL-KDD also had only five attack categories. Figure 2 represents how a Multi-Class classification approach is applied to both the training and testing models derived from the KDD-99 and NSL-KDD datasets.

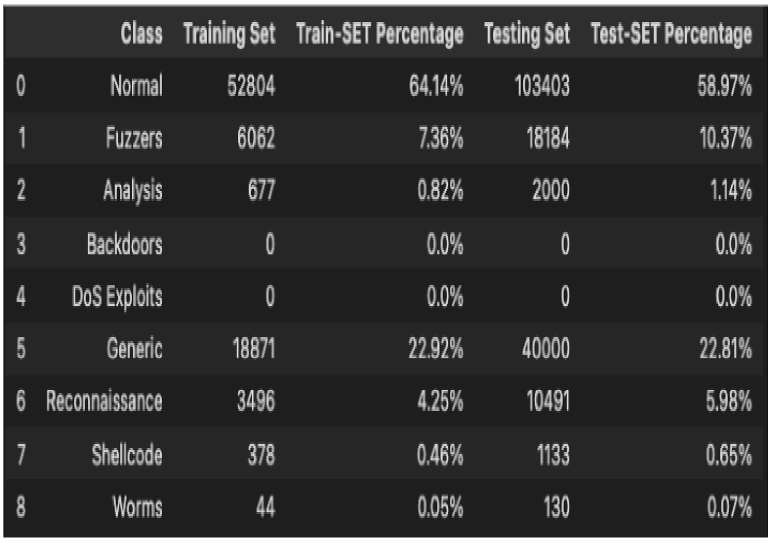


	Class	Training Set	Train-SET Percentage	Testing Set	Test-SET Percentage
0	Normal	67351	53.47%	9855	43.72%
1	DOS	45927	36.46%	7459	33.09%
2	Probe	11656	9.25%	2421	10.74%
3	Privilege	43	0.03%	65	0.29%
4	Access	995	0.79%	2743	12.17%

Figure 2 - Multi-Class Classification distribution ex. (KDD-99 & NSL -KDD)

With only 5 attack labels, this type of multi-class classification wouldn't be considered the strongest towards labeleding predictive classes for most of the modern cyber attacks we see in today's cyber threat landscape. The KDD-99 and NSL-KDD datasets were among the pioneer datasets to be used for anomly detection system creation, and at the time only five attack categories were suitable for a strong system. These datasets served as a good starting point for researchers and helped to lay the foundation for modern intrusion detection systems.

However, with the increasing complexity and diversity of modern cyber-attacks, it became clear that a more diverse and comprehensive dataset was necessary to keep up with all the new attack types being faciliated in the threat landscape.. The UNSW-NB15 dataset was introduced to address this need and included nine attack categories. The additional 4 attack categories in the UNSW-NB15 dataset help provide a more realistic representation or updated version of the current threat landscape, allowing for the development of more sophisticated and up-to-datemachine-learning models that can better detect and classify attacks. Figure 3 represents how a Multi-Class classification approach is applied to both the training and testing models derived from the UNSW-NB-15 dataset for this study.



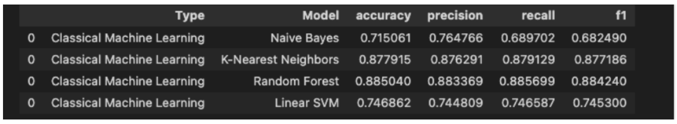
	Class	Training Set	Train-SET Percentage	Testing Set	Test-SET Percentage
0	Normal	52804	64.14%	103403	58.97%
1	Fuzzers	6062	7.36%	18184	10.37%
2	Analysis	677	0.82%	2000	1.14%
3	Backdoors	0	0.0%	0	0.0%
4	DoS Exploits	0	0.0%	0	0.0%
5	Generic	18871	22.92%	40000	22.81%
6	Reconnaissance	3496	4.25%	10491	5.98%
7	Shellcode	378	0.46%	1133	0.65%
8	Worms	44	0.05%	130	0.07%

Figure 3 - Multi-Class Classification distribution ex. (UNSW-NB-15)

The increased diversity of the UNSW-NB15 dataset has several benefits. First, it provides a more comprehensive view of the modern types of attacks that modern networks face in the current threat landscape. Second, it allows us to train and evaluate machine-learning models that are more effective at detecting and classifying modern types of attacks. This, in turn, can lead to the development of more robust and effective anomaly detection systems that are better equipped to combat modern cyber threats compared to our prior datasets.

