A LEAP TOWARDS DEEP LEARNING FOR INVESTIGATING TOR TRAFFIC

VINDHYA RANI KOLURU [1], DR SAMPATH [2],
INFORMATION TECHNOLOGY AND COMPUTER APPLICATIONS,
ANDHRA UNIVERSITY COLLEGE OF ENGINEERING.

Abstract—TOR (The Onion Routing) is famous among the anonymous users over internet. Due to this anonymous presence of network activity it is very difficult for network surveillance .70-80% of illegal activities use multiple IP addresses which is provided in TOR .Now it's very important to investigate our TOR network in order for providing cyber security. Now a model is proposed using deep learning and neural network for investigating TOR traffic by analyzing time based features.

Keywords—Anonymity, Deep Learning, OR Traffic, Cyber Security.

I. INTRODUCTION

TOR stands for The Onion Routing. It uses onion routing protocol to direct traffic over internet for free, overlay network consisting thousands of relay servers for concealing the identity from network surveillance. As a result of these features TOR traffic accounts to up to 20% of illegal activities. Due to the advancement in technology the crimes also have taken new face .Now we really need an efficient method for analyzing the Tor in order to detect the crimes. Intruder's main aim is to steal confidential data from unauthorized sources for sensitive information. They steal that data and transfer it from one end to the other using TOR along with the normal encrypted traffic. The data is encrypted hence we can use the time based features for investigating the traffic.

Now deep learning has extended to User and entity Behavior Analytics (UEBA).UEBA uses anomaly detection techniques and machine learning algorithms to find any deviations from the proposed behavior and send alerts to the security analysts.

The main objective of this paper is to effectively classify the TOR traffic form the normal ones. Hence compromising the privacy up to some extent by revealing the activity within the tor traffic. Our work is done by extracting several time related features from the flows and hence analyzing them. And also prove that we can characterize the traffic only with time based features effectively. In the past there are very few papers on characterizing the TOR traffic but only on the TOR node itself but not on the traffic generated. And most of the papers have used packet based features. We then generate a deep learning model to classify the traffic.

II. ALL ABOUT TOR

Tor was the idea of, mathematician Paul Syverson, Michael G. Reed and DavidGoldschlag to protect the U.S. intelligence communications online. Onion routing is implemented by encryption in the application layer of the TCP protocol stack.

Tor network is based on the encryption of data between the relay servers. Typically there are many intermediate relay servers between the source and destination host. There will be an entry node and exit node for the TOR network .When the encrypted packet enter its entry node it will be sent to many intermediate relay nodes before reaching the destination through the exit node. The packet is when decrypted will be only containing the details of the next relay node hence concealing the source and destination IP.

