Cyber Attack Detection in Cloud Forensics

Yasaman Emami
Department of Applied Data Science
San José State University
San José, United States
yasaman.emami@sjsu.edu

Shamama Afnan
Department of Applied Data Science
San José State University
San José, United States
shamama.afnan@sjsu.edu

Purnima Bhukya
Department of Applied Data Science
San José State University
San José, United States
purnima.bhukya@sjsu.edu

Poojitha Katta
Department of Applied Data Science
San José State University
San José, United States
poojitha.katta@sjsu.edu

Deepali Zutshi
Department of Applied Data Science
San José State University
San José, United States
deepali.zutshi@sjsu.edu

***Abstract*—Cloud Computing is ubiquitous now-a-days because of its scalability, accessibility, and cost-effectiveness. Cybersecurity is one of the biggest challenges in cloud computing as the customer resources are compromised during the attack. The attackers often use Virtual Machines to attack anonymously. In this project, we are proposing the application of various supervised and unsupervised machine learning algorithms for detecting attacks based on the collected evidence. The Evidence Detection in Cloud forensics dataset will be used from IEEE Data Port. The target variables are labeled as the binary state of attack or normal to corresponding virtual machines. We will provide a comparison analysis of the performance of the machine learning models used in this study.**

***Index Terms*—cloud computing, cybersecurity, machine learning, supervised learning, unsupervised learning**

I. Introduction

Cloud computing involves the use of the internet to provide computer resources to customers such as data analytics, data storage and management, and increased computing power. Cloud based environments provide users with various services such as software, infrastructure, and platform [1]. These services build on top of each other to allow ease of use to customers and help achieve their business goals.With data being produced at unprecedented rates, more companies and organizations are moving to the cloud. Although the cloud offers many benefits such as speed, scalability, performance, and reduced costs, it is also vulnerable to security threats and malicious attacks.This has resulted in an increased need for security against attacks designed to circumvent the system and gain access to the network and the data. The battle between security experts and malware developers is an always going activity. Malware behavior evolves at the same rate as security innovation.

II. Significance to Real World

Modern research focuses on evolution of malware patterns. To detect and identify such infections, Machine Learning methods are applied. Security experts and specialists must continually upgrade their cyber defenses to stay a step ahead of malware. An Intrusion Detection System (IDS) is a software that monitors traffic and detects attacks within the network and when combined with machine learning, such a model would not only be able to detect attacks but also predict and prevent them. The main aim of this project is to create a machine learning based IDS that can successfully analyze network traffic features and classify whether they are normal or malicious in nature. Attacks on such systems are evolving and becoming increasingly effective, therefore it is extremely important to come up with a system that can prevent them.

III. Literature Review

There has been some research on the relevance and application of digital forensics in cloud computing. A review of the current methodologies gives important insights into the problem and probable solutions. [2] Employs machine learning to detect traffic in the network that may be malicious. The ISOT-CID dataset is used to create an Intrusion Detection System and contains information such as network traffic memory, properties of system call, and resource utilization. A variety of models were created and evaluated including an Artificial Neural Network (ANN), different values for K-fold cross validation were tested and the highest accuracy was 94 % when K=10. The Decision Tree, Random Forest, and K-Nearest Neighbors models performed surprisingly well with an accuracy of 100 % for all values of K (5,10,15). A Support Vector Machine was also implemented and performed moderately well with accuracies of 81% and 84% for K=10 and 15 respectively. The model with the poorest performance was the Naive Bayes classifier with an accuracy of only 60% for all values of K. The paper concludes that Decision Tree

and Random Forest are the most optimal models for intrusion detection with their performance being validated by cross validation and split validation. [3] aims at detecting and classifying anomalies encountered in cloud-based environments which are very susceptible to malicious attacks. Feature selection is performed to use only the most important attributes which boosts the performance of the models exponentially. The original dataset contains 49 features and best-features selection technique is applied to come up with the 20 most useful features. The Logistic Regression (LR) and Random Forest (RF) algorithms were implemented and their performance was compared. A comparison of the models reveals that RF outperforms LR with an overall error of about 0.9% at 0.4% UND and an FAR of 1.9% as compared to LR which has an error of 4% with a UND of 3.5%. The categorization of the type of attack is done in two ways- the first is simply categorization of each of the single attacks and the second one involves a multi-stage categorization process. The single type attack categorization results in an accuracy of 80% while the step-wise categorization performs with an accuracy of 93.35%. In [4], the authors applied Naive Bayes, K-means and Decision Tree models to detect DDoS attacks. Their goal was to achieve a fast detection rate, low computational cost and high accuracy. As it was not possible to create DDoS attacks in the public cloud such as AWS, Azure, or Google Cloud Platform, they simulated using virtual environments such as VMs. They achieved 98.8% accuracy applying the C4.5 Decision Tree Model, 91.4% for Naive Bayes and 95.9% for K-means in 0.58s, 0.58s and 1.12s detection time respectively. In [5], the authors introduced an SVM based framework for detecting attacks in virtual machines under changing conditions. Their proposed model enables the quantification of the effect resource adjustments. In the first approach, they trained and tested the model using the non-linear mapping of original data to higher dimensions. They also considered the resource adjustments by generating many sub-features for each single feature. In the second approach, they applied SVM without changing the dataset. Their experiments showed that their model performed better than traditional SVM and Decision-Tree. The accuracy is increased by 1.79% under the effect of resource adjustments. In [6], the authors mentioned a technique to Detect DDoS attacks using Apache Spark framework in 2020. They mentioned a way to detect attacks in an open stack based private cloud. The paper claims to have observed more accuracy using the Random Forest method amongst Decision Tree and Logistic Regression. By setting up Spark as a service and using streaming on a packet sniffer to capture real-time data. MapReduce algorithms are used to process the data. Features such as number of connections, number of data bytes between source and destination, metrics on connections to the same service were considered to build the model. Considering False-positives, Accuracy and Precision as the evaluation metrics, Random Forest showed a high of 94.40% with an accuracy of 89.87% on real-time data. The same has been employed on the KDD Cup dataset as well to justify that Random Forest was a better performer among the three approaches. [7] compared various algorithms which included Decision Tree, Naive Bayes, SVM and KNN for data captured in a virtualized cloud environment.The data of normal traffic along with log files was captured on the virtualized cloud environment. The analysis was performed based on the efficiency,performance,and f-score,accuracy. It was concluded that Decision tree is the most efficient algorithm when compared to others for analysis over virtualized cloud data. [8] used WEKA machine learning tool for detecting and defining DDoS attacks. They used both explorer and knowledge flow environment of WEKA was used for evaluation.THe dataset consisted of 27 features and 5 different classes.They compared four different classifiers which included J48,MLP,Random Forest and Naive Bayes to classify the attacks.The results of the comparison concluded that J48 classifier performed the best and Naive Bayers had less accuracy. In [9], the authors conducted multiple tests on two distinct datasets: the NSL-KDD dataset and the CIDDS-001 dataset. The findings are then evaluated and compared using parameters such as overall accuracy, precision, and recall. They also proposed the I-SiamIDS technique, which separates hostile network traffic from secure network traffic. They are filtering the benign traffic many times to decrease the possibility of misclassification. K-Means, IF, Random Forest, CNN, KNN, DNN, and XGBoost are among the algorithms that have been evaluated. The accuracy of both datasets was assessed and ranged from 42.46 percent for IF to 97.26 percent for DNN on the CIDDS-001 dataset. On the NSL-KDD dataset K-Means has the lowest accuracy of 72.8 percent, while XGBoost has the highest accuracy of 80.1 percent. Because False Negative (FN), which shows misclassification of hostile traffic, is more important, they choose the first layer of the proposed technique with a mixture of algorithms that produce fewer FN. Although KNN produces decent results due to being a lazy learner and would be too slow for real-time classification with large input numbers, they chose an ensemble of Siamese-NN, b-XGBoost, and DNN for the first layer. For the second layer's aim of categorizing attacks, XGBoost's multiclass classification was trained, allowing for response selection to the threat. In [10], the authors of this study used multiple algorithms and compared the outcomes. SVM, Naive Bayes (NB), Decision Tree (DT), and Mini Batch K-Means are among these approaches (MBK). In this comparison research, F-measure Precision and Recall are considered for assessment on two separate datasets. For the first dataset, F-measure ranges from 75% for NB and MBK approaches to 91% for SVM. Precision varies from 73 percent for MBK to 98 percent for SVM, while Recall ranges from 62 percent for NB to 85 percent for SVM. For the second dataset, trials reveal that the maximum F-score is 99 percent for DT and the lowest is 72.5 percent for NB. Precision DT is 100 percent and NB is 75 percent, while Recall of DT is 98 percent and NB is 70 percent. Based on the accuracy it provided for the test sets and the new input malicious data detection power, the authors proposed DT as the final model. After new input being classified as malicious, the severity is determined in order to compute a risk score for each IP

