# CMPE 255 Data Mining

Project: Network Intrusion Detection

Spring 2021

Gheorghi Guzun

Group #1

- Yi Hu (015239913) yi.hu@sjsu.edu
- Arun Hiremath (014527877) arungangayya.hiremath@sjsu.edu
- Xin Miao (3086569) xin.miao@sjsu.edu

Github Link: https://github.com/emilyfoxhu/CMPE255-group1-project.git

# 1. Introduction

## Motivation

Nowday, the internet is a necessary thing for people's lives. There is tons of data that is transferred on the net, through those information transferring, network connections are critical. Therefore, network security will be an intensive problem. There could be a huge increase of network intrusions happening due to expansion of internet usage. How to distinguish normal connections and internet attacks will be difficult due to the large amount of requests on the internet. Hence, what if we could predict whether the connection is toxic or normal by analyzing the connection's attributes and features? A smart classifier to distinguish those connections could prevent harmful intrusion to damage the website and server.

## Objective

The objective of this project is to build some classifiers to distinguish computer network intrusions from normal connections. Given the basic features of TCP connections, content features, and traffic features, we need to predict if the connection is a normal or a malicious one. By testing multiple models to predict the dataset, we would conclude a few best performance models and try to explain why it performs so well.

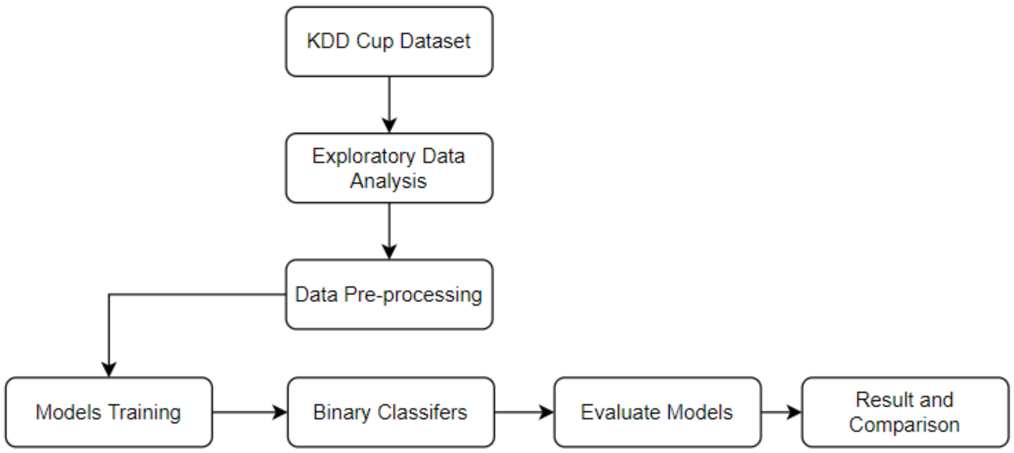
## Literature review

Based on World Intrusion Detection and Prevention Systems Market 2020 – 2025 report from Research and Markets, the IDPS market produced about USD 4.8 billion in 2020, and it will grow to USD 6.2 billion by 2025, that is a large increasing due to an increasing number of security breaches and cyberattacks. Since big data is used in many scenarios and fields, data protection is becoming more important. Furthermore, the popularity of cloud computing is causing more potential cyberattacks. Hence, the technology of identifying network intrusion quickly and smartly is valuable and economical.

# 2. System Design and Implementation details

## Architecture Design

The following figure shows the architecture of the system. After analyzing and preprocessing the datasets, we have applied Logistic Regression, Decision Tree, Random Forest, SVM, Naive Bayes classifiers for binary classification. Finally, we compared the accuracy, true positive rate, true negative rate, f1-score, etc. of these classifiers.



## Algorithms Selected

- Logistic Regression
- Naive Bayes
- Random Forest
- Decision Tree
- Support Vector Machine
  The above five algorithms are selected to perform classification on network intrusion dataset. Random Forest is an ensemble algorithm, the rest of the algorithms are base classifiers.

