International Journal of Research (IJR)
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

Network Traffic Analysis Using Machine Learning

Tamtam Sunil Goud[1], Gorla Yaswanth[2], Shaik Sohail Ahammed[3] , Sirisha Kamsali[4], M. Sri Lakshmi[5]
[1,2,3] U.G. Scholar, [4]Guide Assistant Professor, [5]Head of the Department
[1,2,3,4,5] Computer Science And Engineering
[1,2,3,4,5] G. Pullaiah College Of Engineering And Technology
Email : [1]sunilgoudmjss@gmail.com , [2]vkrish783@gmail.com , [3]sohailahammed@gmail.com
[4]sirisha@gpcet.ac.in

ABSTRACT

Stood out from the past, upgrades in PC and correspondence developments have given wide and pushed changes. The utilization of new advancements gives fantastic benefits to individuals, associations, and states, nevertheless, messes face them. For example, the insurance of critical information, security of taken care of data stages, availability of data, etc. Dependent upon these issues, advanced dread based mistreatment is one of the main issues nowadays. Computerized dread, which made a ton of issues for individuals and foundations, has shown up at a level that could subvert open and public safety by various social affairs, for instance, criminal affiliations, capable individuals, and advanced activists. Thusly, Intrusion Detection Systems (IDS) have been made to avoid advanced attacks.

At the present time, learning the reinforce support vector machine (SVM) estimations were used to perceive port scope tries reliant upon the new CICIDS2017 dataset with 97.80%, and 69.79% accuracy rates achieved independently. As opposed to SVM we can present a few different calculations like the irregular timberland, CNN, and ANN where these calculations can procure exactnesses like SVM - 93.29, CNN - 63.52, Random Forest - 99.93, ANN - 99.11.

Keywords : Hacking breach, data breach, cyber threats, cyber risk analysis, breach prediction, trend analysis, time series, cyber security data analytics

INTRODUCTION

OBJECTIVE OF THE PROJECT

The utilization of new advancements give mind boggling benefits to individuals, associations, and state run administrations, nevertheless, wrecks some against them. For example, the insurance of huge information, security of taken care of data stages, availability of data, etc. Dependent upon these issues, computerized dread based persecution is one of the main issues nowadays. Computerized dread, which made a lot of issues individuals and foundations, has shown up at a level that could subvert open and country security by various get-togethers, for instance, criminal affiliation, capable individuals and advanced activists. Thusly, Intrusion Detection Systems (IDS) has been made to avoid advanced attacks

THE EXISTING SYSTEM

Blameless Bayes and Principal Component Analysis (PCA) have been utilized with the KDD99 dataset by Almansob and Lomte. Also, PCA, SVM, and KDD99 were being used by Chithik and Rabbani for IDS. In Aljawarneh et al.'s. Their evaluation and assessments were conveyed because of the NSL-KDD dataset for their IDS model. They formed judgments to

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

show that the KDD99 dataset is constantly utilized for IDS. There are 41 features in KDD99, and it was made in 1999. Subsequently, KDD99 is old and gives no information about state-of-the-art new attack types, for instance, multi-day abuses, etc. This way, we used a high level and new CICIDS2017 dataset in our examination.

PROPOSED SYSTEM

The important steps of the algorithm are given in below.
1) Normalization of every dataset.
2) Convert that dataset into the testing and training.
3) Form IDS models with the help of using RF, ANN, CNN and SVM algorithms.
4) Evaluate every model's performances

Advantages:

- Protection from malicious attacks on your network.
- Deletion and/or guaranteeing malicious elements within a preexisting network.
- Prevents users from unauthorized access to the network.
- Deny's programs from certain resources that could be infected.
- Securing confidential information

SCREENS

```
: import numpy as np
  import pandas as pd
  import matplotlib.pyplot as plt
  %matplotlib inline
```

```
: import itertools
  import seaborn as sns
  import pandas_profiling
  import statsmodels.formula.api as sm
  from statsmodels.stats.outliers_influence import variance_inflation_factor
  from patsy import dmatrices
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning: pandas.util.testing is dep
recated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

```
: from sklearn import datasets
  from sklearn.feature_selection import RFE
  import sklearn.metrics as metrics
  from sklearn.svm import SVC
  from sklearn.linear_model import LogisticRegression
  from sklearn.feature_selection import SelectKBest
  from sklearn.feature_selection import chi2, f_classif, mutual_info_classif
```

```
: train=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Train.txt',sep=',')
  test=pd.read_csv('/content/drive/My Drive/kdd/NSL_Dataset/Test.txt',sep=',')
```

Data preprocessing