address.

IV. Methodology

The methodology can be divided into three main parts- Exploratory Data Ananlysis, Data Preprocessing, and Model Implementation. Each step contains further steps that form an essential part of understanding the data before modeling may be performed.

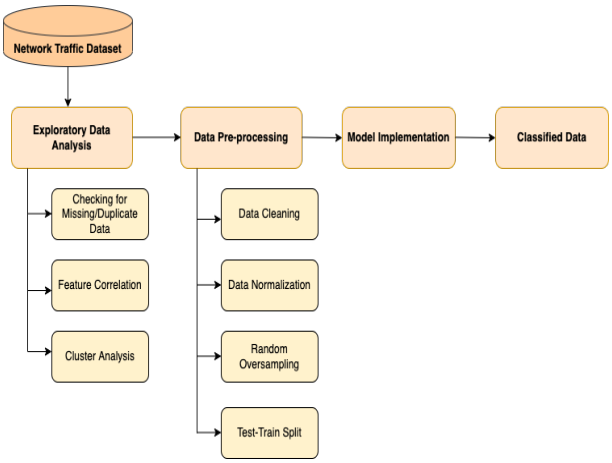

Fig. 1. Implementation Process

A. Exploraroty Data Analysis (EDA)

a) Target Variables: The dataset is not balanced. In the dataset, 24% status are marked as Attack and 76% status are Normal. [Fig. 2]

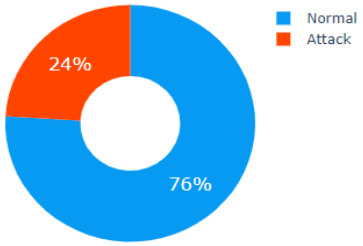

Fig. 2. The percentage of status

b) Feature Correlation: From the feature correlation heatmap [Fig. 3], we can observe that the status is highly correlated to the rate of transmitted packets from the network. The status is positively correlated to the rate of transmitted bytes from the network, the rate of received packets from the network, the rate of received bytes from the network, the rate of time spent in kernel space, and the rate of time spent in userspace. The status is negatively correlated with the rate of the number of reading bytes on the vda block device, the rate of the number of write requests on the vda block device, and the rate of time spent by vCPU threads executing guest code.

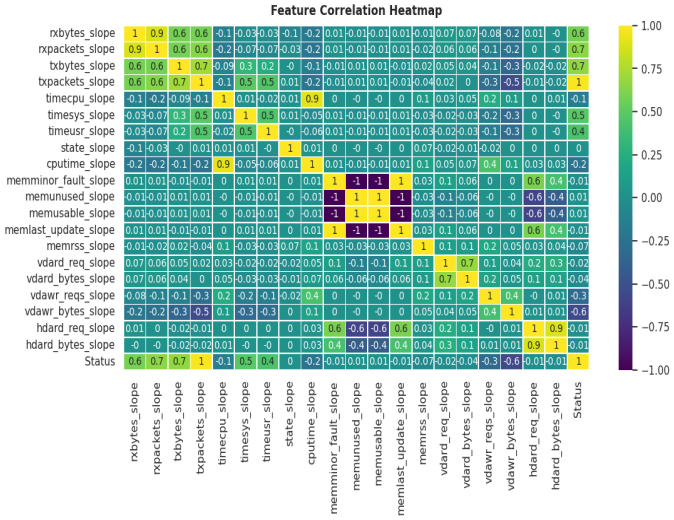

Fig. 3. Heatmap

c) Outliers: On plotting the boxplot for the data, it was observed that the data consisted of outliers that were removed to improve the statistical significance of our findings. float

d) Cluster Analysis: To further explore the data, cluster analysis was performed and elbow method was used to determine the number of clusters. The optimum number of clusters is 2. [Fig. 4] A homogeneity score of 0.87 was obtained. This score indicates how many of the clusters predicted contain only members of a single class.

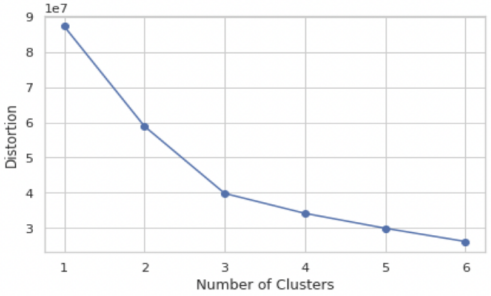

Fig. 4. Elbow Method

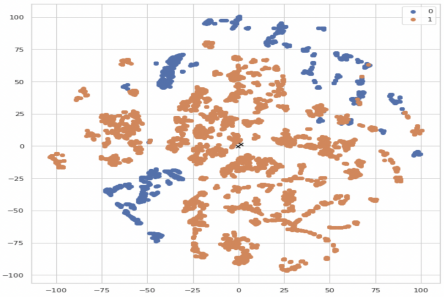

Fig. 5. Cluster Analysis

B. Data Preprocessing

The process data preparation and preprocessing consists of the following steps.

a) Data Cleaning: The data consisted of a total of 9594 rows of which 16 rows contained Null values. These rows could not be used and were therefore removed from the dataset. The dataset then consisted of 9578 rows. Of the 48 features in the data, 4 were unusable and were dropped. The columns represented the timestamp, the virtual machine identity, the unique domain identifier, and the domain name. Also, the label were strings which were converted to integer with 0 representing 'Normal' instances and 1 representing 'Attack' instances.

b) Data Normalization: The raw datatset was normalized using the StandardScaler function available in the SkLearn library in Python. The function calculates the score for each of the instances in the dataset as:

$$z = (x - u)/s$$

where u is the mean of the sample and s is the standard deviation. Standardization of the data helps the model perform better. The graph (Fig. 6) below shows the distribution of the data before and after standardization has been performed.

