Performance Estimating And Optimizing Neural Network For Botnet Detection In Machine Learning

Dr. Nitya Nand Dwivedi , Dr. Shashiraj Teotia , Mr. Ankur Chaudhary

Department of Computer Application, Swami Vivekanand Subharti University, Meerut U.P , India.

Abstract: These days, botnets are being considered as the most vital security dangers in the web and it is vital to discover new routes for their recognition. Shared (P2P) botnets are the most imperative sorts of botnets that utilization P2P correspondence conventions to control their bots remotely. Along these lines, their recognition is more troublesome than different botnets. In this paper, we propose another way to deal with distinguish P2P botnets in the direction and control (C&C) period of life cycle dependent on the examination of activity conduct. The proposed methodology can distinguish C&C activity of P2P botnets by utilizing stream based highlights and order techniques. The execution of the proposed approach is assessed dependent on various parameters. The aftereffects of the assessment demonstrate that the proposed approach can recognize P2P botnet from ordinary system activity with high identification rate.

Keywords: Botnet Detection, Machine Learning, Orange Data Mining Tool, Networks

1. Introduction

These days, with the advancement of Information and Correspondence Technology (ICT) and spread of PC organize, the quantity of web clients has expanded quickly, and the web has turned into a fundamental piece of our lives. Consistently, a tremendous sum of information is exchanged through the web. In addition, there are a few trespassers or individuals who need to cause harms to the web clients. In this manner, in accordance with the advancement of Information and correspondence Innovation, it is important to give data security for the clients and balance security dangers. Among the security dangers, we can allude to malware, which puts an expansive number of clients in threat. Jihyun al confirm that the deep learning approach is effective for IDS [1]. Malware is a program with malevolent reason, which is intended to pulverize the PC or the system that runs it . Malware comprises of various classifications of projects, for example, infections, Worms, Trojans, Bots, and so on. Todays, botnets have turned into the principle wellspring of assaults in the web, according to Wurzinger, botnet is a network of compromised hosts that is under the control of a single, malicious entity [2]. Approximately the results of botnet detection methods are usually presented without any comparison, even it is generally accepted that more comparisons with third-party methods may help to improve the area [3]. Botnet detection algorithms have been partially successful, but are difficult to reproduce and verify; being often commercialized [4].

According to Preda al current malware (botnet) detectors work by checking for "signatures," which attempt to capture (syntactic) characteristics of the machine-level byte sequence of the malware [5].In machine learning traditional approaches using signatures to detect malicious programs pose little danger to new and unseen programs whose signatures are not available [6]. In machine learning the performance of network is calculated based on bandwidth, throughput, latency etc [7]. According to Moheeb al botnet behavior has never been methodically studied, botnet prevalence on the Internet is mostly a mystery, and the botnet life cycle has yet to be modeled [8]. A botnet is a system of traded off machines associated with the web that is tainted by malignant programming (bot) and is remotely controlled by botmaster. P2P botnets are the freshest sort of botnets that utilization P2P systems to remotely control their bots. P2P botnets have numerous malignant purposes, for example, Spreading spam, Conveyed Denial of Service (DDoS) assaults, malware conveyance and taking critical data. According to Hynsang al botnets can cause severe Internet threats such as DDoS attacks, identity theft, spamming, and click fraud [9]. Accordingly, it is critical to discover aversion approaches to distinguish them in the underlying stage. The issue of ebb and flow look into is about the discovery of P2P botnets in C&C period of life cycle. After considering the problem of identifying obscure chat-like botnet command and control (C & C) communications, which are indistinguishable from human-human communication using traditional signature-based techniques [10]. Albeit a few methodologies have been proposed for the discovery of P2P botnets, their recognition is testing on account of the accompanying reasons: Botnet movement is like ordinary system activity.

Moreover, now and again botnets utilize encoded correspondence directs with the end goal to counteract recognition. In this manner, approaches that play out the recognition dependent on the investigation of parcel content, can't distinguish them. Moreover, a few approaches need to investigate a lot of information, which is not really conceivable to be performed progressively for an extensive scale arrange. At long last, P2P botnets recognition is all the more difficult in correlation with different botnets.

