Cyber Security of Critical Infrastructure
Assignment 5 Report
KDD CUP-99 Dataset

Gouni Anuraag Kiran(18111017)
Hariom(18111019)
M V Ravi Kumar(18111037)
Shikhar Barve(18111065)
Siddharth Kumar(18111068)
Kamlesh Kumar Biloniya(160317)

In our Assignment, we are provided with Knowledge, Discovery and Data mining tool competition(KDD-99) Dataset held in 1999. Our task is to build a predictive model for intrusion detection which can distinguish between bad connections(intrusions or attacks) and good(normal) connections. :-

- Number of features = 41 some of which are higher-level features that help in distinguishing normal connections from attacks.
- There are several categories of derived features.
  1. ”Same host” features examine connections in the past two seconds that have the same destination host as the current connection.
  2. ”Same Service” features examine connections in the past two seconds that have the same service as the current connection.Above two categories are together called time-based traffic features of the connection records.
  3. ”host-based traffic features” are those which are used to sort connection records by destination host.For Ex. some probing attacks scan the hosts using a much larger time interval than 2 seconds(say twice a minute).These features are constructed using a window of 100 connections to a particular host rather than connections made in a time interval.
  4. ”Content Features” are used to detect unstructured data portions of packets.These features look for suspicious behavior in the data portions, such as the number of failed login attempts.Unlike DOS and probing attacks no sequential patterns appear in records of R2L and U2R attacks.These attacks are embedded in the data portions of packets, and normally involve only a single connection.
- Some data features have Discrete values(Ex. flag, num_root, su_attempted) and some have Continuous values(Ex. Duration, num_failed_logins).
- There are 4898431 such records which we would use for Training data.
- There is a labels associated with each connection record 1 to 23 depending on whether System it is ’Normal’ connection or one of the 22 ’Attack’ Connections.

KDD Dataset:-

Dataset contains raw training data which is about four gigabytes of compressed binary TCP dump data from seven weeks of network traffic, processed into about five million connection records. Similarly, the two weeks of test data yielded around two million connection records. Each Connection is a sequence of TCP Packets starting and ending at some well defined times.In between data flows to and from a pair of IP addresses.Each connection is labeled as either Normal or a specific attack class from 22 attack types. These Attack could be broadly classified into four categories as follows:-

- DOS : Denial of Service, Ex. Syn-Flag Flooding.
- R2L : unauthorized access from remote machine, Ex.Guessing Password.
- U2R : unauthorized access to local superuser (root) privileges, Ex. "buffer overflow" attacks;
- Probing : surveillance and other probing, e.g., port scanning

Data Analysis:-

Given data is prepared and managed by MIT Lincoln Labs to survey and evaluate research in intrusion detection. We are using whole dataset after duplicate removal. The Main Aim is being able to predict normal and different type of attack scenarios in real-time by seeing the fields of TCP packets and data portion.

- Some Data-features are protocol names(text).
- Classes are imbalanced i.e. not of equal size.If we simply train our model for such data, model would be biased for a particular class. To resolve this problem we could use various methods.

1. Upsampling :- A strategy to handle imbalanced classes by repeatedly sample with replacement from minority class to make it of equal size as the majority class.
2. Downsampling :- we creates a balanced dataset by matching the number of samples in the minority class with a random sample from the majority class.
3. Penalised Models :- Penalized classification imposes an additional cost on the model for making classification mistakes on the minority class during training. These penalties can bias the model to pay more attention to the minority class.
4. class_weights :- using parameter *class_weight="balanced"* as a parameter we can give same weight to each class i.e. entries of minority class would have higher weight and those of Majority class would have lesser weights.

Upsampling each class upto points in normal class increases data without giving any additional benefit over downsampling. We wanted to train our model on whole data that's why we haven't reduced data size.

- We have numerous classifiers namely Gaussian Naive Bayes(GNB), Bernoulli Naive Bayes(BNB), Multi Layer Perceptron, Random forest and many more. We can choose any one which suits our data best.

Model Selection :-

There are several models and each of them have there own pros and cons. We incorporated many of them in our code so we can find best model.

- Before training the model, we are removing duplicate examples from the dataset.
- We used stratified split which takes a fixed percentage of data points from each class while extracting test points from data. This would maintain similar ratio of data points in training and validation data.
- We are using Leave-One-Out Cross-validation(LOOC) on 50% of entire dataset.

1.Random Forest Classifier Model:-
It is an ensemble algorithm. It combines one or more than one algorithm of same or different kinds.It creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object.The subsets in different decision trees created may overlap.

2.SVM with Linear Kernel:-
It is highly efficient in dealing with extra large data sets.It is an ideal model for solving multiclass classification problems.Since Linear kernel used it would learn only linear boundaries.

3.Bernoulli Naive Bayes Classifier:-
It is efficient classifier and suitable for discrete data.It is designed for Binary/Boolean features.

There are some other models as well which we have tried over KDD dataset.
For Detailed result(including precision, F1 score, Confusion Matrix) check the text file for corresponding model uploaded with deliverable code on Canvas.

Selected Model	Validation Accuracy	Test Accuracy
Random Forest	99.97%	96.40%
SVM(Linear Kernel)	98.22%	95.79%
Extra Tree	99.93%	95.11%
Gaussian Naive Bayes	41.87%	45.51%
Decision Tree	99.97%	95.99%
Bernoulli Naive Bayes	93.65%	77.13%
LDA	98.40%	95.32%
5-Nearest Neighbours	99.94%	95.82%
Multilayer Percetron	99.94%	95.98%
Extra Trees	99.97%	95.96%

Conclusion:-

There are some other classifiers which we have used in our code some have better performance over others on this data.

From all models that we used in our code Decision Tree, LDA and Linear SVM are the best options for the reasons which are as follows.

- Decision Tree is very fast at testing time. Once trained, it takes only few comparison before predicting label.
- It does not get affected by unbalanced dataset.
- Linear SVM, LDA and Decision Tree models are less biased to most abundant class.

We have not trained Quadratic Discriminant Analysis(QDA)because there is only 1 example of 12^{th} class and covariance matrix is not defined in that case. We could have also trained two models 1^{st} model would check whether a point belongs to Normal Or Attack class. If the point belongs to Attack class then we could further use a multiclass classifier to decide which attack class it actually belongs to among 22 attack classes.But, we choose to do it with a single multiclass classifier model only.