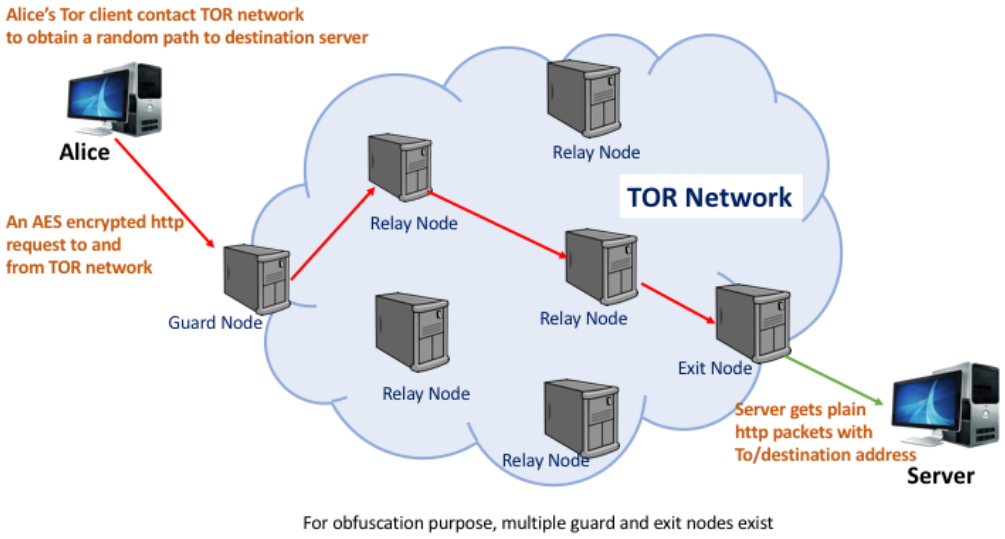


Figure 1: Tor Architecture

III. RELATED WORK

Many research papers have been on TOR .Here are few of them.

- Analysis of TOR traffic (Bai et al., 2008; Chaabane et al.,2010; Ling et al., 2014; AlSabah et al., 2012),but mostly done on one node .
- (Juarez et al., 2014), the author try to identify the browsing pattern, like they try to identify the websites the user is trying to use.
- One more analysis by comparing the similar characteristics of the user at client side and server side (Chakravarty et al., 2014) Chakravarty et al. This is very efficient method and show 100% in lab test and 81 % in real life scenarios.
- (He et al., 2014) HMM(Hidden Markov Models are used to categorize the traffic into FTP,P2P,web and IM. Burst volumes and directions are the classifiers used and achieved an accuracy rate of nearly 92%.
- In few researches they have found that Tor browser leaves traces of digital anti facts which are sensitive hence compromising the anonymity.
- In cyber security the previous anomaly detection techniques were based on rule based, datamining, machine learning techniques. Where the intruders can easily surpass all the above and in many cases there were false alarms generated. Hence we need a smart approach to identify the correct intrusion.

IV. PROPOSED SYSTEM

There are two steps in our proposed system.

- OFFLINE LEARNING: runs several typical applications over TOR network and measures all important time based features.(fiat, biat which are define in the section5)they represent the traffic behaviour .some Deep learning algorithms to model different types of traffic.
- ONLINE LEARNING : Now we classify the targeted TOR traffic with the designed models

V. DATA SET GENERATION

For this experiment we used labelled TOR traffic dataset .The dataset is generated in a simulated environment and to represent the real world behaviour set of tasks are defined. Accounts for users like BOB and ALICE was created for using the services like SKYPE and Facebook. A typical WHONIX (ready to use linux OS is used to route all the traffic through the TOR node. The whonix distribution consists of two virtual machines the gateway and workstation. The workstation connects to the virtual machine gateway and it will act as a connection between the host and the TOR network. The collected samples are from .pcap files one from the workstation (regular traffic) and the other from gateway (TOR traffic).

We now need to analyse the flows and a flow is defined by packets having same values (Source IP, Destination IP, Source Port, Destination Port).Once all the flows are identified and generated we then move on to packet gathering analysis ISCX Flow Meter application (we can even use NET Flow application) to calculate all the parameters from the flows. These include 23 parameters.

- Forward Inter Arrival Time(fiat) :The time between two packets sent forward direction (mean.min,max,std)
- Backward Inter Arrival Time (biat): The time between two packets sent backwards (mean, min, max, std).
- Flow Inter Arrival Time (flowiat): The time between two packets sent in either direction (mean, min, max, std).
- Active: The amount of time a flow was active before going idle (mean, min, max, std).
- Idle: The amount of time a flow was idle before becoming active (mean, min, max, std).
- fb psec: Flow Bytes per second.
- fp psec: Flow packets per second.
- Duration: The duration of the flow.

```
Source IP, Source Port, Destination IP, Destination Port, Protocol, Flow Duration, Flow Bytes/s, Flow Packets/s, Flow IAT Mean, Flow IAT Std, Flow IAT Max, Flow IAT Min,Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min,Bwd IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min,Active Mean, Active Std, Active Max, Active Min,Idle Mean, Idle Std, Idle Max, Idle Min,label
10.0.2.15,53913,216.58.208.46,80,6,435,0,4597.7011494253,435,0,435,435,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,nonTOR
```

V1. EXPERIMENT

For this experiment we used two types labelled data sets one for each flow time out value 5S,10s respectively. Two algorithms have been implemented, Logistic regression and Random Forest Classifier .The metrics used to evaluate. The metrics used to evaluate the performance are Precision, Recall, f1 score, Support.