C. Classical Machine Learning Algorithm Results

Now that we have various attack classes labeled, classical machine learning models can be formulated to predict an attack instance. Figure 4 represents the reults from running a classcial machine learning approach to our most recent dataset, the UNSW-NB15 dataset.



	Type	Model	accuracy	precision	recall	f1
0	Classical Machine Learning	Naive Bayes	0.715061	0.764766	0.689702	0.682490
0	Classical Machine Learning	K-Nearest Neighbors	0.877915	0.876291	0.879129	0.877186
0	Classical Machine Learning	Random Forest	0.885040	0.883369	0.885699	0.884240
0	Classical Machine Learning	Linear SVM	0.746862	0.744809	0.746587	0.745300

Figure 4 - Classical ML matrix of results from the UNSW-NB-15 Dataset

In terms of performance for the UNSW-NB-15 dataset, the method results vary from one another. However, in general, we see that Random Forest is the best performing algorithm with this dataset, followed by KNN, Linear SVM, and Naive Bayes. Random Forest is the best compared to the rest while Naive Bayes turns out to be the worst performing. Potentially due to the fact that it assumes that all values are independent of each other and do not affect one another. Our values could very well depend on one another causing the Naive Bayes method to prove itself not useful within the scenario. Meanwhile, Random Forest is able to detect outliers and combines multiple decision trees to make the final prediction, which forms a stronger result. For this dataset, Random Forest performs better due to having these characteristics. Our literature review was able to build classical Ml learning algorithms similar to ours through slightly different feature engineering than ours, but we wanted to be able to go beyond it to formulate a stronger model that also takes feature weights into consideration. The following sections go over this criteria.

V.

VI. Methodology

The use of machine learning algorithms is becoming increasingly important in the field of cybersecurity, as they can help detect and prevent cyber attacks in real-time. The algorithms have the ability to analyze large amounts of data and identify patterns that may indicate potential threats. In this research, several machine learning algorithms were tested to determine their effectiveness in detecting anomalies in network traffic data, which can be a crucial factor in enhancing cybersecurity. The aim was to determine which algorithm could offer the most accurate and efficient detection of anomalies, ultimately providing a more secure environment for online activities.

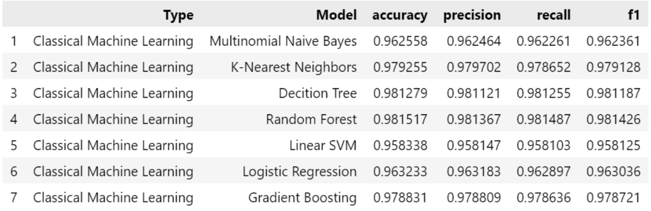
- Gradient Boosting: This algorithm is an ensemble method that combines multiple decision trees to improve the overall performance of the model. In this research, Gradient Boosting was used to classify the network traffic data into different categories based on their attributes.
- K-Nearest Neighbor: This algorithm is a non-parametric classification technique that uses a distance metric to classify data points. In this research, K-Nearest Neighbor was used to identify anomalous network traffic based on their similarity to other data points.
- Support Vector Machine: This algorithm is a binary classifier that uses a hyperplane to separate data points into two classes. In this research, Support Vector Machine was used to identify anomalous network traffic by classifying data points as either normal or anomalous.
- Random Forest: This algorithm is an ensemble method that combines multiple decision trees to improve the overall performance of the model. In this research, Random Forest was used to classify the network traffic data into different categories based on their attributes.
- Decision Tree: This algorithm is a tree-based classification technique that recursively splits the data into different categories based on their attributes. In this research, Decision Tree was used to classify the network traffic data into different categories based on their attributes.
- Naive Bayes: This algorithm is a probabilistic classification technique that uses Bayes' theorem to calculate the probability of a data point belonging to a particular class. In this research, Naive Bayes was used to classify the network traffic data into different categories based on their attributes.

