SACRAMENTO STATE
COLLEGE OF ENGINEERING & COMPUTER SCIENCE

CSC 215-01
Artificial Intelligence

Mini-Project 3
Network Intrusion Detection

By

Mansi Patel
Sac State ID - 219760009

Megha Mathpal
Sac State ID – 219695880

4 pm, Monday, March 11, 2019

***Abstract*— This project report is about the network intrusion detection system that is a predictive model capable of distinguishing between bad connections called intrusions or attacks and good normal connections.**

I. PROBLEM STATEMENT

The project aims to build a network detection system to predict the type of connection been formed in the network i.e. either good or bad. The goal is to implement this system by different models like logistic regression, nearest neighbor, CNN, SVM, GNB and fully-connected neural network. We will compare the accuracy, F1-score, recall, precision of all models to find the best model of all.

II. METHODOLOGY

A. *Understanding the data*

- First, we collect all the data that we need to build such system. We got our dataset from https://drive.google.com/open?id=1J3Fgo1DCwuQpljJo8Yspe_PQad7oJb52.
- This database contains a wide variety of intrusions simulated in a military network environment. We consider understanding this data by analyzing it.
- The data has 494020 rows and 42 columns in total that has various fields like protocol type, duration of connection, outcome, service type etc. that gives us the details of connection.
- Each connection is labeled either normal or as an attack which exactly specifies which attack. The data has total 22 types of attack. All these attacks fall in four major categories DOS, R2L, U2R and probing.

B. *Data Pre processing and Cleaning*

- As this is a large dataset data cleaning and preprocessing for noisy data is needed. First, we check for null values in our data. There were no null values found in our data.
- We remove missing values in our data by dropping that row. There were no missing values found in our data.
- Redundant records in the train set will cause learning algorithms to be biased towards the more frequent records so, we remove all duplicate records by doing so our data size reduced to 145585 rows.
- Label encoded and found the good and bad connections by creating a new column connection_type that has value ‘0’ for normal connection and ‘1’ for attack. By doing so we get 878731 normal connection and 57754 connections with attack. We drop the outcome and connection type column as we want our model to predict the connection type.
- We perform one hot encoding on protocal_type, services and flag columns as it is a categorical feature.
- We perform normalization by z-score on the all numeric features. Later we check if any column has ‘0’ values in it if so, we drop it from our data frame.

C. *Classification Models*

- We model this problem as a binary classification problem to do the prediction.
- We split the test and train data. We implemented various models to detect the connection type. Those are
  - Logistic Regression (LR)
  - Nearest Neighbor (KNN)
  - Gaussian Naive Bayes (GNB)
  - Support Vector Machine (SVM)
  - Fully-Connected Neural Networks
  - (FCNN)
  - Convolutional Neural Networks (CNN)
- Compared the accuracy, recall, precision and F1-score for ALL the models. Showed confusion matrix and the ROC curve of each model.
- Used Early Stopping and Model Checkpoint when training neural networks using TensorFlow.
- We did hyper parameters tuning of activation, layer and neuron count, optimizer, kernel number and size when training neural networks using TensorFlow

III. EXPERIMENTAL RESULTS AND ANALYSIS

- Table I has the comparison of all models implemented by tenserflow

Model	F1score	Recall	Precision	Accuracy
LR	0.9922	0.9922	0.9922	0.9922
KNN	**0.9990**	0.9990	0.9990	0.9990
SVM	0.9969	0.9969	0.9969	0.9969
GNB	0.9159	0.9178	0.9261	0.9178
FCNN	0.9986	0.9986	0.9986	0.9979
CNN	0.9985	0.9985	0.9985	0.9973

Table I. comparison of model scores with Tenserflow

Optimizer	Activation and layers	Number of neuron count	F1 score
Adam	(sigmoid, sigmoid, sigmoid)	(20,10,5)	0.9981
Adam	(relu,relu)	(25,20)	**0.9988**
Adam	(tanh,tanh, tanh,tanh)	(30,20,15,5)	0.9985
Adam	(sigmoid, tanh, relu)	(35,20,15)	0.9983

Table II. FCNN different optimizers, layers and neuron count comparison

- We had a loop in our code that code has 4 optimizers ‘SGD’, 'Adagrad', 'Adam', 'Adamax'.
- In table II contains the best optimizer’s F1 score after comparing it with all four optimizers.
- Table II reflects that ‘Adam’ optimizer is performing best as compared to the other optimizers in all cases.
- Table III shows different combinations used in CNN

Optimizer	Kernel size	Kernel number	**Activation and layers**	F1 score
Adam	(1,3) (1,3)	64 128	(relu,relu)	**0.9983**
SGD	(1,1) (1,1)	58 150	(sigmoid, sigmoid)	0.4533
Adam	(1,5)	50	(tanh)	0.9973