## Technologies and Tools Used

| **Tools/Technologies** | **Comments** |
|---|---|
| Jupyter | Workbench for complete implementation |
| Pandas library | Pandas provides tools for data manipulation. We use it to read data sets and create dataframe. |
| Numpy library | Numpy is a library containing multi-dimensional array and matrix structures. We use it to perform advanced mathematical operations on arrays and matrices. |
| Skit-learn library | Skit-learn provides many machine learning algorithms. We use it to perform data preprocessing, classification algorithms, and metrics. |
| Seaborn | Seaborn provides a higher level of interfaces for plotting graphs based on matplotlib. We use it to create heatmap during exploratory data analysis, and in generating result metrics. |

| | |
|---|---|
| Matplotlib library | Matplotlib is a plotting library which provides functionalities similar to MATLAB. We use it to create graphs during exploratory data analysis. |

## Use Cases

A current use of this would be to monitor the network data in real time and potentially flag the malicious network packets. This ensures that the network is secured.

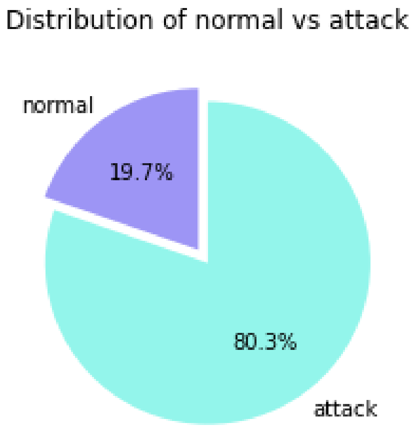
# 3. Experiments and evaluation

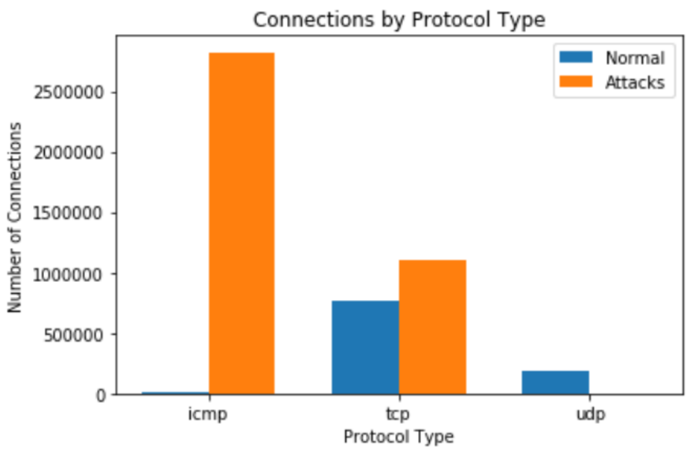
## Dataset

The dataset for the project is KDD Cup 1999. It is a simulation of the military computer network. Each record in the training and test data set stands for a network connection, and is labeled with either “normal” or “attack” with a specified attack type. The intrusion can be mainly categorized into 4 types: Denial of Service (DOS), unauthorized access from remote machines (R2L), unauthorized access to local root privileges (U2R), and Probing.

| Data Set | Features | Records |
|---|---|---|
| Training Data Set | 42 | 4,898,431 |
| Test Data Set | 42 | 311,029 |

However for this project, we aim to classify the connection as either “Normal” or “Malicious” one. For this label from the data set is converted to reflect the target label. After the binary label conversion the data set comprises 396,742 records belonging to label “Malicious” and 97278 records of type “Normal”. The graph on the right side shows the distribution of normal and malicious connections of different protocol types.