In the comparative analysis of various machine learning algorithms, we observed that the Random Forest and Decision Tree algorithms exhibited the best performance for our problem. The reasons for their superior performance are as follows:

- Handling Mixed Data Types: Both the Random Forest and Decision Tree algorithms can efficiently handle datasets with mixed data types, such as numerical and categorical features. Our dataset contains a mixture of these data types, making these algorithms well-suited for the task[11].
- Robustness to Overfitting: The Decision Tree algorithm, although prone to overfitting, performs well on our dataset. The Random Forest algorithm, on the other hand, is inherently robust to overfitting due to its ensemble approach. By constructing multiple decision trees and aggregating their predictions, the Random Forest algorithm reduces the impact of any single overfit tree, leading to improved overall performance.
- Feature Importance: Both Random Forest and Decision Tree algorithms can provide insights into the importance of individual features, which can be helpful in understanding the key factors that contribute to accurate predictions. This information can be valuable in developing more effective cybersecurity systems and strategies.
- Scalability: Random Forest, in particular, is highly scalable and can easily handle large datasets, making

it suitable for real-time cybersecurity applications. Additionally, the algorithm's parallel processing capabilities enable it to efficiently process large amounts of data, enhancing its practical applicability in cybersecurity settings.

The superior performance of the Random Forest algorithm in our analysis underscores its potential as a powerful tool for predicting cyber attacks. However, it is important to note that the performance of machine learning algorithms may vary depending on the dataset and problem context. Therefore, further research on more recent datasets and other emerging machine learning techniques is essential to stay abreast of the latest developments in the field and to continuously enhance cybersecurity approaches.



	Type	Model	accuracy	precision	recall	f1
1	Classical Machine Learning	Multinomial Naive Bayes	0.962558	0.962464	0.962261	0.962361
2	Classical Machine Learning	K-Nearest Neighbors	0.979255	0.979702	0.978652	0.979128
3	Classical Machine Learning	Decition Tree	0.981279	0.981121	0.981255	0.981187
4	Classical Machine Learning	Random Forest	0.981517	0.981367	0.981487	0.981426
5	Classical Machine Learning	Linear SVM	0.958338	0.958147	0.958103	0.958125
6	Classical Machine Learning	Logistic Regression	0.963233	0.963183	0.962897	0.963036
7	Classical Machine Learning	Gradient Boosting	0.978831	0.978809	0.978636	0.978721

Figure 5 - Model Analysis

Also, the results obtained by running various machine learning models and evaluating their performance using accuracy, precision, recall, and F1 scores provided valuable insights into the effectiveness of these algorithms in detecting anomalies in network traffic data [Figure 5] . High scores across all metrics for the best-performing models demonstrate their potential for accurately and consistently identifying cyber threats. Accuracy: the accuracy score measures the proportion of correctly classified instances out of the total instances in the dataset. The random forest model's high accuracy of 0.9815 indicates that it correctly identifies normal and anomalous network traffic with a very high success rate. Although the decision tree and gradient boosting models also achieve high accuracy around 0.98, the random forest model slightly outperforms them. Precision: which is the ratio of true positive instances to the sum of true positive and false positive instances. Our top model has a high accuracy score of close to 0.98, which means that when the model predicts instances of abnormal network traffic, it is correct most of the time. This demonstrates the effectiveness of these models in reducing false positives, which is critical in cybersecurity applications where too many false positives can lead to alert fatigue and reduced responsiveness to real threats. Recall: high recall of a model means it can successfully identify most instances of actual anomalous network traffic. This is an important aspect of cybersecurity, as failure to detect a real threat can have significant consequences for affected systems and organizations. F1: the F1-score is the harmonic mean of precision and recall, providing a balanced assessment of model performance. Although some models, such as decision trees and gradient boosting, had performance scores close to the random forest model, the latter was still the best model in our analysis. The superior performance of random forest models can take advantage of the characteristics discussed above.

Overall, the use of multiple machine learning algorithms allowed us to compare their effectiveness in detecting anomalies in network traffic data. We found that each algorithm had its own strengths and weaknesses, and that combining methods can offer more thorough defense against cyber threats. Specifically, we found that Random Forest had the strongest performance in our tests, but further research with more recent datasets may reveal different outcomes. Our findings suggest that ongoing research and development in this area is necessary in order to stay ahead of emerging cyber threats and to strengthen cybersecurity approaches.

VII. Results & Discussions

Our results were similar to previous literature, realizing that Random Forest proves to have the best performance and accuracy when compared to other Machine Learning Algorithms. The result shows the number of estimators has an impact on accuracy. Below is the average accuracy where Random Forest refers to the number of Decision Trees within the forest. However, the relationship between the number of estimators and the accuracy is not always linear and can vary depending on the specific problem and dataset. Average accuracy, on the other hand, is the average performance of the model over multiple iterations or folds of cross-validation. By repeatedly splitting the data into training and validation sets, cross-validation can provide a more reliable estimate of the model's performance. The relationship between average accuracy and the number of estimators in a random forest model can be complex. Initially, increasing the number of estimators can lead to a rapid increase in the accuracy of the model as the ensemble of trees is able to capture more complex patterns in the data.