Table III. CNN different combinations

IV. TASK DIVISION

Mansi Patel

- Data Pre-processing
- Data cleaning
- Feature Selection
- Data ready for every model processing
- Additional features

Megha Mathpal

- Classification Model
- Parameter Tuning
- Multi-class classification
- Model comparison
- Additional features

V. PROJECT REFLECTION

- Learned to use the libraries of keras classifier with CNN model, l1-l2 regularization and feature importance analysis
- The real time big data consisting a large number of the duplicate data or noisy data so data cleaning and preprocessing needed.
- Learned to handle the big data with such large number of the features and how to extract the important features by feature selection methods.
- CNN
  - Input size - 4D array
  - Output size - 1D array
- Models of the Neural Network and CNN using Tensorflow needs output to be one hot encoded 1D array while LR, KNN, SVM & Naive Bayes.
- CNN takes a lot of time to run comparatively.
- Observed that the accuracy of SGD is comparatively less than Adam. And overuse of L1 and L2 regularization for CNN results very low F1 score approximately 0.45.
- Learned to use the feature selection modules: Pearson’s correlation coefficient, RFE, Select Percentile, f_classif, standard scaler, Tree classifier, Random Tree Classifier.
- Features could be evaluated by the coefficient values and some models like SVM, KNN does not support coefficient parameter to evaluate and perform RFE.

- Learned how to visualize how probability each label has been predicted by plotting the 'Probability Calibration Curve'.
- The parameter tuning can enhance the model error for instance n_neighbors for KNN and C for SVM.
- Plotted graph for best K values and C values for the KNN and SVM models
- The problem statement could be subcategorized for the attack types, the same problem could be multi-class classification.
- Sub categories of the attacks for the multi-class classification

DoS = ['back', 'land', 'neptune', 'pod', 'smurf', 'teardrop.]
Probe = ['ipsweep', 'nmap', 'postsweep','satan']
U2R = ['buffer_overflow', 'loadmodule', 'perl', 'rootkit']
R2L = ['ftp_write', 'guess_passwd', 'imap', 'multihop', 'phf', 'spy', 'warezclient', 'warezmaster']

- The multi-class classification f1-scores observed a little bit more than the binary classification comparatively.

VI. EXTRA FEATURES

A. *Importance feature selection*

- Feature selection is the process of selecting a subset of relevant features for use in model construction. We implemented Pearson Correlation Coefficient method for feature selection.
- Pearson correlation measures the linear association between continuous variables. In other words, this coefficient quantifies the degree to which a relationship between two variables can be described by a line. Corr() Computes pairwise correlation of columns, excluding NA/null values.
- We took features above 0.029323 that is the mean value related to connection type. By doing so we got 68 features in total
- Table IV shows the F1 score, Recall, precision for all models with feature selection.

Model	F1score	Recall	Precision
LR	0.9836	0.9837	0.9838
KNN	0.9937	0.9937	0.9937
SVM	0.9912	0.9912	0.9912
GNB	0.9158	0.9178	0.9265
FCNN	0.9937	0.9937	0.9937
CNN	**0.9943**	0.9943	0.9943

Table IV. comparison of model scores with Pearson Correlation Coefficient method

B. *Feature selection within each model*

We tried to apply different feature selection on every model separately to see if it boosts up our accuracy. The table V has column F1 score before that shows the F1 score of Pearson Correlation Coefficient method and the F1 score after column has the scores obtained by different feature selection method as explained below:

Logistic Regression

- Further applied RFE to experiment with results got.
- Applied on top 40 features and the f1 score was approximately same.

KNN

- Got the best n_neighbors=4 value by plotting a graph for model error given a certain range
- Further experimented standard scaler but did not work also increased the number of misclassified results
- Fig1 shows the best K value graph for this.

SVM

- Got the best value of C=14 by plotting a graph for model error given a certain range
- It decreased the number of misclassified results
- Fig2 shows the best C value graph for this.

Naive Bayes

- performed comparatively less accurate than other models
- performed the feature selection by extra tree classifier and got 56 features
- But performance was a bit less accurate

Probability Calibration Curve

- Plotted a 'Probability Calibration Curve' to see with what probability the label is predicted and to compare it with sigmoid and isotonic for KNN, SVM and Naive Bayes with the mean error values for each model, the good result should be aligned with (nearby) diagonal.
- KNN gives the best prediction without any calibration while SVM and Naive Bayes calibrates well with sigmoid.