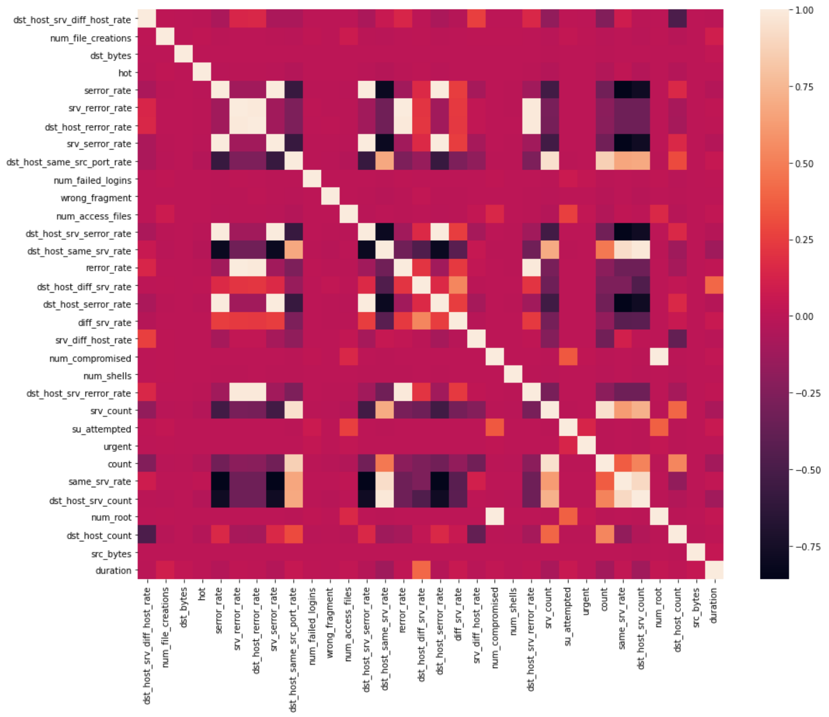
## Exploratory Data Analysis:

### Data Pre-processing:

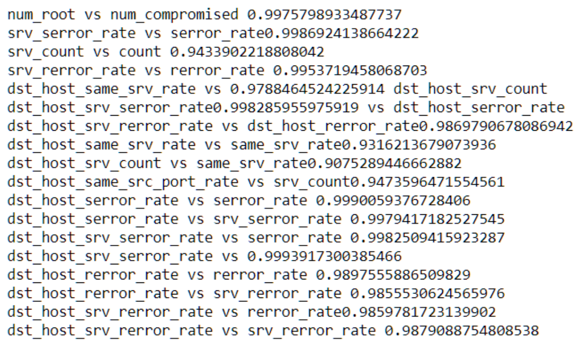
In the preprocessing stage, we have used a combination of data analysis and statistical computing techniques to filter the data set. The initial data set has 38  columns 4898431 records with no missing values in any  records. The initial analysis showed us that we had 4 categorical fields, and 4 binary fields. The rest 29 columns represented continuous data.

Of the 4 categorical fields, we found “flag” has a very high cardinality ratio and num_outbound_cmds has only one type of value. These two values were eliminated from the data set. The other two categorical values service and protocol type were converted into numeric values using a Label encoder. Initially we did our experiment, by not including the service feature in our classification due to high cardinality. However, from our data analysis we realized that most of the records classified as malicious belonged to service type “private”.

For exploring the continuous values and binary values, we relied on the coveriation of the feature set. We generated a heatmap to visualize the covariation between various features.



A careful analysis indicates that many of the variables have a significant correlation with each other. The feature set and the numerical value is as follows. The features with significant correlation were filtered and removed from the data set.