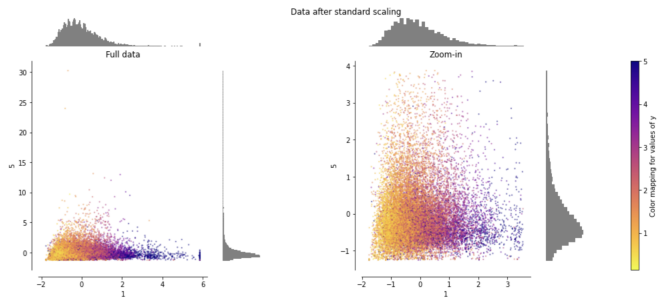

Fig. 6. Standardized Data

c) Balancing the Data: The available data consists of two kinds of instances for the network traffic- Normal or Attack. The frequency of the 'Normal' instance is 7288 and that for 'Attack' is 2306. This data is highly imbalanced and therefore any model trained on this data would be biased towards the 'Normal' instances. To overcome this problem, random oversampling was performed. Instances from the 'Attack' instance were randomly chosen and and duplicated within the dataset to get an equal distribution for both the categories. Fig. 7 depicts the distribution of the target variables after oversampling has been performed.

d) Splitting the Data: The oversampled data must be split into training and testing subsets to be fed to the model. The data was divided in a ratio of 7:3 where 70% of the data formed the training subset and 30% formed the testing subset. Fig. 8 indicates the count of instances of each type in the training and testing subsets.

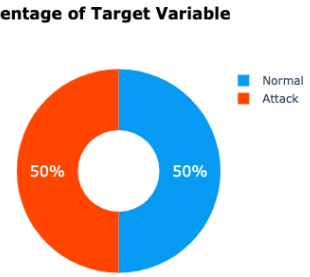

Fig. 7. Distribution of data after Oversampling

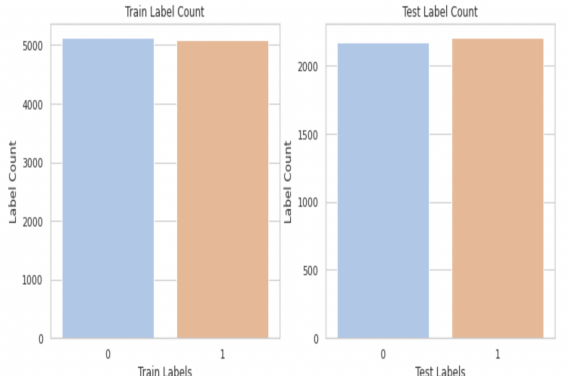

Fig. 8. Distribution in Train-Test Subsets

C. Evaluation Metric

a) Classification Accuracy: The classification accuracy of a model that summarizes the performance of the model by dividing the number of correct predictions by the total number of predictions made. It is one of the most commonly used evaluation metrics in determining the reliability of the model. Accuracy = Number of Correct Predictions / Total Number of Prediction

b) Confusion Matrix: In classification problems, confusion matrix is a way to visualize the performance of a model in table-like layout usually for supervised algorithms. Each of the columns in the matrix represent the actual classes of the data and each of the rows represent the predicted classes. The rest of the table contains the count of the predictions made by the model for each of the class labels. Using a confusion matrix is a good way to find out the classes where the model gets confused and the type of error it makes. When the data to be classified is imbalanced, a confusion matrix is a good way to evaluate the performance of the model.

c) Precision: Precision is calculated as the number of true positives identified by the model divided by the sum of the total number of true positives and the false positives. The precision value of a model tries to answer the question, out of all the positively identified instances what proportion was actually correct.

$$\text{Precision} = TP / TP + FP$$

Where, TP = true positives and FP = false positives

d) Recall: Recall is calculated as the total number of true positives identified by the model divided by the sum of the total number of true positives and the false negatives. The recall value of a model tries to answer the question, of all the actual positive instances, how many were identified correctly by the model.

$$\text{Recall} = TP / TP + FN$$

Where, TP = true positives and FN = False Negatives

e) F1 Score: The F1 score is used to compare the performance of different classifiers when one has a high recall whereas the other has a higher precision. It uses harmonic mean to combine the precision and recall into a single value which makes model comparison easier.

$$\text{F1 Score} = 2 .(P . R) / (P + R)$$

Where, P = precision and R = recall.

f) Kappa Statistic: The Kappa statistic is used to measure the inter rater relatibility which rate the same quantity. It is calculate as

$$\text{Kappa} = \text{(observed agreement-expected agreement)/(1-expected agreement)}$$

g) ROC Curve: The receiver operator characteristic (ROC) curve demonstrates the performance of a classification model using the true positive rate and the false positive rate.

D. Model Implementation

All models were run on unbalanced and oversampled data, and hyperparameter tuning was used to discover the optimal values for models. The findings and comparison of model performances based on several evaluation metrics are shown in the section below. Precision, recall, F-score, Kappa score, and accuracy are the evaluation measures.

a) Logistic Regression: Logistic regression is applied as a binary classification strategy in this study. Logistic Regression is a strong classification approach that makes some assumptions. First, it presupposes that the independent variables have no or little collinearity. Second, the logit of the result and each predictor variable have a linear connection.

$$logit(p) = log(p/(1-p))$$

Finally, observations must be independent of one another. The sigmoid function is used in logistic regression to project probability to either of the two groups. The Sigmoid function removes outliers by disregarding extreme values. Logistic Regression from sklearn package was used. The sklearn package's Logistic Regression was used in this research, and the optimum parameters from conducting Grid search on Balanced and Imbalanced data are as follows:
Imbalanced: C=1000.0, penalty='l2', solver='newton-cg'
Balanced: C=10.0, penalty='l1', solver='liblinear'
where C is inverse regularization strength Lower values suggest more regularization. Penalty is the norm of penalty and Solver is an algorithm for the optimization problem.
10-fold cross validation technique was also performed on the best results of Balanced Imbalanced data to mitigate the overfitting and the results for each iteration are shown in the Figures below.

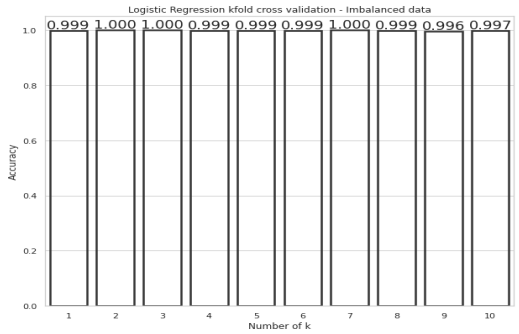

Fig. 9. K-Fold Cross Validation on Logistic Regression for Imbalanced Dataset

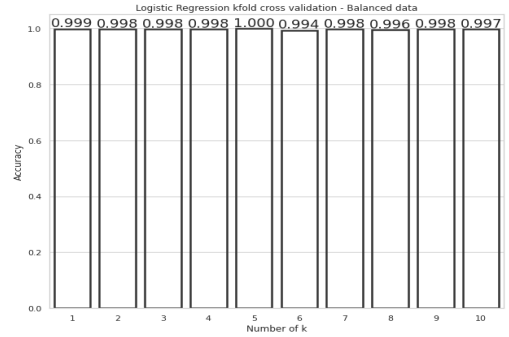

Fig. 10. K-Fold Cross Validation on Logistic Regression for Balanced Dataset

Plots for comparison are the results for several performance indicators based on balanced and unbalanced data.