In this investigation, another way to deal with P2P botnet recognition in C&C period of life cycle and before their assault is proposed. To this end, we break down movement conduct with the end goal to recognize P2P botnets by flow based highlights. Subsequently, we make utilization of order strategies in information mining to recognize request to accomplish the best location rate with the least computational prerequisites, after investigating Elhalabi al botnets have exploited this technology efficiently and introduced the P2P botnet, which uses P2P network for remote control of its bots and become one of the most significant threats to computer networks [11].

These pernicious assaults happen when web associated gadgets are seized. When they’re assumed control by a programmer, gadgets are normally contaminated with malware and controlled from a remote area by a solitary programmer. After surveying M. Feily al defining characteristic of botnets is the use of command and control channels through which they can be updated and directed. Recently, botnet detection has been an interesting research topic related to cyber-threat and cyber-crime prevention [12]. Globally Internet threats have undergone a profound transformation from attacks designed solely these attacks are collections of compromised computers, or Botnets, remotely controlled by the attackers, and whose members are located in homes, schools, businesses, and governments around the world [13,14]. In spite of the fact that a programmer can penetrate your gadget with a botnet, it can likewise utilize a multitude of botnets to bring down your website or online business. A digital assault utilizing botnets penetrated 1.5 million associated cameras to surge a columnist’s site, overpower it, and power it to go down. What’s more, that is exactly what botnets did to a solitary site. Envision what programmers could do with those 1.5 million gadgets in the event that they got to them to keep an eye on individuals in their homes or attempted to penetrate their private information. On the off chance that you haven’t been hit by an assault yet, you should remain alarm to the inescapable risk. In this present reality where the Internet of Things (IoT) market will develop from 15.4 billion gadgets in 2015, to 30.7 billion gadgets in 2020 and up to 75.4 billion by 2025, botnet recognition and expulsion is critical for our computerized wellbeing. Here are some accepted procedures and techniques to battle botnets and remain responsible for your gadgets, the Peer-to-Peer (P2P) botnet is a new type of botnets a peer controlled by hackers and thus its defense is more difficult [15].

At long last, a botnet discovery technique through closeness examination of groups has been produced for continuous discovery of new bots. According to K. Muthumanickam al botnet is a collection of interconnected compromised computers (Bots) which are remotely controlled by its owner (Bot Master) under a common command-and-control(C&C) infrastructure [16].The bunches are produced utilizing Expectation Maximization (EM) bunching calculation. This technique depends on highlights that precisely coordinate amid an age (commonly one day). The general sizes of groups are assessed, from swhere the dominant part bunches are considered for further assessment. At that point, copy streams from larger part groups are evacuated and decrease in size of the larger part bunch is legitimately evaluated. At long last, by utilizing Jaccard closeness coefficient on the sets acquired from diminished groups bots are precisely recognized in an observed system. This structure has additionally been tried on botnet, which are networks formed by malware-compromised machines, have become a serious threat to the Internet [17], dataset arranged from Nugache, Waledac and P2P Zeus botnet follows. Sheharbano Khattak al providing information can be used to devise integrated detection strategies by combining complementary approaches [18]. the Bot Net-the network of Bots-detection is difficult and possible only after they have spread widely[19]. Botnets are collections of compromised computers which are remotely controlled by its originator (Bot Master) under a common Commond-and-Control (C&C) infrastructure [20]. Pijush Barthakur al provide a comparative analysis of machine-learning based classification of botnet command & control(C&C) traffic for proactive detection of Peer-to-Peer (P2P) botnets [21]. Justin Leonard al presenting a framework, which especially suggests a general architecture that could be coupled with certain advanced techniques that have not been exploited in existing botnets [22]. P2P botnets are resilient and more difficult to detect due to their nature of using different distributed approaches and encryption techniques [23]. P. Narang al proposes Peer Shark, a novel methodology to detect P2P botnet traffic and differentiate it from benign P2P traffic in a network [24]. The normal exactness, affectability and positive prescient estimation of the classes to bunch order models from three diverse botnet datasets are 0.895, 0.959 and 0.906 separately. The normal level of streams in larger part bunches of Nugache, Waledac and P2P Zeus botnet datasets are 81.39, 82.315 and 74.15 separately. Likewise, the normal level of decrease in group size of the dominant part bunches of Nugache, Waledac and P2P Zeus botnet datasets are 96.69, 92.365 and 87.48 separately. A Jaccard similitude coefficient between diminished bunches that have a place with bots inside the equivalent botnet are 0.1926, 0.2157 and 0.1008 for Nugache, Waledac and P2P Zeus separately, we can present a figure of botnet life cycle, which is shown in figure 1 as follows:.

