BITCOIN HEIST RANSOMWARE ATTACK PREDICTION USING DATA SCIENCE PROCESS

Mrs. ASHWINI J.P.(M.E.) , ASSISTANT PROFESSOR,
Department of Computer Science and Engineering
Mr. K. SANJAY B.E, Student of Computer Science Engineering
Mr. M. AVINASH, B.E, Student of Computer science and Engineering
St. Joseph College of Engineering, Sriperumbudur, Chennai.

Abstract:

Ransomware attacks are emerging as a major source of malware intrusion in recent times. While so far ransomware has affected general-purpose adequately resourceful computing systems. Many ransomware prediction techniques are proposed but there is a need for more suitable ransomware prediction techniques for machine learning techniques. This paper presents an attack of ransomware prediction technique that uses for extracting information features in Artificial Intelligence and Machine Learning algorithms for predicting ransomware attacks. The application of the data science process is applied for getting a better model for predicting the outcome. Variable identification and data understanding is the main process of building a successful model. Different machine learning algorithms are applied to the pre-processed data and the accuracy is compared to see which algorithm performed better other performance metrics like precision, recall, f1-score are also taken in consideration for evaluating the model. The machine learning model is used to predict the ransomware attack outcome..

Introduction:

Cryptocurrencies, such as Bitcoin, are a form of digital currency designed to work outside of the traditional banking ecosystem. Cryptocurrencies are decentralized currencies that use blockchain technology to record transactions. Cryptocurrency transactions, aka the buying and selling of digital currency, are typically handled using a crypto-exchange platform. These transactions often involve large sums of cryptocurrency, typically anonymized utilizing the blockchain, hence attracting cybercriminals. Like any system, cryptocurrency platforms and exchange mechanisms are vulnerable to cyberattacks.

INTRODUCTION

Data Science

The term "data science" has been traced back to 1974, when Peter Naur proposed it as an alternative name for computer science. In 1996, the International Federation of Classification Societies became the first conference to specifically feature data science as a topic. However, the definition was still in flux.

Literature Survey

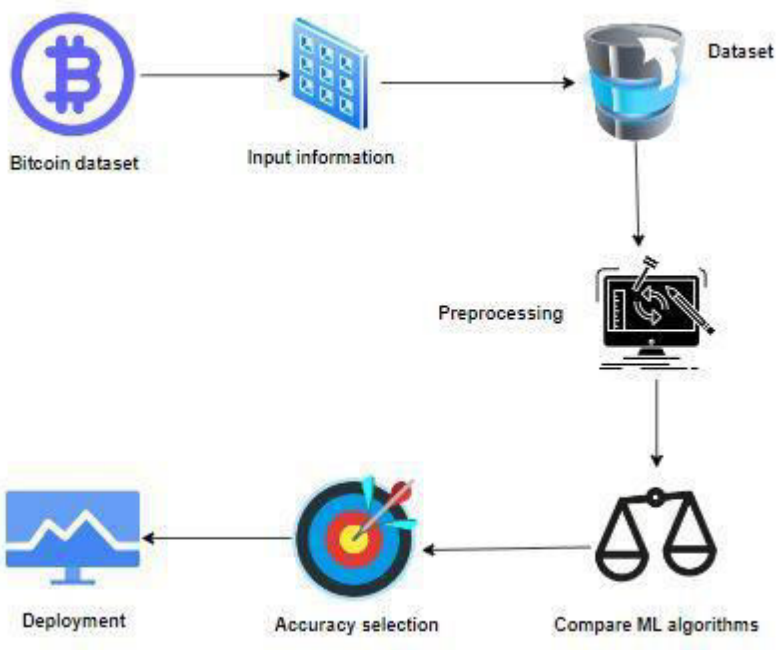
Bitcoin Heist : Topological Data Analysis for Ransomware Detection on bitcoin Blockchain by Yulia R. Gel, Murat Kantarcioglu in 2019

Ransomware is a type of malware that infects a victim's data and resources, and demands ransom to release them. In two main types, ransomware can lock access to resources or encrypt their content. In addition to computer systems, ransomware can also infect IoT and mobile devices [23]. Ransomware can be delivered via email attachments or web based vulnerabilities. More recently, ransomware have been delivered via mass exploits. For example, CryptoLocker used Gameover ZeuS botnet to spread through spam emails. Once the ransomware is installed, it communicates with a command and control center. Although earlier ransomware used hard-coded IPs and domain names, newer variants may use anonymity networks, such as TOR, to reach a hidden command and control server Once resources are locked or encrypted, the ransomware displays a message that asks a certain amount of bitcoins to be sent to a bitcoin address.