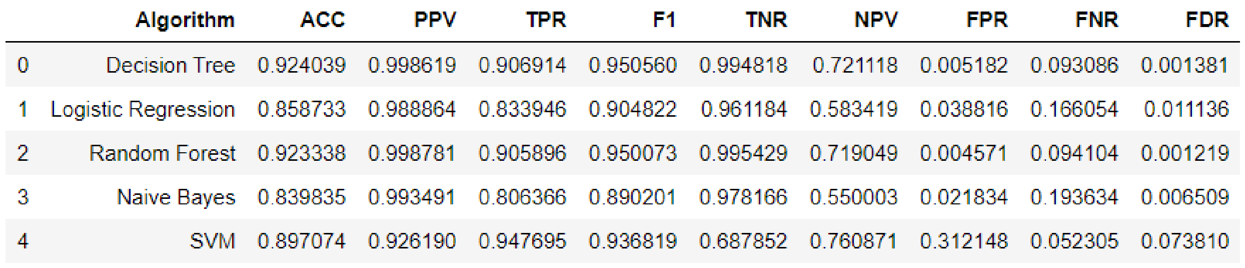
```
num_root vs num_compromised 0.9975798933487737
srv_serror_rate vs serror_rate0.9986924138664222
srv_count vs count 0.9433902218808042
srv_rerror_rate vs rerror_rate 0.9953719458068703
dst_host_same_srv_rate vs 0.9788464524225914 dst_host_srv_count
dst_host_srv_serror_rate0.998285955975919 vs dst_host_serror_rate
dst_host_srv_rerror_rate vs dst_host_rerror_rate0.9869790678086942
dst_host_same_srv_rate vs same_srv_rate0.9316213679073936
dst_host_srv_count vs same_srv_rate0.9075289446662882
dst_host_same_src_port_rate vs srv_count0.9473596471554561
dst_host_serror_rate vs serror_rate 0.9990059376728406
dst_host_serror_rate vs srv_serror_rate 0.9979417182527545
dst_host_srv_serror_rate vs serror_rate 0.9982509415923287
dst_host_srv_serror_rate vs 0.9993917300385466
dst_host_rerror_rate vs rerror_rate 0.9897555886509829
dst_host_rerror_rate vs srv_rerror_rate 0.9855530624565976
dst_host_srv_rerror_rate vs rerror_rate0.9859781723139902
dst_host_srv_rerror_rate vs srv_rerror_rate 0.9879088754808538
```

## Methodology

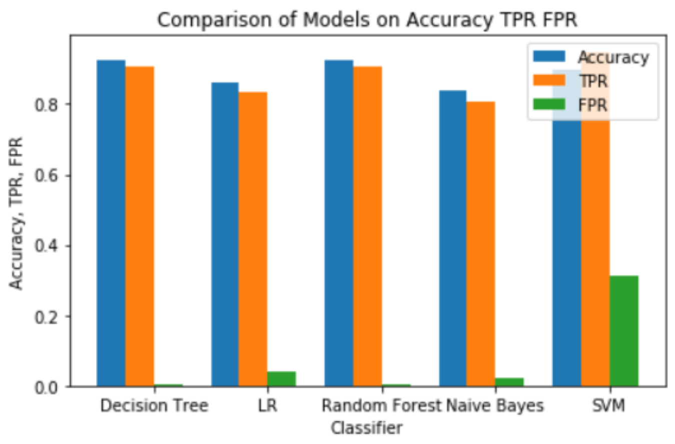
The KDD data set comes with two disjoint labelled sets. The train data set is used for training on classifiers and test data set is used for label generation. While experimenting and training with the data set, we have split the kdd.train data set in to train and test sets. The classification is carried out on 5 Classifiers as described in the algorithm selection section.

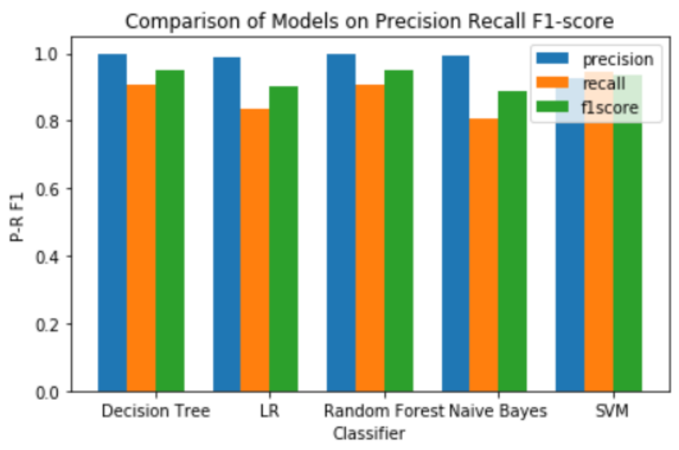
## Graphs

The following 5 algorithms are used and compared with 10 different metrics: overall accuracy(ACC), precision(PPV), recall(TPR), f1-score(F1), true negative rate(TNR), Negative predictive value(NPV), false positive rate(FPR), False negative rate(FNR), False discovery rate(FDR), and Receiver Operating Characteristic Curve(ROC).