Precision can be defined as proportion of instances relevant to our class x from along all our retrieved instances. That means the number of relevant instances to the total number of relevant and irrelevant instances.

$$\text{Precision} = \frac{tp}{tp + fp}$$

Tp-number of true positives
Fp-number of false positives

Recall can be defined as proportion of instances relevant to our class from along all instances which truly belong to our class. That means the ratio of relevant instances retrieved to the total number of relevant instances.

$$\text{Recall} = \frac{tp}{tp + fn}$$

Tp-number of true positives
Fn-number of false negatives

Accuracy can be predicted as error rate and calculated as the number of correctly predicted instances from along all instances. That means the ratio of true positives and true negatives to the sum of true positives, true negatives, false positives, and false negatives.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

F1 score can be calculated as weighted average of precision and recall.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Support is the number of actual occurrences in the class of the dataset .Support remains same among all the models. The ROC area can be defined as possibility of a selected positive instance which is randomly chosen is above the negative instance randomly chosen.

The below table is the summary of all results for various algorithms with tested on various datasets.

	Flow out time (10s)					
	Precision	recall	f1score	Target class	accuracy	ROC Area
Logistic regession	0.76	0.69	0.73	Non Tor	0.94	0.8326
	0.96	0.97	0.97	TOR		
Random forest classifier	0.95	0.88	0.91	Non Tor	0.98	0.9918
	0.98	0.99	0.99	TOR		
	Flow out Time (5s)					
	Precision	recall	f1score	Target class	accuracy	ROC Area
Logistic regression	0.79	0.74	0.77	Non Tor	0.92	0.852
	0.95	0.96	0.95	TOR		
Random forest classifier	0.95	0.91	0.93	Non Tor	0.97	0.9915
	0.98	0.99	0.98	TOR		

Figure 2: Comparison of estimators

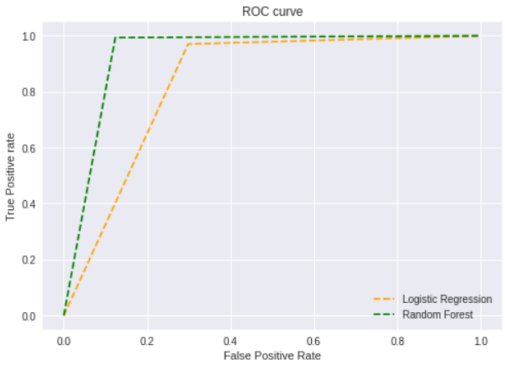


Figure 3: ROC curve

As of now we are done with applying machine learning algorithms, now let's switch to deep learning .Deep learning is suggestible in this case because in real world scenario the traffic is very huge (TB of data generated every minute) and to classify such huge data we need model which is really efficient and fast .In case Generating similar types of applications we need to train the whole dataset for a machine learning model which may use significant resources. But in case of our deep learning approach we will just train the first layer and the remaining layers train themselves.

A feed forward neural network is used which typically 2 to 10 have hidden layers where one hidden layer will feed its output for another hidden layer as input. For our scenario we found n=5 to be the optimal one. All the hidden layers are activated using RELU and the output node is activated by a sigmoid function.