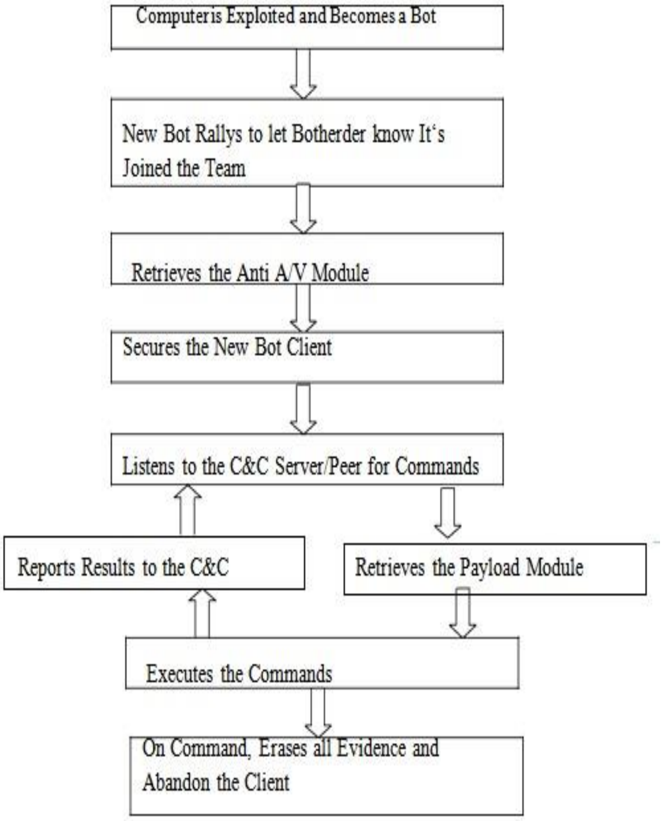

Computer is Exploited and Becomes a Bot
New Bot Rallys to let Botherder know It's Joined the Team
Retrieves the Anti A/V Module
Secures the New Bot Client
Listens to the C&C Server/Peer for Commands
Reports Results to the C&C
Retrieves the Payload Module
Executes the Commands
On Command, Erases all Evidence and Abandon the Client

Figure 1. Flow chart for botnet life cycle

2. Methodology

Now we can use orange machine learning software for for comparing the different dataset. The broad methodology adopted is to first develop efficient classification algorithms for botnet C&C traffic and then to develop an efficient result to new P2P botnets. Accordingly, Highlight determination is an imperative issue that influences the precision of identification. The subject of distinguishing futile, less critical and genuinely helpful highlights is pertinent in light of the fact that the exactness of identification, the computational speed and the general execution of the recognition framework can be fundamentally upgraded by dispensing with pointless highlights. In situations where there are no futile highlights, focusing on the most noteworthy ones perhaps will enhance the execution of the location component.

We can use orange software for comparing the CTU-13 files. The three scenarios compare the each files like Background, Botnet and Normal. Now, the best results will come in CTU-13-11 dataset. The score is give below in table 1.

We can compare all the four classifier. Where KNN is scoreless as compare to Neural Network then we can again compare result to Neural Network with Naïve Bayes, again compare to Neural Network with Logistic Regression. Now we can have good result score in Neural Network.

In table 2 we can show the Confusion matrix for Neural Network (showing number of instances). In this matrix we can show the Background, Botnet and Normal, whereas the Background number is Predicted number.

- Orange machine learning environment has been chosen to perform the classifications and to generate the Botnet. Orange provides a collection of Machine Learning (ML) algorithms and several visualization tools for data analysis and predictive modeling. Before using in the classification / Botnet processes, datasets are passed through Randomize filter available with Orange unsupervised instance filter category for randomization of instances.
- The Expectation-Maximization (EM) Neural Network algorithm is selected for generation because of its ability to generate soft Network the Jaccard Similarity Coefficient selected for similarity analysis of the generated clusters for detection of new botnets. Figure shows the complete machine learning based framework for real-time detection of P2P Botnets.
- Orange machine learning environment has been chosen to perform the classifications and to generate the clusters. Orange provides a collection of Machine Learning (ML) algorithms and several visualization tools for data analysis and predictive modeling. Before using in the classification processes, datasets are passed through Randomize filter available with, Orange unsupervised instance filter category for randomization of instances.