System Design

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming (including by meta-programming and meta-objects (magic methods). other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.



The best way to get started using Python for machine learning is to complete a project.

- It will force you to install and start the Python interpreter (at the very least).
- It will give you a bird's eye view of how to step through a small project.
- It will give you confidence, maybe to go on to your own small projects.

.

The result of the machine learning model is the prediction of stroke risk for a given patient. This information can be used by clinicians to make more informed decisions about patient care, such as recommending lifestyle changes or prescribing medication to reduce the risk of stroke.

IMPLEMENTATION

Coding:

Module – 1

Pre-Processing

```
import pandas as p
import numpy as n
```

In [ ]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [ ]:

```
data = p.read_csv('Data.csv')
```

In [ ]:

```
data.head()
```

In [ ]:

```
data.shape
```

```
df = data.dropna()
```

In [ ]:

```
df.shape
```

In [ ]:

```
df.isnull().sum()
```

In [ ]:

```
df.columns
```

In [ ]:

```
df.describe()
```

In [ ]:

```
df.length.unique()
```

In [ ]:

```
p.crosstab(df.label,df.year)
```

In [ ]:

```
p.Categorical(df['label']).describe()
```

In [ ]:

```
p.Categorical(df['year']).describe()
```

In [ ]:

```
print("Days: ", sorted(df['day'].unique()))
```

In [ ]:

```
df['length'].value_counts()
```

In [ ]:

```
df.duplicated()
```

In [ ]:
```
df.duplicated().sum()
```
In [ ]:
```
df.columns
```
In [ ]:
```
print("Minimum Income : ", df.income.min())
print("Maximum Income : ", df.income.max())
```

Module - 2
Visualization:

```
#import library packages
import pandas as p
import matplotlib.pyplot as plt
import seaborn as s
import numpy as n
```
In [ ]:
```
import warnings
warnings.filterwarnings('ignore')
```
In [ ]:
```
df = p.read_csv("Data.csv")
```
In [ ]:
```
df
```
In [ ]:
```
df.columns
```
In [ ]:
```
df[
'ye
ar'
].h
ist
(fi
gsi
ze=
(10
,4)
,
col
or=
'c'
)
plt
.xl
)
```

In [ ]:

```
#Propagation by variable
def PropByVar(df, variable):
dataframe_pie = df[variable].value_counts()
ax = dataframe_pie.plot.pie(figsize=(8,8), autopct='%1.2f%%', fontsize = 12)
ax.set_title(variable + ' \n', fontsize = 15)
return n.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'label')
```

In [ ]:

```
#Propagation by variable
def PropByVar(df, variable):
dataframe_pie = df[variable].value_counts()
ax = dataframe_pie.plot.pie(figsize=(8,8), autopct='%1.2f%%', fontsize = 12)
ax.set_title(variable + ' \n', fontsize = 15)
return n.round(dataframe_pie/df.shape[0]*100,2)
PropByVar(df, 'year')
```

Voting Classifier

In [ ]:

```
import RandomForestClassifier
fro
from sklearn.linear_model import LogisticRegression
m
from sklearn.ensemble import VotingClassifier
xgb
from sklearn.metrics import confusion_matrix,
oos
classification_report, accuracy_score, plot_confusion_matrix
t
Training Process
imp
ort
XGB
Cla
ssi
fie
r
fro
m
skl
ear
n.e
nse
mbl
e
```

In [ ]:

```
xg = XGBClassifier()
rf = RandomForestClassifier()
lr = LogisticRegression()
```

In [ ]:

```
vc = VotingClassifier(estimators=[('XGBoost', xg), ('RandomForestClassifier',
rf), ('LogisticRegression', lr)], voting='hard')
```