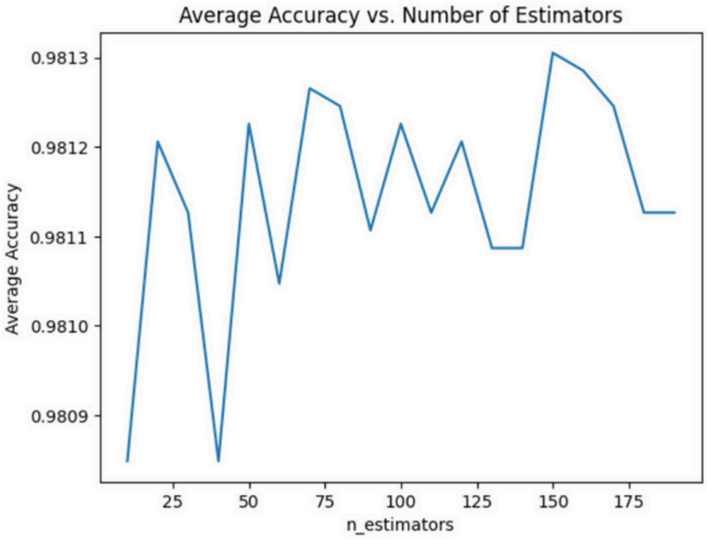


Figure 6 - Average Accuracy vs Number of Estimators

To determine the optimal number of estimators for a random forest model, it is often necessary to perform a grid search or other hyperparameter tuning techniques. By testing different values for the number of estimators and comparing the resulting average accuracy, it is possible to find the optimal balance between model complexity and generalization performance. We found that the random forest performs best for our task when the estimator is close to 150 in Figure 5. Then, we did randomized search cross validation, which samples a fixed number of parameter settings from the specified distributions to evaluate a model with a specified set of hyperparameters to find the best ones.

Hyperparameters in a random forest algorithm are parameters that are set before the model is trained and cannot be learned from the data. These parameters control the behavior of the algorithm and can have a significant impact on the performance of the model. Determining how many Decision Trees will be used within the Random Forest ensemble while also controlling the maximum depth of each Decision Tree. According to the use of grid search cross validation, which is a technique for repeatedly evaluating various combinations of hyperparameters to find the ideal set of values for the best performance. Upon evaluating the model for each combination of hyperparameters, we identified the optimal parameter values that enhance the accuracy of the Random Forest model for our task. The results reveal that the model achieves its best performance when the number of trees (n_estimators) is set to 152, which aligns closely with our initial prediction. This fine-tuning of hyperparameters further improves the model's ability to detect anomalies in network traffic data and solidifies Random Forest as a robust choice for this particular cybersecurity application. We also use k-fold cross validation to assess the performance of the random forest we have now. The highest accuracy of using k-fold validation is 98.3%. Now, we have the best model with suitable parameters and important features to train, it is time to show the training results and explain why choosing random forest. Overall, hyperparameters for a Random Forest model can be used to fine- tune the balance between model complexity and generalization performance[12].