```
n [6]: columns=["duration","protocol_type","service","flag","src_bytes","dst_bytes","land",
"wrong_fragment","urgent","hot","num_failed_logins","logged_in",
"num_compromised","root_shell","su_attempted","num_root","num_file_creations",
"num_shells","num_access_files","num_outbound_cmds","is_host_login",
"is_guest_login","count","srv_count","serror_rate", "srv_serror_rate",
"rerror_rate","srv_rerror_rate","same_srv_rate", "diff_srv_rate","srv_diff_host_rate","dst_host_count","dst_host_sr
"dst_host_diff_srv_rate","dst_host_same_src_port_rate",
"dst_host_srv_diff_host_rate","dst_host_serror_rate","dst_host_srv_serror_rate",
"dst_host_rerror_rate","dst_host_srv_rerror_rate","attack", "last_flag"]

n [7]: train.columns=columns
       test.columns=columns

n [8]: train.head()
```

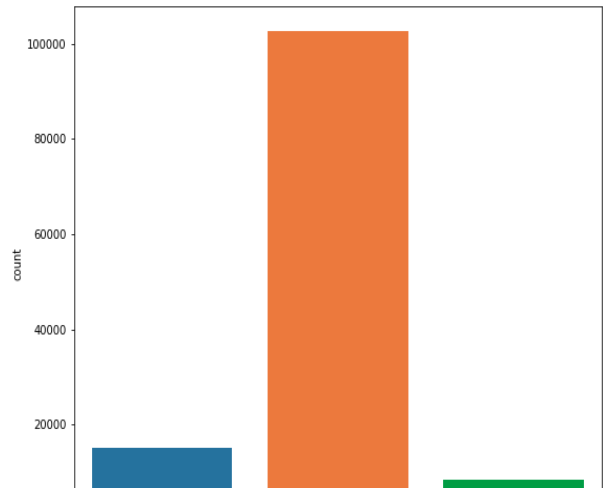
ut[8]:

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_compromised	root_s
0	0	udp	other	SF	146	0	0	0	0	0	0	0	0	
1	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	
2	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	
3	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	
4	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	

```
n [9]: test.head()
```

Data EDA

```
: # Protocol type distribution
  plt.figure(figsize=(9,8))
  sns.countplot(x="protocol_type", data=train)
  plt.show()
```



Model Building

```
train_X=train_new[cols]
train_y=train_new['attack_class']
test_X=test_new[cols]
test_y=test_new['attack_class']
```

ML Deploy

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

Logistic Regression

```
# Building Models
from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression(random_state=0,solver='lbfgs',multi_class='multinomial')
logreg.fit( train_X, train_y)
logreg.predict(train_X)   #by default, it use cut-off as 0.5
```

```
list( zip( cols, logreg.coef_[0] ) )
```

```
logreg.intercept_
```

```
logreg.score(train_X,train_y)
```

Decision Trees

```
train_X.shape
```

```
param_grid = {'max_depth': np.arange(2, 12),
              'max_features': np.arange(10,15)}
```

```
train_y.shape
```

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier, export_graphviz, export
tree = GridSearchCV(DecisionTreeClassifier(), param_grid, cv = 10,verbose=1,n_jobs=-1)
tree.fit( train_X, train_y )
```

```
tree.best_score_
```

```
tree.best_estimator_
tree.best_params_
```

```
train_pred = tree.predict(train_X)
```

```
print(metrics.classification_report(train_y, train_pred))
```

```
test_pred = tree.predict(test_X)
```

International Journal of Research (IJR)
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

Random Forest

```
from sklearn.ensemble import RandomForestClassifier
pargrid_rf = {'n_estimators': [50,60,70,80,90,100],
              'max_features': [2,3,4,5,6,7]}
```

```
from sklearn.model_selection import GridSearchCV
gscv_rf = GridSearchCV(estimator=RandomForestClassifier(),
                       param_grid=pargrid_rf,
                       cv=10,
                       verbose=True, n_jobs=-1)

gscv_results = gscv_rf.fit(train_X, train_y)
```

```
gscv_results.best_params_
```

```
gscv_rf.best_score_
```

```
radm_clf = RandomForestClassifier(oob_score=True,n_estimators=80, max_features=5, n_jobs=-1)
radm_clf.fit( train_X, train_y )
```

```
radm_test_pred = pd.DataFrame( { 'actual':  test_y,
                                 'predicted': radm_clf.predict( test_X ) } )
```