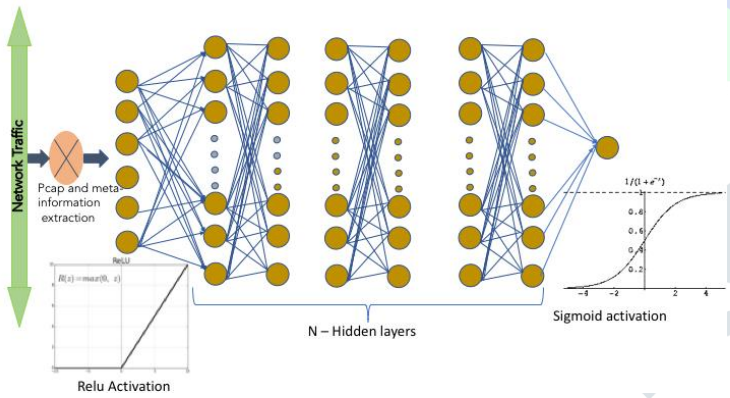


Figure 4: Feed Forward Neural network

Keras and Tensorflow packages are used to generate the Deep learning model and the Feed forward neural network is optimized by binary cross entropy loss.
We can now even observe the results where the accuracy is increased with the number of epochs and the loss is decreased with the increase in epochs.

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 24)	600
dense_2 (Dense)	(None, 128)	3200
dense_3 (Dense)	(None, 128)	16512
dense_4 (Dense)	(None, 128)	16512
dense_5 (Dense)	(None, 128)	16512
dense_6 (Dense)	(None, 128)	16512
dense_7 (Dense)	(None, 128)	16512
dense_8 (Dense)	(None, 128)	16512
dense_9 (Dense)	(None, 128)	16512
dense_10 (Dense)	(None, 128)	16512
dense_11 (Dense)	(None, 1)	129

Total params: 136,025
Trainable params: 136,025
Non-trainable params: 0

Figure 5 : output of the trained model

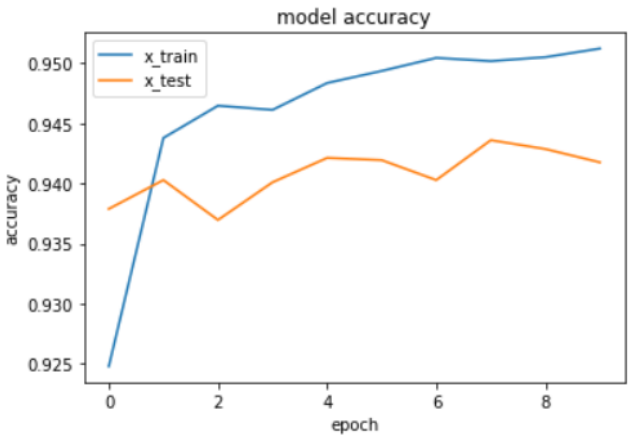
```
Epoch 1/10
 - 4s - loss: 0.2431 - acc: 0.8827 - val_loss: 0.1746 - val_acc: 0.9237
Epoch 2/10
 - 4s - loss: 0.1741 - acc: 0.9230 - val_loss: 0.1706 - val_acc: 0.9209
Epoch 3/10
 - 4s - loss: 0.1692 - acc: 0.9241 - val_loss: 0.1675 - val_acc: 0.9269
Epoch 4/10
 - 4s - loss: 0.1661 - acc: 0.9275 - val_loss: 0.1679 - val_acc: 0.9295
Epoch 5/10
 - 4s - loss: 0.1613 - acc: 0.9319 - val_loss: 0.1566 - val_acc: 0.9367
Epoch 6/10
 - 4s - loss: 0.1558 - acc: 0.9344 - val_loss: 0.1452 - val_acc: 0.9403
Epoch 7/10
 - 4s - loss: 0.1491 - acc: 0.9380 - val_loss: 0.1422 - val_acc: 0.9412
Epoch 8/10
 - 4s - loss: 0.1409 - acc: 0.9416 - val_loss: 0.1317 - val_acc: 0.9445
Epoch 9/10
 - 4s - loss: 0.1376 - acc: 0.9436 - val_loss: 0.1267 - val_acc: 0.9477
Epoch 10/10
 - 4s - loss: 0.1337 - acc: 0.9459 - val_loss: 0.1264 - val_acc: 0.9510

Test acc: 94.53%
6467/6467 [==============================] - 0s 29us/step

Test acc for class 0: 85.12%
31363/31363 [==============================] - 1s 28us/step

Test acc for class 1: 96.47%
```