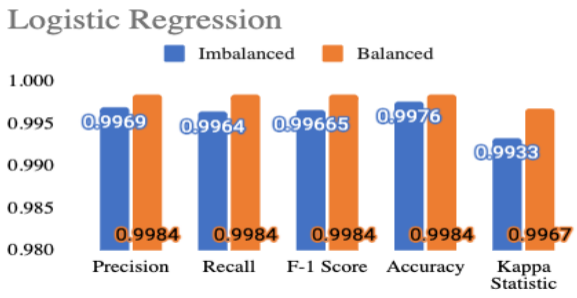

Fig. 11. Evaluation Metrics on Logistic Regression for Imbalanced vs Oversampled Dataset

As demonstrated in the bar chart above, the performance for both balanced and unbalanced data is rather good, but oversampled data performs somewhat better. In this study, we place a higher value on false negative mistakes since we want to catch all of the attacks and not miss even one of them, therefore type two errors are more important than type one errors. For the best parameters, Logistic Regression gives accuracy of 99.84% and recall(which takes into consideration FN) of 99.84% as well. The absolute values for weights estimated by the model are displayed below.

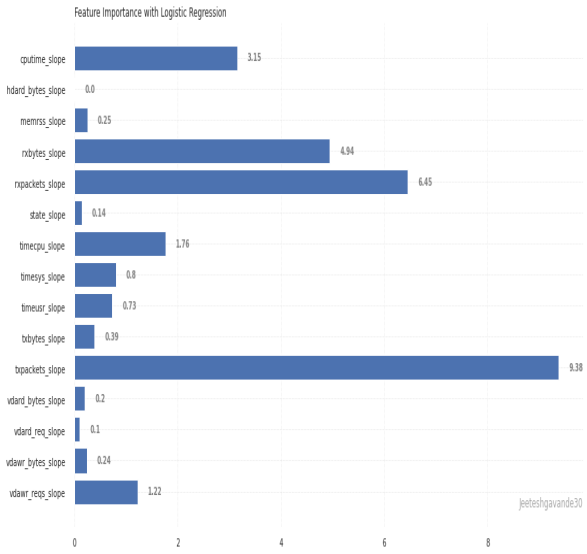

Fig. 12. Feature Importance plot for Logistic Regression

As shown in the bar chart top 3 features resulted from the best model built from sklearn Logistic Regression based on the best hyperparameters and balanced data are txpackets_slope, rxpackets_slope and cputime_slope respectively.

b) XGBoost: a popular and efficient gradient boosted trees solution. Gradient boosting is a supervised learning approach that combines the estimates of a collection of smaller, weaker models to attempt to accurately predict a target variable. Because XGBoost works in parallel, it performs quicker and is more desirable when compared to other trees. The data for this study is sparse, and XGBoost works well with sparse data. In this investigation, XGBoost produced extremely strong results, with assessment metrics offering 99 percent accuracy and k-fold cross validation with k = 10 done to ensure that over-fitting with training data did not occur, and the average accuracy remained 99 percent.XGBoost, like Logistic Regression, was applied to unbalanced and oversampled datasets, with gridsearch used to find the optimal parameters in both scenarios. Below are the bar charts for kfold cross validation for balanced and imbalanced datasets.

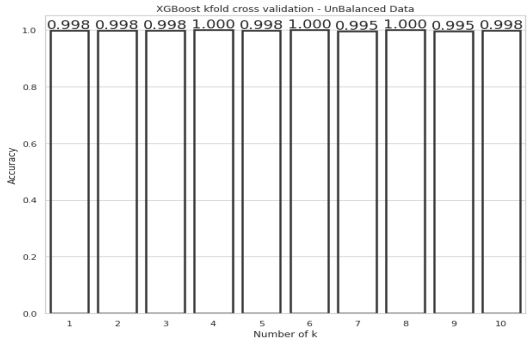

Fig. 13. K-Fold Cross Validation on XGBoost for Imbalanced Dataset

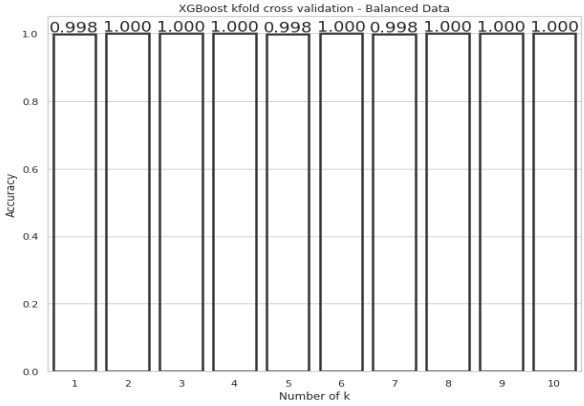

Fig. 14. K-Fold Cross Validation on XGBoost for Balanced Dataset

As shown in Figure 15 the performance metrics for balanced data is showing better results so the best model is built for XGBoost based on oversampled data. For generating the model, hyper parameter tuning was conducted, and the settings for the balanced dataset are colsample bytree= 0.5, gamma= 0, learning rate= 0.01, max depth= 5, reg lambda= 0, scale pos weight= 5, subsample= 0.8.

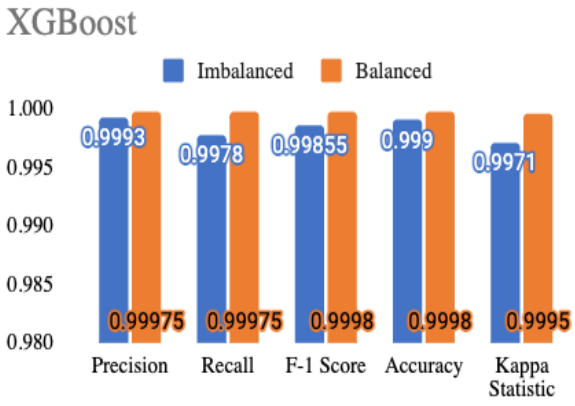

Fig. 15. Evaluation Metrics on XGBoost for Imbalanced vs OverSampled Dataset

Perfoming XGBoost on best parameters reveals the most significant features.