In [ ]:

```
vc.fit(X_train,y_train)
pred_vc = vc.predict(X_test)
```

Getting Accuracy

In [ ]:

```
accuracy = accuracy_score(y_test,pred_vc)
print('Accuracy of Voting Classifier is: ',accuracy*100)
```

Finding Clasiification Report

In [ ]:

```
cr = classification_report(y_test,pred_vc)


print('Classification report\n----------------------\n',cr)
```

Finding Confusion Matrix

In [ ]:

```
cm = confusion_matrix(y_test,pred_vc)
print('Confusion matrix\n----------------\n',cm)
```

In [ ]:

```
imp
ort
mat
plo
tli
```

```
b.pyplot as plt
fig, ax = plt.subplots(figsize=(7,7))
plot_confusion_matrix(vc, X_test, y_test, ax=ax)
plt.title('Confusion Matrix of Voting
Classifier\n')
plt.show()
```

In [ ]:

```
DF
=
p.D
ata
Fra
me(
)
```

```
DF["y_test"] = y_test
DF["predicted"] = pred_vc
DF.reset_index(inplace=True)
plt.figure(figsize=(20, 5))
plt.plot(DF["predicted"][:100], marker='x',
linestyle='dashed', color='red')
plt.plot(DF["y_test"][:100], marker='o', linestyle='dashed',
color='green')
plt.show()
```

```
#       LOGISTIC REGRESSION:
Mod     #import library packages
ule     import p
- 5
pandas as

cm = confusion_matrix(y_test,predicted_lr)

print('Confusion matrix\n----------------\n',cm)
```

ln [ ]:

```
import matplotlib.pyplot as plt

fig, ax = plt.subplots(figsize=(7,7))

plot_confusion_matrix(lr, X_test, y_test, ax=ax)

plt.title('Confusion Matrix of Logistic Regression\n')

plt.show()
```

ln [ ]:

```
DF = p.DataFrame()

DF["y_test"] = y_test

DF["predicted"] = predicted_lr

DF.reset_index(inplace=True)

plt.figure(figsize=(20, 5))

plt.plot(DF["predicted"][:100], marker='x', linestyle='dashed', color='red')

plt.plot(DF["y_test"][:100], marker='o', linestyle='dashed', color='green')

plt.show()
```

```
import numpy as np

from flask import Flask, request, jsonify, render_template

import pickle

import joblib

app = Flask(__name__)

model = joblib.load('xgb.pkl')

@app.route('/')

def home():
```

```
return render_template('index.html')

@app.route('/predict',methods=['POST'])

def predict():

'''

For rendering results on HTML GUI

'''

int_features = [(x) for x in request.form.values(
```

In [ ]:
```
import warnings
warnings.filterwarnings('ignore')
```
In [ ]:
```
#Load given dataset
data = p.read_csv("Data.csv")
```
In [ ]:
```
df = data.dropna()
```
In [ ]:
```
df
```
In [ ]:
```
del df['Unnamed: 0']
del df['address']
```
In [ ]:
```
df['label']=df['label'].map({'montrealCryptXXX':'Crypto','montrealCryptoLoc
```

SNAPSHOTS

Advantages:

- • We are implementing particularly on bitcoin ransomware attacks.
- • We are implementing the voting classifier.
- • Deployment can be done

Disadvantages:

1. They did not mentioning what kind of ransomware attacks they are predicting.
2. Voting Classifier is not implemented.
3. Deployment is not done.

Conclusion:

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation.

The best accuracy on public test set of higher accuracy score algorithm will be find out. The founded one is used in the application which can help to find the Bitcoin Heist randsomeware attack.

AUTHOR 1

Mrs. Ashwini M.E., is a Assistant professsor in the Department of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, Tamil Nadu. Completed Master of engineering

AUTHOR 2

Mr. K. SANJAY B.E., Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops, Seminars in Python, Machine Learning. I got placed in Reputed Companies like Q Spider .

AUTHOR 3

Mr. M. AVINASH B.E., Student of Computer Science and Engineering at St. Joseph College of Engineering, Sriperumbudur, Chennai, TamilNadu. I had attended many Workshops and Seminars in the area of Python and Machine Learning.