Support Vector Machine (SVM)

```
from sklearn.svm import LinearSVC
svm_clf = LinearSVC(random_state=0, tol=1e-5)
svm_clf.fit(train_X,train_y)
```

```
print(svm_clf.coef_)
print(svm_clf.intercept_)
print(svm_clf.predict(train_X))
```

```
from sklearn.svm import SVC
from sklearn.pipeline import make_pipeline

model = SVC(kernel='rbf', class_weight='balanced',gamma='scale')
```

```
model.fit(train_X,train_y)
```

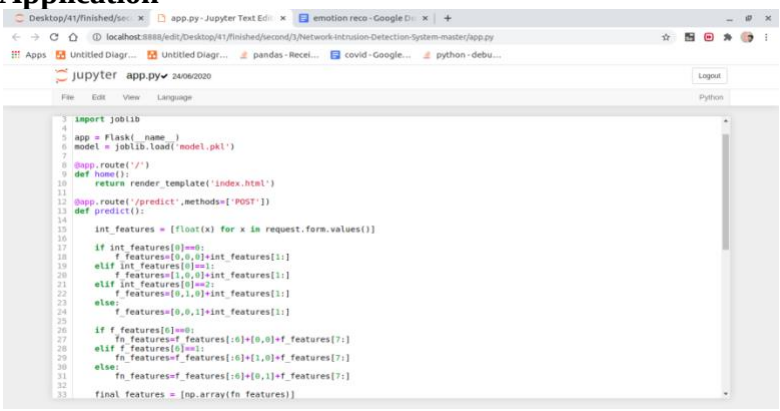
```
from sklearn.model_selection import GridSearchCV
param_grid = {'C': [1, 10],
              'gamma': [0.0001, 0.001]}
grid = GridSearchCV(model, param_grid)

grid.fit(train_X,train_y)
```

```
print(grid.best_params_)
```

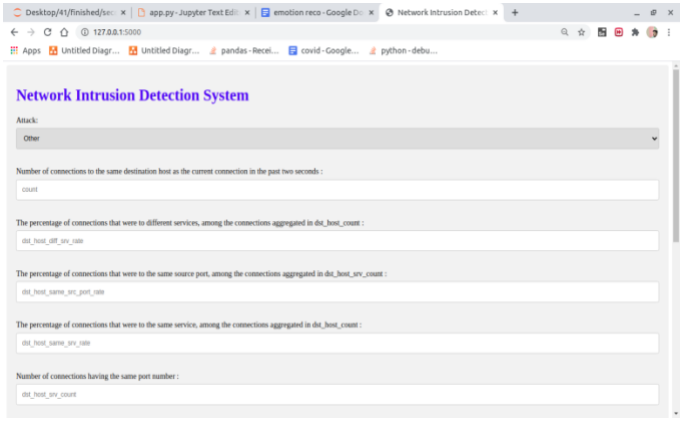
From the score accuracy we concluding the DT & RF give better accuracy and building pickle file for predicting the user input