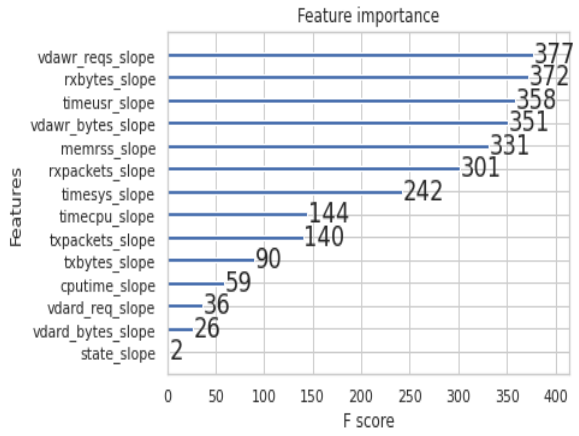

Fig. 16. Feature Importance plot for XGBoost

As its shown in Fig16 the three top important features are vdawr_reqs_slope, rxbytes_slope, timeusr_slope respectively. Comparision between Imbalanced and Balanced XGBoost data modeling is visualized in Figure 16.

c) Decision Tree: it is a popular supervised learning algorithm used for regression and classification problems. It involves creating a tree like structure where each of the nodes on the tree represent a test on the attribute, each branch represents an outcome on the decision and each of the leaf nodes represents the decision taken after computing all the attributes. The decision tree was implemented for the raw (Fig. 17) as well as oversampled data (Fig. 18) and using both the gini impurity and entropy and information gain parameters with grid search. The process of post-pruning was also performed on both the trees to observe if it had any affects on the performance of the model. The tree generated using the gini impurity and entropy gave accuracies of 100% each and overfitting was suspected. Therefore post-pruning was performed and the accuracy came down to 99% which is not a significant improvement. The trees were also generated on the oversampled data to observe if there was any improvement in the performance when the data was not skewed.The samples selected by the pruned tree match the ones generated by the feature selection in XGBoost and confirm the findings that these are important features in the determination of network traffic being malicious or not. The learning rates are set at 0.010 for the pruned decision tree created using the unbalanced data(Fig. 19) and 0.0010 for the pruned decision tree created using the balanced dataset (Fig. 20).

An evaluation of the generated trees has been visualized in Fig. 21 on various performance metrics and it can be seen that that the pruned decision tree generated using the balanced dataset performs better than the others.

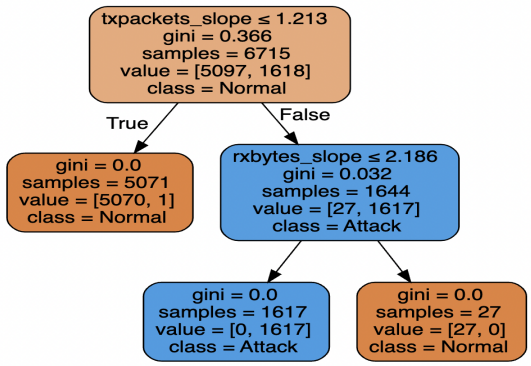

Fig. 17. Grid Search DTree on UnBalanced Data

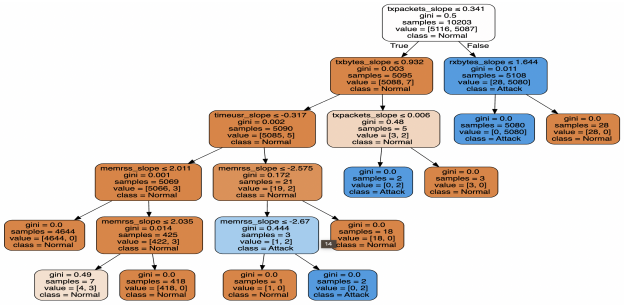

Fig. 18. Grid Search DTree on Balanced Data

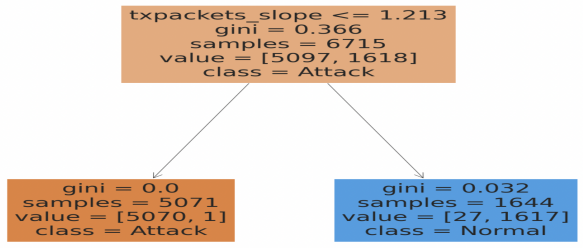

Fig. 19. Pruned DTree on UnBalanced Data

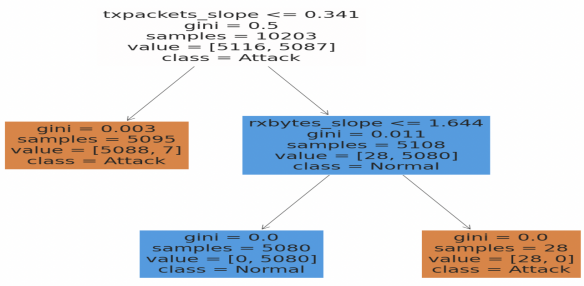

Fig. 20. Pruned DTree on Balanced Data

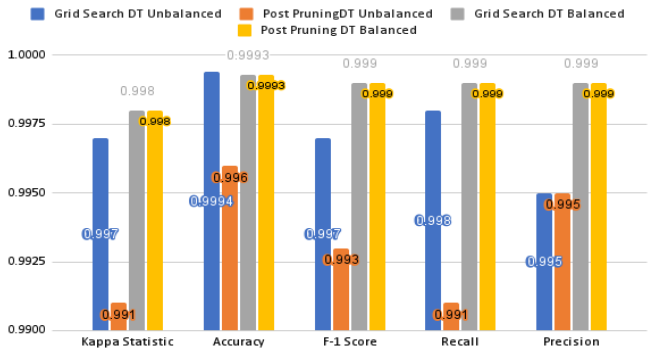

Fig. 21. Evaluation of DTree

d) Support Vector Machine: The Support Vector Machine is a Machine Learning model developed by Vapnik (1995). SVM is one of the most popular supervised learning algorithms used for binary classification problems. Our project is a binary classification problem of detecting attack or normal in Cloud status. SVM maximizes the gap between two classes. The training data in form (xi,yi) with weights w and threshold b. Then w.x + b = 0 would be the equation of the separating hyperplane and w.x + b = ±1 would be the equations of the hyperplanes that should satisfy all the data points such that the margin is maximum. Above 1 and below -1 can have the data points such as 'Attack' or 'Non-attack'. SVM optimization is a convex optimization problem that can be solved by introducing the Lagrange multiplier: L(x,y,) = f(x,y) - (g(x,y)-c) where c is constant for margin maximization and error minimisation . A grid search method is applied fort hyper parameters tuning among 'C':[0.001,0.01,0.1,0.5,1, 10, 100, 1000],'gamma':['scale','auto'], 'kernel': ['linear','rbf']. The best parameters found are 'C': 100, 'gamma': 'scale', 'kernel': 'linear'. A loaded SVM model is implemented with the best parameters. In the second step, a balanced dataset is used after over sampling the minority class for grid search and the SVM implementation. In the third step, a subset of selected features is created according to Random forest feature importance and applied to the SVM model. The accuracy was observed 99% in all three cases. The evaluation metrics for SVM are presented in Fig. 22.

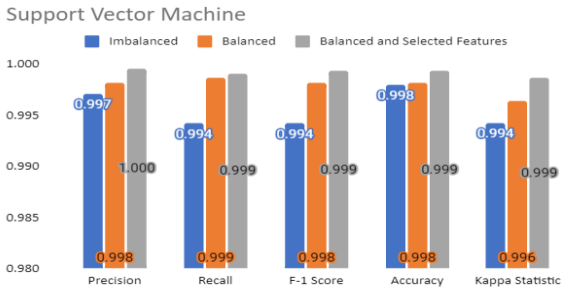

Fig. 22. Evaluation Metrics for SVM

e) Random Forest: it is a machine learning algorithm that is used to solve both regression and classification problems.Outcome of random forest is established based on the predictions of decision trees. It makes the predictions by taking the average or mean of the output from various trees.The precision of random forest algorithm is directly proportional to the number of trees.Random Forest Algorithm is implemented for the raw and over-sampled data.
The accuracy of this model on raw and over-sampled data was observed to be around 99%.The evaluation metrics are presented in Fig 23.