2.1 Botnet Traffic Classification using KNN

K-Nearest Neighbor (KNN) is used for classification of large volume of control traffic generated by a P2P botnet from that of normal web traffic. A model selection of KNN is a type of instance-based learning, where the function is only approximated locally and all computation is deferred until classification. The KNN algorithm is among the simplest of all machine learning algorithms. The KNN classifier offers an alternative approach to classification using lazy learning that allows us to make predictions without any model training but at the cost of expensive prediction step.

2.2 A Rule based Classification Model using C4.5 Algorithm

A rule induction method for botnet traffic classification is proposed. An indirect method is used to derive the initial rule set from the decision tree generated using C4.5 algorithm. This is followed by a step-by-step approach for optimization of the rule set. The final rule set has a uniform structure providing significant insight in to similarities within P2P botnet C&C traffic.

2.3 Generation of Fuzzy Rules for Botnet C&C Traffic Classification

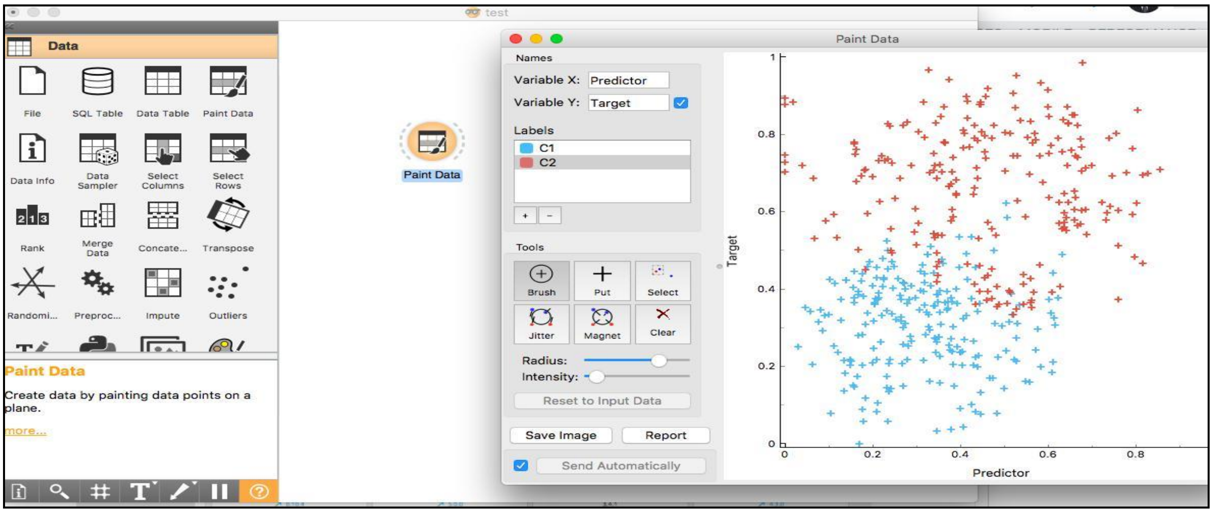
Another framework proposed for botnet C&C traffic classification is through generation of fuzzy rules using Fuzzy Unordered Rule Induction Algorithm (FURIA). Fuzzy logic often leads to creation of small rule, where each rule is an embodiment of meaningful information. Moreover, there is an inherent fuzziness in security issues and thus an approximate fuzzy rule set can be generated for detection of security threats. Inference using conventional rules depends on crisp boundaries that lead to abrupt transition between the two classes. However, a more general rule where its support for a class decreases from ―full‖ (inside the core of the rule) to ―zero‖ (near the boundary) in a gradual rather than an abrupt way is more appropriate. Therefore, a set of fuzzy rules that have ―soft‖ boundaries definitely has merit.