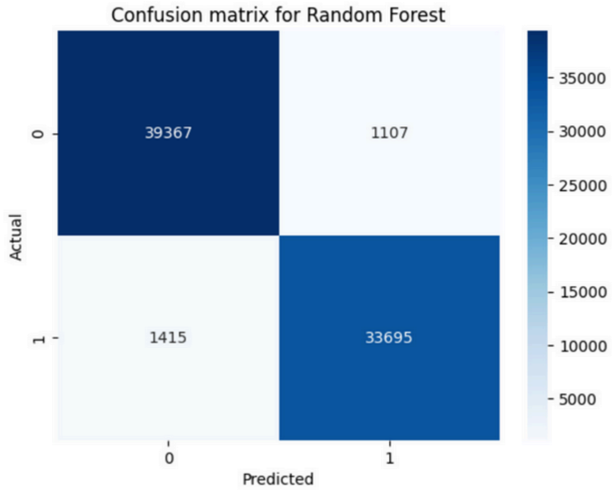


Figure 8 - Confusion Matrix for Random Forest

The Confusion Matrix allows us to summarize the performance of a classification model to help better determine the accuracy of our results at a different angle of performance. A confusion matrix for a random forest model can be used to evaluate the performance of the model and to calculate various performance metrics such as accuracy, precision, recall, and F1 score. These metrics can be used to assess the model's ability to correctly classify instances and to identify areas for improvement. First, according to the Confusion Matrix, the random forest was able to identify around 39,000 true positives, more than false values. Referring to the number of instances that were correctly classified as positive, while false positives refer to the number of instances that were incorrectly classified as positive. This means that the model is doing a good job of correctly classifying positive instances, and there are relatively few instances that are being incorrectly classified as positive. This could be an indication that the model is effectively capturing the underlying patterns and relationships in the data. Looking at the values for False Negatives and False Positives, having True Positives and True Negatives is a good sign that shows our model is determining the best results. Showing that our dataset is not resulting in overfitting the data suggests that it is correctly classifying the dataset. Following with the Confusion Matrix, we decided to continue on with evaluating the performance of our datasets by using the technique below.

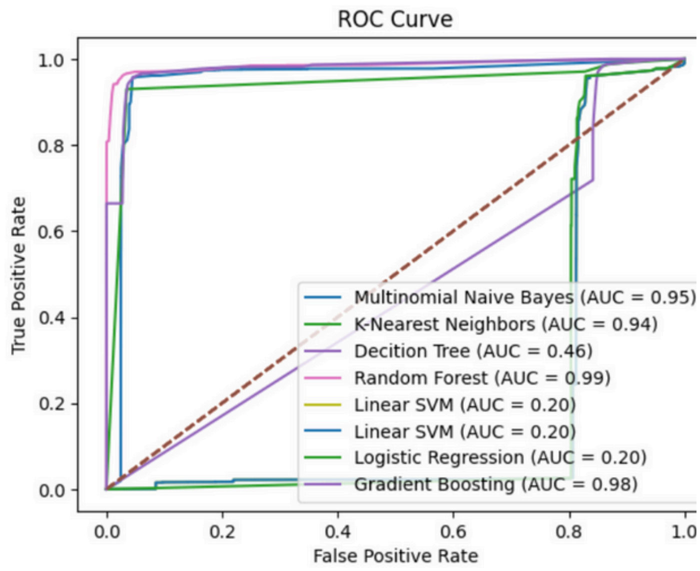


Figure 9 - Models ROC Curve

Figure 8 illustrates the Receiver Operating Characteristics (ROC) Curve, a graphical representation of the performance of binary classification models. The curve plots the true positive rate (TPR) against the false positive rate (FPR) for different classification thresholds. In our study, we visualized the ROC curve for each model to summarize the trade-offs between TPR and FPR. In the context of a Random Forest, the ROC curve can be generated by varying the probability threshold used to classify an instance as positive or negative. Ideally, we want a model that achieves a high TPR while maintaining a low FPR, meaning it should correctly identify most positive instances while minimizing the number of false positives. Among the tested models, the Random Forest classifier's curve is closer to the top-left corner, indicating its superior performance. It is worth noting that other models also demonstrated good performance according to the ROC curve, with four lines exhibiting a similar pattern to that of the Random Forest model, tending towards the upper left corner. However, the Random Forest model still emerges as the best choice for our task. The AUC (Area under curve) represents the probability that a randomly chosen positive instance will have a higher predicted probability than a randomly chosen negative instance. In the context of a random forest, a higher AUC indicates that the model is better at distinguishing between positive and negative instances. The Random Forest model, with its chosen parameters, emerges as the most suitable classifier for our task, boasting an AUC close to 1.0, even in the presence of other models with commendable performance.