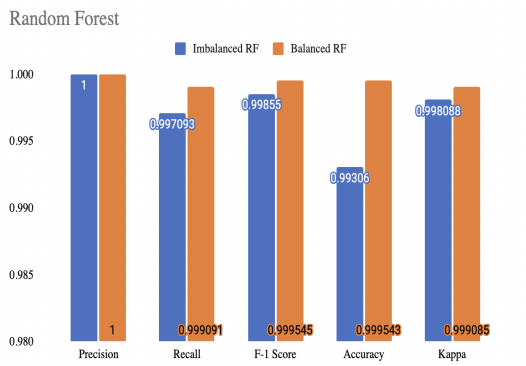

Fig. 23. Evaluation Metrics for Random Forest

f) Naive Bayes: is a generative learning algorithm that is used for classification of attack or normal based on the prior probability. Bernoulli Naive Bayes from sci-kit learn library is used for implementation. The prior for unbalanced case was [0.76 , 0.24] and for balanced case was [0.50, 0.50]. The parameter alpha was set to 1 in both cases. The accuracy has increased after balancing the data. The evaluation metrics are presented in Fig. 24.

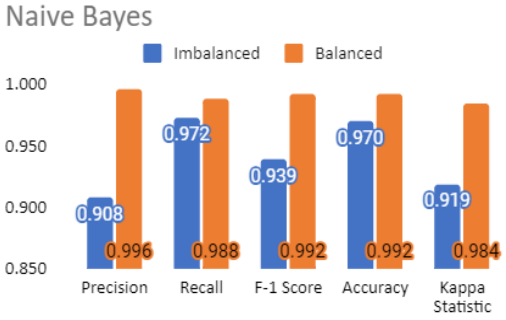

Fig. 24. Evaluation Metrics for NB

g) K-Nearest Neighbours: The K-Nearest Neighbor is a non-parametric based machine learning algorithm that is used for classification based on k closest data points in the data set. A test data point is to be classified mathematically as +1 (attack) or -1 (non-attack). That makes the K-NN to be

$$yt = sgn(\sum_{i=1}^{k} y_i)$$

where (xi,yi) are the k nearest neighbors of the test data. The distance between the neighbors are basically determined by the Euclidean method. The k here is determined using the number of training data points near the test data point that is most likely to be classified as an attack. It has a high computational complexity but provides good accuracy in real world classification problems. KNN coupled with Genetic Algorithm (GA) is proven to be a more powerful classifier which may provide more accuracy.

The algorithm was applied on raw data set and accuracy was computed to be as 99.72%. In order to reduced the Root Mean Square error and reduce the issue with the data, Ridge and Lasso regression is used and the accuracy has increased slightly. The error rate for each value of k in range 1-30 is calculated and the best value of k apart from k=1 was found out to be as k=11 with least error value as seen in Figure 25. Research has proven that KNN coupled with Genetic Algorithm(GA) is the best approach for Cyber Attack Detection, thus, GA is applied to find the accuracy being 99.82% but GA takes more processing time comparatively. The same process is repeated on over-sampled data set. The accuracy was 99.62%. The same pattern was seen. The accuracy increased with Ridge and Lasso regression. The GA improved the accuracy as it is seen upto 99.88% accuracy. The other metrics such as Precision, Recall, F-1 score, Kappa statistic are also compared apart from accuracy for raw and balanced data are as seen in Figures 26 and 27 below.

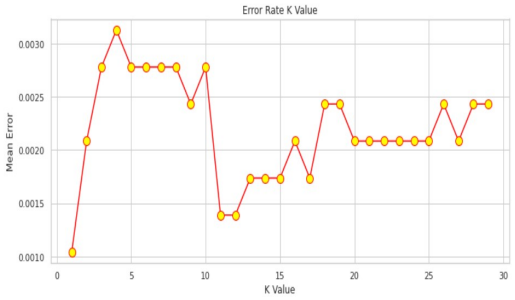

Fig. 25. Finding the best k-value

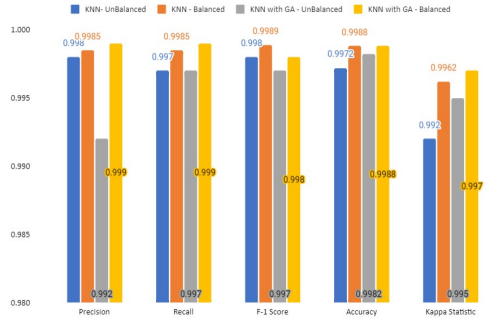

Fig. 26. Evaluation Metrics for KNN and KNN with Genetic Algorithm

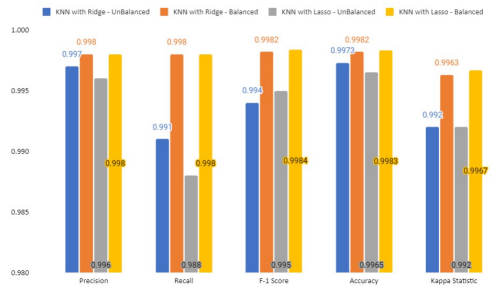

Fig. 27. Evaluation Metrics for KNN - Ridge and Lasso

V. Results & Analysis

All the models were implemented on the unbalanced and the balanced data set each, and their precision, recall, F-1 Score and accuracy were calculated and compared. The following figures 27 and 28 shows the graphical form of the performances of the models and helps perform sound comparison. It can be seen that the performance of the model

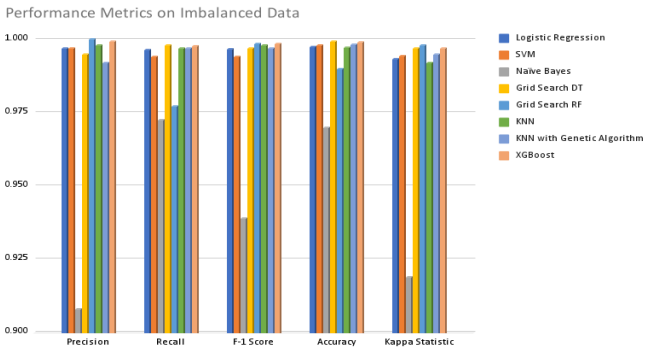

Fig. 28. Performance Metric on Imbalanced Data

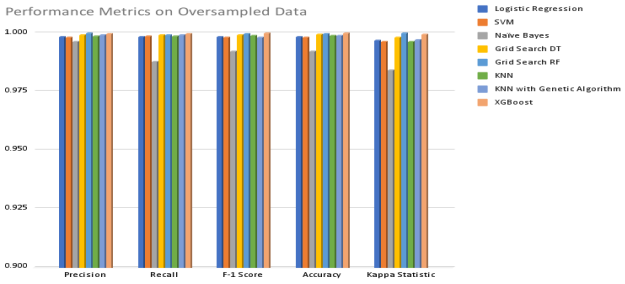

Fig. 29. Performance Metrics on Oversampled Data

improves significantly once the data has been oversampled and the target classes are in equal proportions. Although the metrics for the other models are comparable, the performance of Naive Bayes shows significant improvement.

