PROJECT REPORT

ON

Cybersecurity threats detection of internet of things using deep learning

(CSE III Semester Mini Project)

2020-2021

Submitted to:

Mr Ashwini Kumar

(CC-CE_SPL 1-III Sem)

Guided by:

Dr. Sachin Sharma

Submitted by:

NEELA PRANAY

Roll No:2017345

CE_SPL 1-III SEM

Session-2021-2022

CERTIFICATE

Certified that Mr. NEELA PRANAY (Roll NO.- 2017345) has developed mini project on “Cybersecurity threats detection of internet of things using deep learning” for the CSE III Semester Mini Project in Graphic Era University, Dehradun. The project carried out by Students is their own work as best of my knowledge.

Date:25-02-2022

Project Co-ordinator

Mr. Ashwini Kumar

CE SPL 1-III-Sem

(CSE Department)

GEU Dehradun

Project Guide

MR. Sachin Sharma

Resource Person

(CSE Department)

GEU Dehradun

ACKNOWLEDGMENT

We would like to express our gratitude to The Almighty Sachin Sharma sir, the most Beneficent and the most Merciful, for completion of project.

We wish to thank our parents for their continuing support and encouragement. We also wish to thank them for providing us with the opportunity to reach this far in our studies.

We would like to thank particularly our project Co-ordinator Mr. Hemant Pokhariya sir and our Project Guide Mr. Sachin Sharma sir for his patience, support and encouragement throughout the completion of this project and having faith in us.

At last, but not the least We greatly indebted to all other persons who directly or indirectly helped us during this work.

MR. NEELA PRANAY

Roll No.- 2017345

CE_SPL 1 -III-Sem

Session: 2021-2022

GEU, Dehradun

INDEX

I.INTRODUCTION:

1.1 PROBLEM STATEMENT:

Given a sequence of transactions, can a classifier be used to model the time series inherent in the sequence to such an extent that deviations in card holder shopping behaviour can be detected regardless of the skewness and noise inherent in the data? In addition, our classifier must exhibit both a high probability of detection and a low false alarm rate during generalisation; otherwise, it will be practically useless. Since we are ultimately dealing with a dynamic problem here, the question also arises whether a dynamic neural network machine learning technique will outperform a static one. It can also be noted that the retraining of static networks to deal with changing shopping behaviour can have the effect that certain fraud patterns, that have not occurred in quite some time, cannot be detected by the retrained network when they do reoccur. A network that can learn when to forget information can be beneficial in such cases.

Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically. Every year due to fraud Billions of amounts lost. To analyse the fraud there is lack of research. Many machine learning algorithms are implemented to detect real world credit card fraud. ANN and hybrid algorithms are applied.

1.2 ABSTARCT:

In recent years credit card fraud has become one of the growing problems. It is vital that credit card companies are able to identify fraudulent credit card transactions, so that customers are not charged for the items that they didn't purchase. The reputation of companies will heavily damage and endangered among the customers due to fraud in financial transactions The fraud detection techniques were increasing to improve accuracy to identify the fraudulent transactions. This project intends to build an unsupervised fraud detection method using autoencoder. An Autoencoder with four hidden layers which has been trained and tested with a dataset containing an European cardholder transactions that occurred in two days with 284,807 transactions from September 2013.

II. OVERALL DESCRIPTION:

2.1 MY MOTIVATION:

"Card companies continue to increase the effectiveness and sophistication of customer-profiling neural network systems that can identify at a very early-stage unusual spending patterns and potentially fraudulent transactions".

There are several different factors that make card fraud research worthwhile. The most obvious advantage of having a proper fraud detection system in place is the restriction and control of potential monetary loss due to fraudulent activity. Annually, card issuers suffer huge financial losses due to card fraud and, consequently, large sums of money can be saved if successful and effective fraud detection techniques are applied. While card fraud losses against total turnover have actually declined in the past decade or so - undoubtedly due to card issuers actively fighting fraud - the total monetary loss due to card fraud has increased sharply during the same time due to an increase in the total number of cards issued and the resulting increase in card usage. Figure 1-1 shows that, during 2004 alone, an estimated £504.8 million were lost on UK-issued cards due to fraudulent activity, a staggering increase of 20% over the Chapter 1 Introduction 2 preceding year. In addition to this, newer fraud methods are emerging as fraud detection increases and chip cards become more widely used. These include money laundering on card transactions, identity or application fraud to obtain cards and hacking of card numbers from card processors.

2.2 TOOLS USED:

We use the following libraries and frameworks in credit card fraud detection project.

- Python – 3.x
- Numpy – 1.19.2
- Scikit-learn – 0.24.1
- Matplotlib – 3.3.4
- Imblearn – 0.8.0
- Collections, Itertools

2.3 METHODOLOGY:

DATASETS:

The dataset for this research work is obtained from Kaggle, and it was generated using Sparkov Data Generation, a GitHub tool created by Brandon Harris. The dataset is a simulated credit card transaction containing legitimate and fraudulent transactions. It covers the credit card of 1000 customers doing transactions with a pool of 800 merchants.

https://www.kaggle.com/kartik2112/fraud-detection

DATA PREPROCESSING:

Preprocessing data is required before implementing a machine learning algorithm, considering various models produce diverse specifications to the predictors, and data training can affect predictive production. Data preprocessing purposes are to clean and prepare the data to a spot that comprises more concise prejudice, checking for missing values, and more variation. Data contains both numerical and categorical, which means encoding the categorical data is necessary before using them for modeling. Outlier detection and removal was performed. We have the independent variables in the same range by performing feature scaling. To reduce feature skewness, a box-cox transformation was carried out. Resampling method such as under sampling and oversampling was performed on the imbalanced original dataset to avoid any form of bias and overfitting in our training model. We have adopted Python data manipulation library pandas and machine learning library sci-kit learn to achieve these preprocessing responsibilities.