|  | Algorithm | ACC | PPV | TPR | F1 | TNR | NPV | FPR | FNR | FDR |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Decision Tree | 0.924039 | 0.998619 | 0.906914 | 0.950560 | 0.994818 | 0.721118 | 0.005182 | 0.093086 | 0.001381 |
| 1 | Logistic Regression | 0.858733 | 0.988864 | 0.833946 | 0.904822 | 0.961184 | 0.583419 | 0.038816 | 0.166054 | 0.011136 |
| 2 | Random Forest | 0.923338 | 0.998781 | 0.905896 | 0.950073 | 0.995429 | 0.719049 | 0.004571 | 0.094104 | 0.001219 |
| 3 | Naive Bayes | 0.839835 | 0.993491 | 0.806366 | 0.890201 | 0.978166 | 0.550003 | 0.021834 | 0.193634 | 0.006509 |
| 4 | SVM | 0.897074 | 0.926190 | 0.947695 | 0.936819 | 0.687852 | 0.760871 | 0.312148 | 0.052305 | 0.073810 |





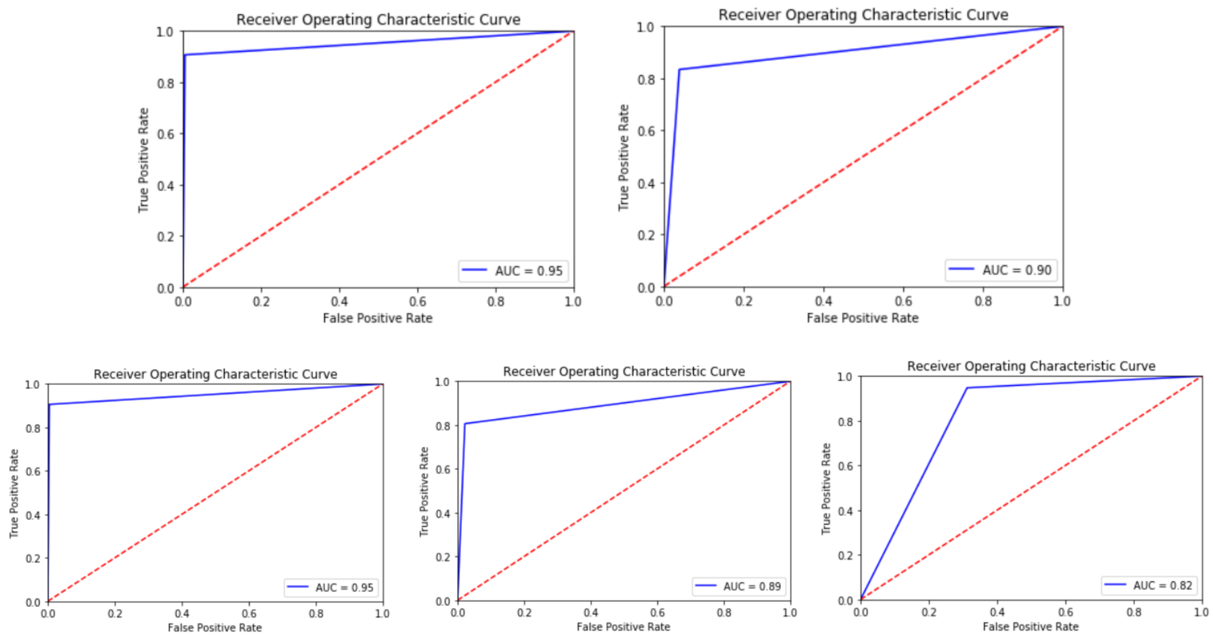


Figure: ROC Curve for Decision Tree, LR, Random Forest, Naive Bayes, SVM

## Analysis of the results

- Accuracy: the decision tree classifier has the highest accuracy, followed by random forest and SVM.
- True positive rate: SVM has the highest true positive rate, followed by decision tree and random forest.
- False positive rate: decision tree classifier has the lowest false positive rate, followed by random forest and naive bayes.
- F1-score: decision tree has the highest f1-score, followed by SVM and random forest.
- AUC: decision tree and random forest classifiers have higher Area Under the Curve compared to other algorithms.

# 4. Discussion and Conclusions

## Difficulties faced and Decisions made

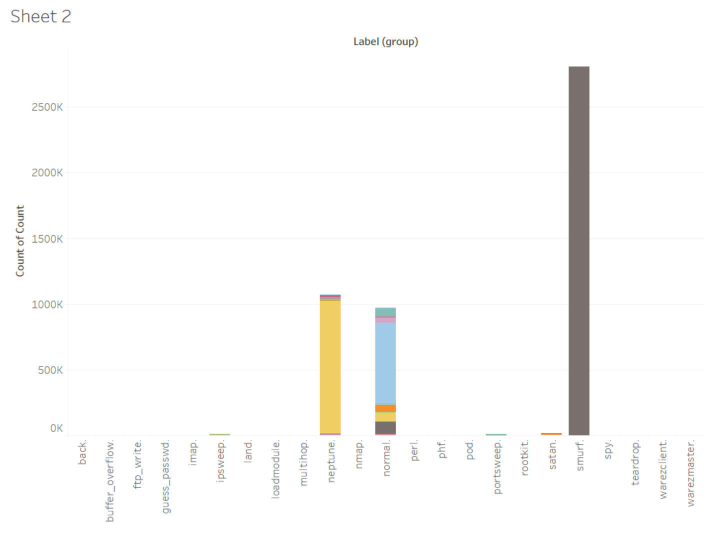
Our initial proposal was to build a multi class classification algorithm. However, because of the complexity in result analysis and due to time constraints, we decided to drop the idea. Instead, we focused more on the data-set exploration, data set preprocessing, classification and result analysis.