The ROC Curve is presented for all the Classifiers in Fig. 30.

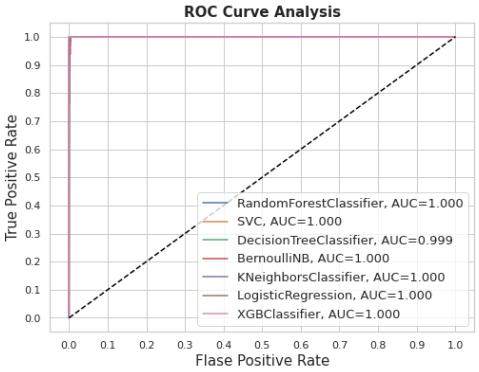

Fig. 30. ROC for all models

VI. Key Learnings

Machine learning is a vast field growing exponentially and has application in almost every field. IDS systems make use of machine learning to improve network security by detecting and preventing cyber attacks that pose a threat to the data. It was observed that while decision tree performs well, if not pruned, it also takes into account irrelevant features. Pruning the trees helps get rid of these unnecessary attributes and saves time spent on training a huge model. KNN coupled with genetic algorithm will improve the classification performance as it calculates the similarities between the training and testing samples at each level. In cyber attack detection the target variable is highly correlated to some features. SVM can classify detection with high accuracy not only with unbalanced dataset but also with a subset of important features.

VII. Conclusion & Future Works

This project was aimed at creating an Intrusion Detection System (IDS) which classifies network traffic as either normal or malicious based on its features such as the rate bytes received, the rate at which packets were transmitted, rate of transmission error, etc. The data was explored and understood using various exploratory methods and pre-processed to improve overall data quality. Multiple machine learning model such as logistic regression, naive bayes, decision tree, random forest, K-Nearest Neighbors, support vector machine, and XGBoost were implemented and their performances compared. It was observed that all models perform well with an accuracy of over 99% . Of all the models, XGBoost performs the best with the highest accuracy. The model is out of the scope of this course so other metrics were evaluated to suggest another model. The next metric, then, to evaluate the performance would be the Recall value. Since the model should not classify an attack as normal network traffic, the false negatives must be as low as possible. Based on recall, the models that perform the best are SVM, Decision Tree and KNN with Genetic Algorithm with recall values of 0.999 each.

This project can be improved by deploying it to detect attacks happening in real time. Streaming the data to the model directly can help prevent intrusions into the network and using prediction algorithms, attacks can be prevented from ever happening. Research suggests that a Hybrid approach combining SVM and KNN on real-time streaming data is the best solution for identifying attacks and protecting the cloud. Although network security systems are improving and becoming more robust, so are the attacks targeted at the networks. Hence it has become extremely important to come up with systems that protect the network and the data stored within them.

Acknowledgment

We would like to express our sincere gratitude to Professor Vishnu Pendyala for his guidance that aided in the completion of this project and helped us understand machine learning concepts using practical examples. We would also like to thank Nikita Balani and Vineeth Reddy Chinthala for their continued help and support throughout the development of the project.

References

[1] "What is cloud computing? A beginner's guide: Microsoft Azure," What Is Cloud Computing? A Beginner's Guide — Microsoft Azure. [Online]. Available: https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/. [Accessed: 13-Mar-2022].

[2] Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," Journal of Big Data, vol. 8, no. 1, 2021.

[3] T. Salman, D. Bhamare, A. Erbad, R. Jain and M. Samaka, "Machine Learning for Anomaly Detection and Categorization in Multi-Cloud Environments," 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), 2017, pp. 97-103, doi: 10.1109/CSCloud.2017.15.

[4] M. Zekri, S. E. Kafhali, N. Aboutabit and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), 2017, pp. 1-7, doi: 10.1109/CloudTech.2017.8284731

[5] A. Abusitta, M. Bellaiche, and M. Dagenais, "An SVM-based framework for detecting DOS attacks in virtualized clouds under changing environments," Journal of Cloud Computing, vol. 7, no. 1, 2018.

[6] S. Gumaste, D. G. Narayan, S. Sindhe and K. Amit, "Detection of DDoS Attacks in OpenStack-based Private Cloud Using Apache Spark", Journal of Telecommunications and Information Technology, 2020, doi:10.26636/jtit.2020.146120

[7] R. Grover, C. R. Krishna, A. K. Mishra, E. S. Pilli and M. C. Govil, "A Comparison of Analysis Approaches for Cloud Forensics," 2016 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 2016, pp. 131-135, doi: 10.1109/CCEM.2016.031

[8] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[9] P. Bedi, N. Gupta, and V. Jindal, "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," Applied Intelligence, Sep. 2020, doi: 10.1007/s10489-020-01886-y.

[10] N. Usman, S. Usman, F. Khan, M. A. Jan, A. Sajid, M. Alazab, and P. Watters, "Intelligent dynamic malware detection using machine learning in IP reputation for forensics data analytics," Future Generation Computer Systems, vol. 118, pp. 124–141, 2021.

[11] A. Mishra, "Intelligent Dynamic Malware Detection using Machine Learning in IP Reputation for Forensics Data Analytics," Medium, 28-May-2020. [Online]. Available: https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234. [Accessed: 13-Mar-2022].

[12] Google, "Classification: Precision and Recall — Machine Learning Crash Course — Google Developers," Google Developers, Mar. 05, 2019. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall.

APPENDIX

A. Visualisations

Different visualizations have been created to understand the data, the affects of the pre-processing steps on the data, visualizing model behavior and output, and for the evaluation of the performance metric of all the models.

B. Presentation

An interactive presentation was created using Prezi which depicts all the major steps and milestones of the project.

C. Significant To Real World

The significance and importance of the project has been described above in Section II of the report.

D. Saving the Model for Demo

Loaded models were created by using 'joblib' library.

```
svm_sf.fit(X_train_sf, y_train_sf)

SVC(C=100)

dump(svm_sf, 'svm_sf.joblib')
loaded_svm_sf = load('svm_sf.joblib')

y_pred_svm_sf = loaded_svm_sf.predict(X_test_sf)
```

Fig. 31. Loded models

E. Code Walk-through

A detailed explanation of the code will be provided during the presentation for this project. The loaded models will be run and demonstrated during the presentation.

F. Report

A detailed report has been created with contributions from all the team members on LaTeX in IEEE format. We ensured to include the important aspects and conclusions of our approach.

G. Version Control

The codes are publicly accessible in GitHub. Link: https://github.com/ShamamaAfnan/Supervised-Learning-for-Attack-Detection-in-Cloud.Fig32 consists of our project repository.