Credit Card Fraud Detection Project

In [17]:
```
# Import the necessary modules

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
import itertools

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, precision_score, confusion_matrix, recall_score, f1_score
```

In [2]:
```
# Load the csv file

dataframe = pd.read_csv("./Desktop/DataFlair/credit_card_fraud_detection/creditcard.csv")
dataframe.head()
```

Out[2]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.1285
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.1671
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.3276
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.6473
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.2060

5 rows × 31 columns

III.MACHINE LEARNING:

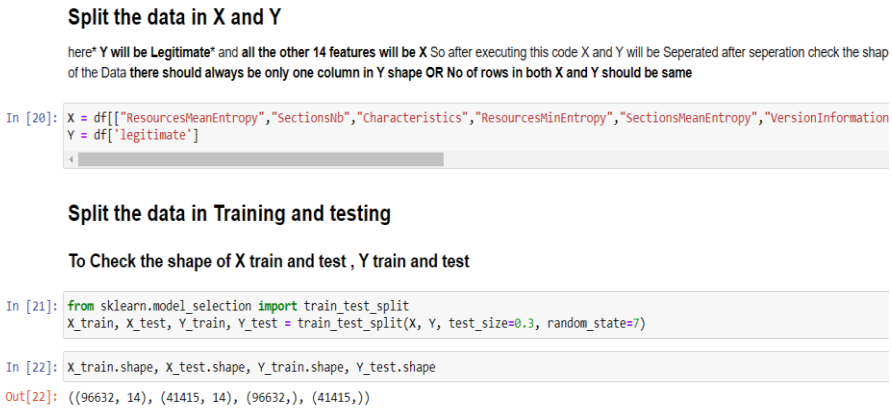
3.1 MODEL BUILDING:

Train/split: One significant aspect of all machine learning models is to determine their delicacy. Now, in order to determine their delicacy, one can train the model using the given dataset and then prognosticate the response values for the same dataset using that model and hence. Find the delicacy of the model. A better option is to resolve our data into two parts: first one for training our machine learning model and second one is for testing our model

- We resolve the dataset into two pieces a training set and testing set
- We Can Train the model on the training set
- Test the model on the testing set and estimate how well our model did.

Advantages of train/test split:

- The Model can be trained and tested on disparate data than the one used for training.
- Response values are known for the test dataset; hence prognostications can be evaluated.
- Testing delicacy is better estimate than training delicacy of out-of-sample performance.



IV.ALGORITHMS:

Machine Learning consists of algorithms that can automate logical/analytical model structure. Using algorithms that repeatedly learn from data, machine learning models facilitate computers to find hidden insights from Big Data being explicitly programmed where to look.

4.1 RANDOM FOREST MODEL:

Random forest is a supervised learning algorithm. The “forest” it builds, is an ensemble of decision trees, generally trained with the “bagging” system. The prevalent idea of bagging method is that a combination of literacy models increases the overall result.

Put simply random forest builds multiple decision tress and merges than together to get a more accurate and stable prediction.

Apply Random Forest Classifier

```
In [35]: from sklearn.ensemble import RandomForestClassifier
         rcf = RandomForestClassifier(n_estimators=50,random_state=7)
```

fit the model in X_train and Y_train

```
In [36]: rcf.fit(X_train,Y_train)

Out[36]: RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                max_depth=None, max_features='auto', max_leaf_nodes=None,
                                min_impurity_decrease=0.0, min_impurity_split=None,
                                min_samples_leaf=1, min_samples_split=2,
                                min_weight_fraction_leaf=0.0, n_estimators=50,
                                n_jobs=None, oob_score=False, random_state=7, verbose=0,
                                warm_start=False)
```

Predict the Accuracy and Score

```
In [37]: accuracy2 = (rcf.score(X_test, Y_test))*100
         accuracy2

Out[37]: 99.35772063262102
```

V.FUTURE SCOPE:

Based on the conclusion we have above, we can further improve our approach in this study. Future work can be done on this topic: The current study agreed with the result of the oversampling dataset, which duplicates. Instead of using this oversampling method that duplicates, we can use another sampling technique called interpolation, where redundant observations are not added to our dataset.

VI. CONCLUSION:

The technology change influenced several improvements. We are talking about online transactions done through credit cards, which leads to credit card frauds, and this study is about improving machine learning algorithms for fraud detection. In this study, we put forth fraud detection methods based on supervised learning such as Random Forest, Decision Tree, Boost, and logistic regression, unsupervised learning such as K-means clusters, and one deep learning algorithm known as Autoencoder Neural Network. We compared all the algorithms with different datasets by first using the original dataset itself; we then use resampling

techniques such as undersampling and oversampling because our dataset is highly imbalanced. Finally, we concluded that Random Forest would be the perfect fit for our model. It can be inferred that oversampling works better because the smaller number of observations helps in training our model efficiently.

VII. APPENDIX:

Code Link:

https://colab.research.google.com/drive/1gP7HkUQcdcgFeD0X9Fq6bA_Th3zr6IrZ?usp=sharing

VIII. REFERENCES:

[1] Aditya Mishra. (2018). Metrics to Evaluate your Machine Learning Algorithm. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learningalgorithm-f10ba6e38234

[2] Arden Dertat. (2017). Applied Deep Learning – autoencoder. https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders1c083af4d798

[3] Azhan, Mohd. (2020). Credit Card Fraud Detection using Machine Learning and Deep Learning Techniques. 10.1109/ICISS49785.2020.931600