VIII. Conclusions & Future Directions

Therefore, the study has shown that machine learning algorithms, specifically Naive Bayes, K-Nearest Neighbors, Random Forest, and Linear SVM, can effectively predict cyber attacks using the KDD-99 and NSL-KDD datasets. Random Forest was found to have the highest accuracy among the tested algorithms, and the study also identified the critical features that contribute to accurate predictions. However, the study also highlights the importance of using more recent and relevant datasets, such as UNSW B15 and CICIDS 2017, to improve the accuracy and reliability of our results. The study has provided valuable insights into the application of machine learning algorithms for predicting cyber attacks and emphasizes the importance of selecting appropriate features for successful prediction. The findings of this research can be beneficial for various entities, including business companies, academic researchers, and individuals who use online platforms. For business and insurance companies, this study can help improve their data protection and cybersecurity measures by utilizing machine learning algorithms to predict and prevent cyber attacks. For academic researchers, this study provides a basis for further research in the field of cybersecurity and machine learning algorithms. For individuals who use online platforms, this study raises awareness of the potential risks associated with these activities and encourages them to take necessary actions to protect themselves. A promising future direction for research in this field would be to focus on using more recent and relevant datasets to further improve the accuracy and effectiveness of machine learning algorithms in predicting cyber attacks. Specifically, using network traffic data from 2023 and beyond would be beneficial in adapting machine learning algorithms to current and emerging cyber threats. Additionally, research could explore the integration of different machine learning algorithms to develop more comprehensive and effective cybersecurity systems. The findings of this research can be useful for various entities in improving their data protection and cybersecurity measures in the future.

Acknowledgment *(Heading 5)*

We would like to thank all those who have supported and contributed to the completion of this study. We thank Dr. Hsinchun Chen (University of Arizona) for providing valuable insights and comments on the manuscript. We thank Steven Ullman (University of Arizona) for assistance in constructing controllable and constructive feedback. We thank Benjamin Ampel (University of Arizona) for providing their experiences and knowledge in data processing with Python.

References

[1] G. Jakka, N. Yathiraju, and M.F. Ansari, “Artificial intelligence in terms of spotting malware and delivering cyber risk management,” Journal of Positive School Psychology 2022, Vol. 6, No. 3, pp.6156-6165, 2022, [online]. Available: https://journalppw.com/index.php/jpsp/article/view/3522/2300 [Accessed April 08, 2023]

[2] A. Barkah, S. Selamat, Z. Abidin, and R. Wahyudi, "Impact of data balancing and feature selection on machine learning-based network intrusion detection", International Journal of Informatics Visualizations, (JOIV), March 2023, pp 241-248

[3] S. Rastogi1, .t Shrotriya, M. Kumar Singh, P. Raghu Vams, “An analysis of intrusion detection classification using supervised machine learning algorithms on nsl-kdd dataset”, Journal of Computing Research and

Innovation (JCRINN), 2022, Vol 7, No. 1, pp124-137, March 2022, [online]. Available: https://ir.uitm.edu.my/id/eprint/60675/1/60675.pdf [Accessed April 13, 2023].

[4] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: statistical analysis of the unsw nb-15 data set and the comparison with the kdd99 data set", INFORMATION SECURITY JOURNAL: A GLOBAL PERSPECTIVE, 2016, vol 25, nos 1–3, 18–31 [online].

[5] R. Rama Devi, and M. Abualkibash, "Intrusion detection system classification using different machine learning algorithms on kdd-99 and nsl-kdd datasets - a review paper", International Journal of Computer Science & Information Technology (IJCSIT), Vol 11, No. 3, June 2019

[6] S. Rastogi1, .t Shrotriya, M. Kumar Singh, and P. Raghu Vams, "An analysis of intrusion detection classification using supervised machine learning algorithms on nsl-kdd dataset", Journal of Computing Research and Innovation (JCRINN), 2022, Vol 7, No. 1, pp124-137, March 2022, [online].

[7] A. Barkah, S. Selamat, Z. Abidin, and R. Wahyudi, "Impact of data balancing and feature selection on machine learning-based network intrusion detection", International Journal of Informatics Visualizations, (JOIV), March 2023, pp 241-248

[8] "Kddcup99", Datahub.com 2019, [online]. Available: Kddcup99 - Dataset - DataHub - Frictionless Data

[9] M. Zaib, "Nsl-kdd", Kaggle.com, May 2022, [online]. Available: NSL-KDD | Kaggle

[10] N. Moustafa, "The unsw-nb15 dataset", The University of South Wales, 2015, [online]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[11] R. Primartha and B. A. Tama, "Anomaly detection using random forest: A performance revisited," 2017 International Conference on Data and Software Engineering (ICoDSE), Palembang, Indonesia, 2017, pp. 1-6, doi: 10.1109/ICODSE.2017.8285847.

[12] Biau, G., & Scornet, E, "A random forest guided tour," 2016, TEST, 25, 197-227. https://doi.org/10.1007/s11749-016-0481-7