Application

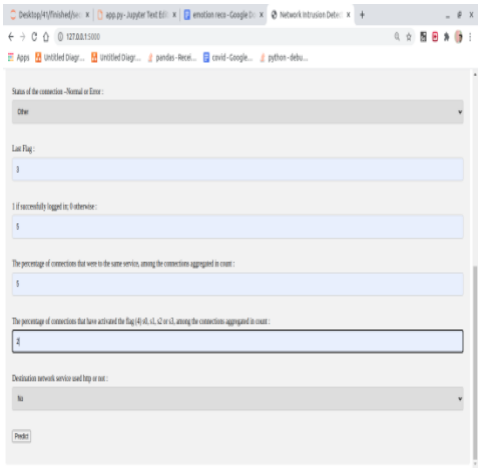


Localhost - in cmd python app.py

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022



Enter the input

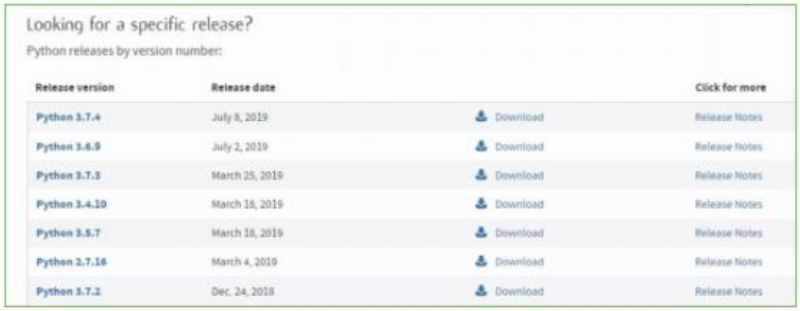


Predict attack -

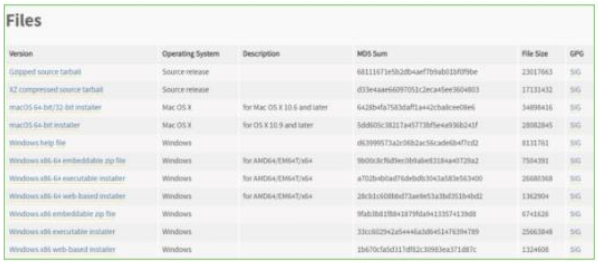
Download the Correct version into the system
Step 1: Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: https://www.python.org

Now, check for the latest and the correct version for your operating system.
Step 2: **Click on the Download Tab.**

Step 3: **You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4**

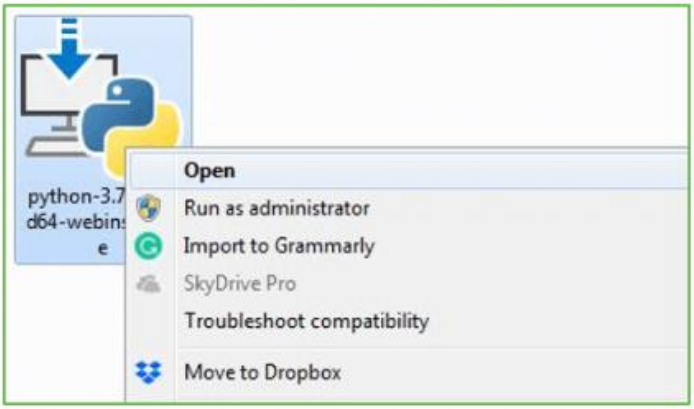


Step 4: Scroll down the page until you find the Files option.
Step 5: Here you see a different version of python along with the operating system.



e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

• To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
•To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.
Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation
Note: To know the changes or updates that are made in the version you can click on the Release Note Option.
Installation of Python
Step 1: Go to Download and Open the downloaded python version to carry out the installation process.



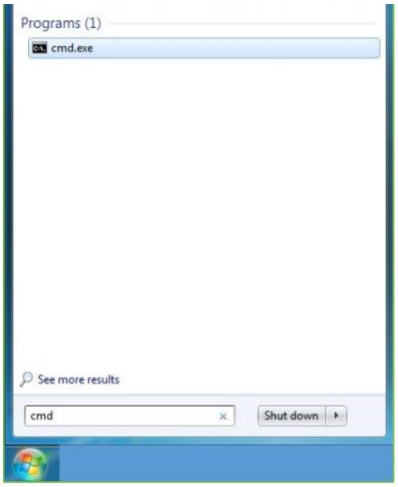
Step 2: Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.

Step 3: Click on Install NOW After the installation is successful. Click on Close.

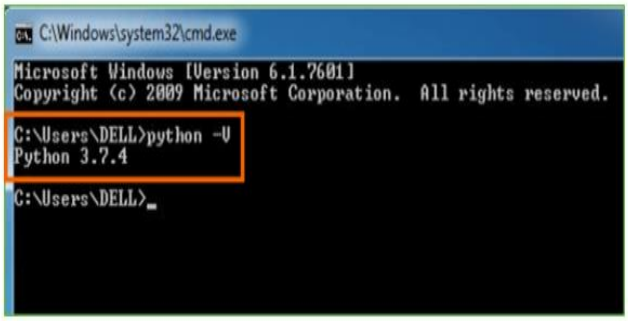
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.
Note: The installation process might take a couple of minutes.

Verify the Python Installation
Step 1: Click on Start
Step 2: In the Windows Run Command, type "cmd".

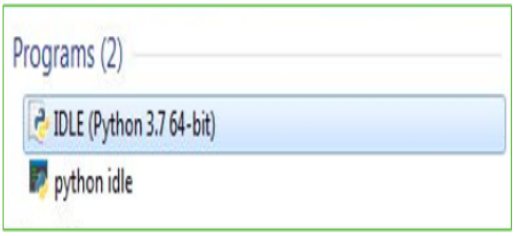


Step 3: Open the Command prompt option.
Step 4: Let us test whether the python is correctly installed. Type **python –V** and press Enter.