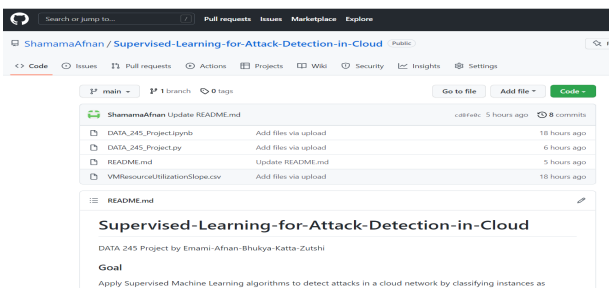

Fig. 32. GitHub

H. Lessons Learned

The key learnings from the research and implementation of this project has been included in detail in Section VI of this report.

I. Prospects of winning Competition / Publication

Detection of attacks can help build more secure cloud platforms aiming at the root cause, as detection can identify the factors responsible for attack. We aim at identification of such parameters that provide an opportunity for publication of the same.

J. Velocity

Real-time streaming data can be captured every 2s, with features like source, destination bytes, errors etc. This data is captured using RDD from dstream. Map Reduce is applied for each connection i.e. the type and service. This accounts for one data item upon which we can apply our attack detection algorithm.
Streaming can be even done using packet sniffer, where the raw sockets are bind to neutron nodes and the data link is captured. TCSP/IP is used to retrieve by parsing the Ethernet. Another way is to use Spark as a service on Cloud. Map reduce clusters with many slave nodes can be used to collect data from all the IPs using Hadoop name node format on master node.

K. Innovation

Feature selection: The current research considers only few features like data bytes to and from the source and destination, the metrics on connections only. Accommodating errors and memory strain is unique. But it seems to not play a major role, but none of the works mention that.
No research has been done on the chosen data-set.
No research on using Pruning techniques on Decision Trees.

L. Evaluation of Performance

A detailed description of the evaluation metrics used has been discussed in Section IV Part C of the report, they have also been visualized for each of the models in their respective sections and a comparative study has been performed Section V.

M. Teamwork

The project was created with the collaboration of all the team members. A shared space on Google Drive was created for everyone to contribute and share their work. The report was created on LaTeX where every member helped in the drafting process, the code was written and implemented on Google Colab with everyone's contribution, and the presentation was created using Prezi and shared within the team to work on.Fig33 shows the distribution of work among the team.

Work Distribution

Member	Task
Deepali Zutshi	Data Preprocessing, Data Visualization, Decision Tree
Poojitha Katta	Data Preprocessing, Data Visualization, KNN + GA
Purnima Bhukya	Data Preprocessing, Data Visualization, Random Forest
Shamama Afnan	Data Preprocessing, Data Visualization, SVM, Naive Bayes
Yasaman Emami	Data Preprocessing, Data Visualization, Logistic Regression or Xgboost

Fig. 33. Work Distribution of Team members

N. Technical difficulty

The bulk of the available datasets are imbalanced for the purposes of this study since benign traffic outnumbers malignant traffic in nature. To address this issue, we either increase malicious samples or reduce normal traffic. There are several approaches for dealing with imbalance datasets. The data is a collection of features of network traffic and understanding their significance and relation took quite a lot of time in terms of research.

O. Practiced Pair Programming

Google Colab was used for code collaboration. Fig34 shows the colab workspace.

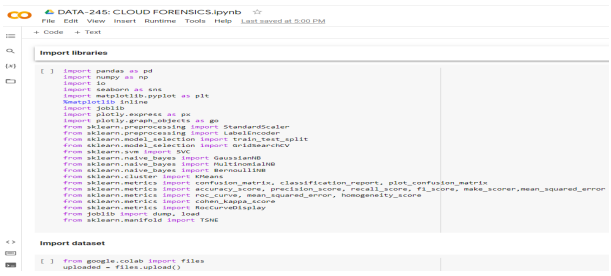

Fig. 34. Google Colaboratory platform

P. Practiced Agile

Agile Management tool is used for managing the sprints.Fig35 shows the Agile Dashboard of our project. We used Trello to incorporate MLOps process cycle in weekly sprint fashion.

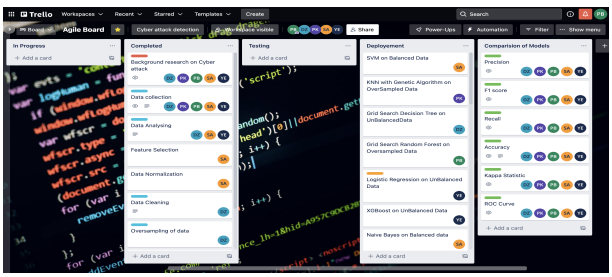

Fig. 35. Agile DashBoard

Q. Used Grammarly

We checked the content in Grammarly for fluency and language corrections as seen in the Fig 36.

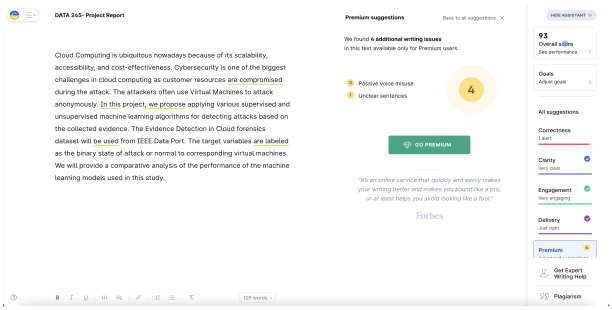

Fig. 36. Grammarly Software

R. Used LaTeX

Fig 37 shows the usage of an online Latex editor, Overleaf in creating this report.

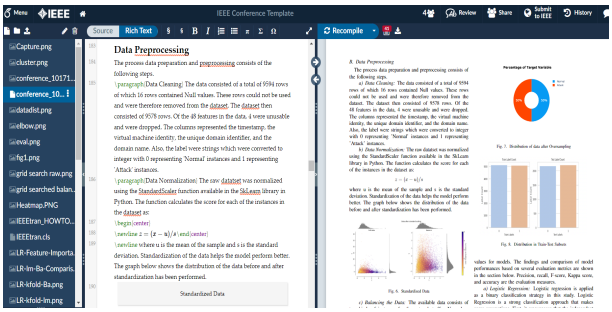

Fig. 37. LaTex Editor

S. Used Creative Presentation Techniques

Prezi is used for making presentation [Fig. 38].

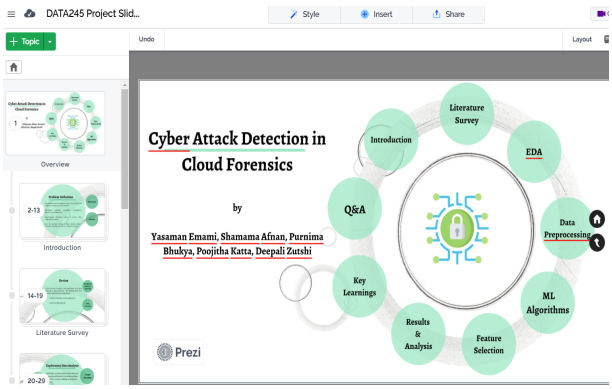

Fig. 38. Prezi Presentation

T. Literature Survey

As part of the review of previous works and researching the topics, fifteen journal and conference papers were studied and reviewed to get a better and deeper understanding of the topic and the available methods to solve the problem. A brief summary of the findings, the advantages, and disadvantages has been provided in the paper.