Figure 6 : Output of the estimators



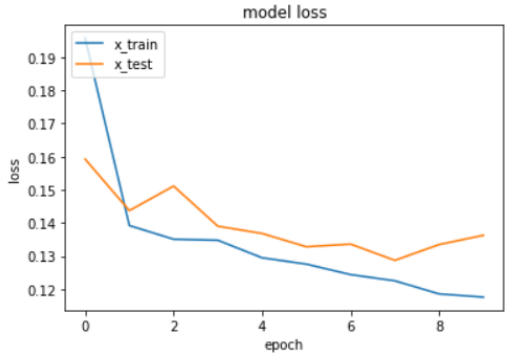


Figure 7&8: Comparison of loss and accuracy

VII. CONCLUSION

When comparing the results with various algorithms we found that Random Forest and Deep learning have the best results for classifying the TOR system. As the dataset taken here is generated in a simulated environment the generated logs are limited to very few numbers. But in real internet traffic this is not the case. The traffic is continuous and a stream of data pours into the dataset. As the data increases the performance will further increase on our deep learning models

REFERENCES

1) Aghaei-Foroushani, V. and Zincir-Heywood, A. N. (2015).A proxy identifier based on patterns in traffic flows.In 2015 IEEE 16th International Symposium on HighAssurance Systems Engineering, pages 118–125.
2) Chakravarty, S., Barbera, M. V., Portokalidis, G., Polychronakis,M., and Keromytis, A. D. (2014). On the effectiveness of traffic analysis against anonymity networks using flow records. PAM 2014, pages 247–257, New York, NY, USA. Springer-Verlag New York, Inc.
3) AlSabah, M., Bauer, K., and Goldberg, I. (2012). Enhancing tor's performance using real-time traffic classification.In Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS '12, pages 73–84, New York, NY, USA. ACM.
4) R. Snader and N. Borisov, "A tune-up for tor: Improving security andperformance in the tor network." in ndss, vol. 8, 2008, p. 127.
5) A. Chaabane, P. Manils, and M. A. Kaafar, "Digging into anonymous traffic: A deep analysis of the tor anonymizing network," in Network and System Security (NSS), 2010 4th International Conference on. IEEE, 2010, pp. 167–174.
6) Bai, X., Zhang, Y., and Niu, X. (2008). Traffic identification of tor and web-mix. In 2008 Eighth International Conference on Intelligent Systems Design and Applications, volume 1, pages 548–551.
7) Dainotti, A., Pescap, A., and Claffy, K. (2012). Issues and future directions in traffic classification. IEEE Network, 26(1):35–40.
8) P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, "Towards and analysis of onion routing security," in Designing Privacy Enhancing Technologies. Springer, 2001, pp. 96–114.
9) A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in Proceedings of the 3rd International Conference on Information Systems Security and Privacy - Volume 1: ICISSP,, INSTICC. SciTePress, 2017, pp. 253–262.
10) He, G., Yang, M., Luo, J., and Gu, X. (2014). Inferring application type information from tor encrypted traffic .In 2014 Second International Conference on Advanced Cloud and Big Data, pages 220–227.
11) ISCXFlowMeter (2016). Information security center of excellence, university new brunswick. http://www.unb.ca/research/iscx/dataset /iscxflowmeter.html.
12) Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
13) R. Singh, H. Kumar, and R. K. Singla, "An intrusion detection system using network traffic profiling and online sequential extreme learning machine," Expert Syst. Appl., vol. 42, no. 22, pp. 8609–8624, 2015. [Online]. Available: https://doi.org/10.1016/j.eswa.2015.07.015
14) Ling, Z., Luo, J., Wu, K., Yu, W., and Fu, X. (2014). Torward: Discovery of malicious traffic over tor. In IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, pages 1402–1410.
15) R. C. Holte, "Very simple classification rules perform well on most commonly used datasets," Machine learning, vol. 11, no. 1, pp. 63–90,