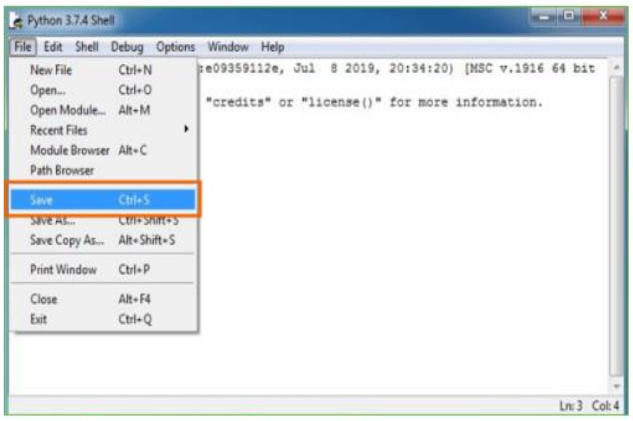


Step 5: You will get the answer as 3.7.4
Note: If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

Check how the Python IDLE works
Step 1: Click on Start
Step 2: In the Windows Run command, type "python idle".



Step 3: Click on IDLE (Python 3.7 64-bit) and launch the program
Step 4: To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



Step 5: Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.
Step 6: Now for e.g. **enter print**

Conclusion

At this moment, assessments of assist vector with machining, ANN, CNN, Random Forest, and significant learning computations subject to the ongoing CICIDS2017 dataset were presented moderately. Results show that the profound learning computation performed essentially ideal results over SVM, ANN, RF, and CNN. We will use port breadth attempts as well as other attack types with AI and significant learning estimations, apache Hadoop and shimmer developments together ward on this dataset later on. Every one of these estimation assists us with identifying a cyberattack on the organization. It occurs in the manner that when we think about lengthy back a very long time there might be such countless assaults that occurred so when these assaults are perceived, then the highlights at which esteems these assaults are going on will be put away in some datasets. So by utilizing these datasets, we will foresee regardless of whether the cyberattack is finished. These forecasts can be made by four calculations like SVM, ANN, RF, and CNN this paper assists with recognizing which calculation predicts the best precision rates, which permits anticipating that the best outcomes should decide if the digital assaults occurred or not..

e-ISSN: 2348-6848
p-ISSN: 2348-795X
Vol. 9 Issue 06
June 2022

REFERENCE

1. Chakraborty, A., J.S. Banerjee, and A. Chattopadhyay. Non-uniform quantized data fusion rule alleviating control channel overhead for cooperative spectrum sensing in cognitive radio networks. in 2017 IEEE 7th International Advance Computing Conference (IACC). 2017.IEEE.
2. Chakraborty, A., J.S. Banerjee, and A. Chattopadhyay, Non-uniform quantized data fusion rule for data rate saving and reducing control channel overhead for cooperative spectrum sensing in cognitive radio networks. Wireless Personal Communications, 2019. 104(2): p. 837-851.
3. Rueda, A. A survey of traffic characterization techniques in telecommunication networks. in Proceedings of 1996 Canadian Conference on Electrical and Computer Engineering. 1996. IEEE.
4. Shahbar, K. and A.N. Zincir-Heywood. How far can we push flow analysis to identify encrypted anonymity network traffic? in NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium. 2018. IEEE.
5. Axelsson, S., Intrusion detection systems: A survey and taxonomy. 2000, Technical report.
6. Wang, P., Y. Li, and C.K. Reddy, Machine learning for survival analysis: A survey. ACM Computing Surveys (CSUR), 2019. 51(6): p. 110.
7. Namdev, N., S. Agrawal, and S. Silkari, Recent advancement in machine learning based internet traffic classification. Procedia Computer Science, 2015. 60: p. 784-791.
8. Cheng, Y., et al., Bridging machine learning and computer network research: a survey. CCF Transactions on Networking, 2019. 1(1- 4): p. 1-15.
9. Mukkamala, S., G. Janoski, and A. Sung. Intrusion detection: support vector machines and neural networks. in proceedings of the IEEE International Joint Conference on Neural Networks (ANNIE), St. Louis, MO. 2002.
10. Taylor, V.F., et al., Robust smartphone app identification via encrypted network traffic analysis. IEEE Transactions on Information Forensics and Security, 2017. 13(1): p. 63-78.
11. Kim, J., et al., Multivariate network traffic analysis using clustered patterns. Computing, 2019. 101(4): p. 339-361.
12. Shafiq, M., et al. Network traffic classification techniques and comparative analysis using machine learning algorithms. in 2016 2nd IEEE International Conference on Computer and Communications (ICCC). 2016. IEEE.
13. Sommer, R. and V. Paxson. Outside the closed world: On using machine learning for network intrusion detection. in 2010 IEEE symposium on security and privacy. 2010. IEEE.