Fully connected Neural Network

- Performed all combinations with 4 different optimizers Adam, SGD, Adamax, Adagrad
- The best model was with adam optimizer and relu activation.
- Applied further feature selection with pertubation_rank method and eliminated the last 25% less ranked features got 50 features
- Accuracy was almost same

CNN

- Tried different combinations with different optimizers, activations and kernel number and size.
- The best model was obtained with adam optimizer and relu activation with kernel size (1,1) and number used is 64 and 128.
- Applied l1 and l2 regularization on it further.
- The accuracy obtained was the same.

Model	F1score before	F1score after
LR	0.9836	0.9829
KNN	0.9937	0.9939
SVM	0.9912	0.9921
GNB	0.9158	0.9111
FCNN	0.9937	0.9934
CNN	0.9943	0.9941

Table IV. comparison of model scores

C. Multi-class classification

- Data cleaning and preprocessing
- One hot encoding for categorical features and normalized the numeric features
- The above performed binary classification converted into multi-class classification by the attack types and safe (not an attack).
- There are 4 types of the attack mentioned in this project which are 'Denial of Service', 'Probing', 'U2R', 'R2L' which are label encoded with 1,2,3 and 4 respectively while safe state is labeled as 0.
- Applied feature selection on it with Select Percentile given f_classif parameter (ANOVA F-value between label/feature for classification tasks) with 50% percentile which selects highest 50% features from the given dataset and gave 58 such features.
- Performed models on the prepared dataset and the results listed in the table VI.

Model	F1score	Recall	Precision
LR	0.9911	0.9911	0.9911
KNN	**0.9987**	0.9987	0.9987
SVM	0.9976	0.9976	0.9976
GNB	0.9170	0.8777	0.9718
FCNN	0.9980	0.9981	0.9980
CNN	0.9980	0.9980	0.9981

Table V. comparison of model scores with multiclass classification

- The comparison of the binary classification and the multi-class classification shown in the table-VI.

Model	Binary F1score	Multi-class F1score
LR	0.9829	0.9911
KNN	0.9939	0.9987
SVM	0.9921	0.9976
GNB	0.9111	0.9170
FCNN	0.9934	0.9980
CNN	0.9941	0.9980

Table VI. comparison of binary classification & multiclass classification

D. Visualizing for the best model parameters

- Fig 1 shows error rate for K values 0 to 19 having

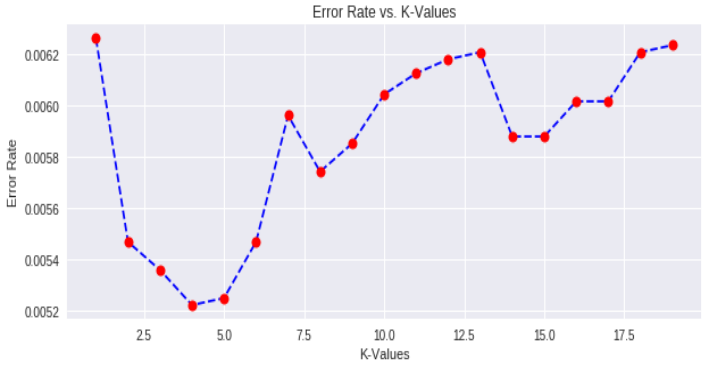

Fig 1. Best K value

- Fig 2 shows error rate for K values 0 to 14 having

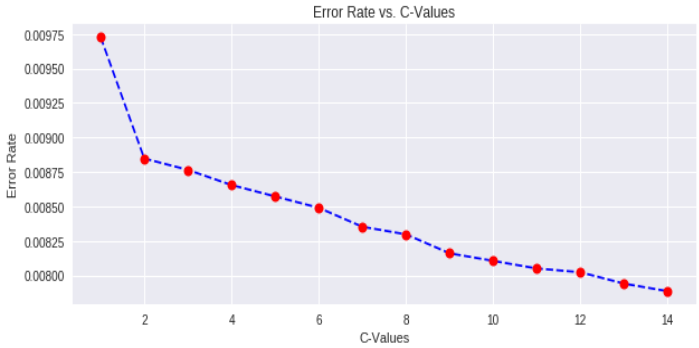

Fig 2. Best C value

VII. COCNLUSION

- This project work deal with comparing the result of different classification model which includes SVM, kNN, Logistic Regression, Gaussian Naïve Bayes, Neural Network and Convolution Neural Network.
- CNN could be work on any dataset as long as the data could be represented as 4D input data shape as image.
- While dealing with the big data the feature selection plays a vital role thus understanding the dataset field and preprocess it in correct format is very important.

REFERENCES

1. HTTPS://SCIKIT-LEARN.ORG/STABLE/MODULES/GENERATED/SKLEARN.FEATURE_SELECTION.SELECTPERCENTILE.HTML
2. https://www.svds.com/classifiers2/
3. https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/