## Things that worked

Our initial experimentation with 35 feature sets included continuous variables that had high correlation among them . So we had an initial accuracy varying between 78% to 88% among all the classification algorithms. Once additional preprocessing was done to remove the continuous variables based on the correlation value, our accuracy increased by 5% more on each of the classification algorithms.

## Things that didn’t work well

The use of “service” categorical features. This feature had the below composition for various attack types. Since one significant value played a major role for malicious labels, we tried to convert it to a continuous value and use it in the classification training. However, it reduced the accuracy of our test data. Eventually, owing to its high cardinality ratio we decided to drop this feature from the train and test data set.



Count of Count for each Label (group). Color shows details about Service.

### Conclusion

Based on the analysis of the results above, we can conclude that overall, Decision Tree, Random Forest and Support Vector Machine algorithms perform better at classifying network connections. Naive Bayes and Linear Regression have lower performance.

## 5. Project Task Distribution

| **Task** | **Responsibility** |
|---|---|
| Data Pre-processing | All |
| Exploratory Data Analysis | All |
| Random Forest | Arun |
| Logistic Regression, Naive Bayes | Xin |
| Decision Tree, Support Vector Machine | Yi |
| Integration | Arun, Yi |
| Project Report and Presentation Slides | All |

## 6. References

1. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
2. https://www.globenewswire.com/news-release/2020/08/21/2081840/0/en/World-Intrusion-Detection-and-Prevention-Systems-Market-2020-2025.html
3. https://nycdatascience.com/blog/student-works/network-intrusion-detection-2/