2.4 Botnet Detection through Similarity Analysis

Botnet detection framework has been developed for real-time detection of P2P bots in a monitored network. Every botnet uses a specific set of commands. Commands frequently exchanged between different peer bots represent flows whose structural characteristic matches with one another. These flows, when considered separately are low in volume, as very less number of packets is transferred and packet sizes are usually small. But, bot C&C flows are high in frequency during initial stages of infection. Therefore, Expectation-Maximization (EM) is used for botnet C&C flows on its structural characteristics. The numbers of generated is fixed two. The efficiency has been analyzed using classes-to-Botnet evaluation method in Orange machine learning environment. The detection framework comprises of three modules: flow clustering module, flow reduction module and similarity analysis module. In the flow clustering module, a comparison is drawn between the two clusters generated using a dataset. If the clusters generated are highly imbalanced, further analysis is done using only the larger cluster (also considered as subject cluster for further analysis) where in the flow reduction module duplicate entries in it are removed. This gives an assessment of the amount of reduction in the subject cluster, which is usually very high in case of bots. After reduction, a set is obtained from the subject cluster. In the similarity analysis module, the Jaccard similarity coefficient is calculated to analyze similarity between such sets derived from probable bots.

2.5 Details

Orange is a platform built for creating machine learning pipelines on a GUI workflow. People with no coding skills can operate Orange with ease. One can perform every task right from data preparation to model evaluation without writing a single line of code. It also has many cool features that I didn't find in many other heavyweight tools. Have you ever painted data? You heard me right. You can paint data in Orange using its Paint Data functionality. It means you can create dummy data as per your requirements just by drawing the data points, and Orange will generate the data for you. This is a unique feature that is much needed for people who experiment a lot with data to come up with a prototype. We painted the data red and blue in Orange in the below figure 2.



Paint Data

Create data by painting data points on a plane.

more...

Names

Variable X: Predictor

Variable Y: Target ☑

Labels

🟦 C1
🟥 C2

Radius:

Intensity:

☑ Send Automatically

Figure 2. Representing Data on Orange.

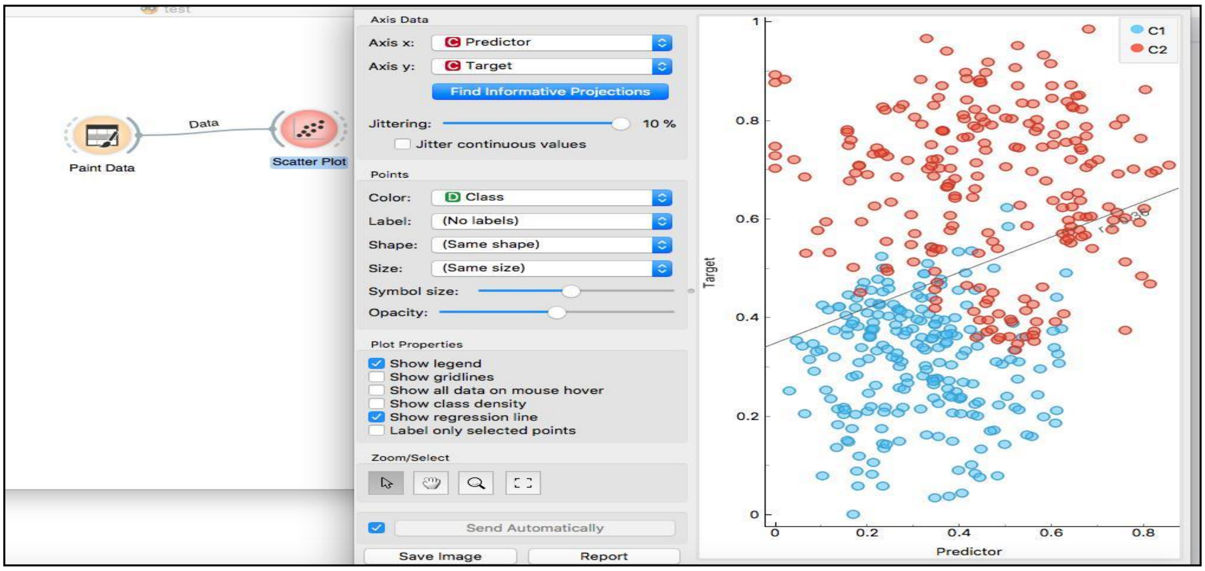
Apart from this, it also has many differentiators like good visualization capabilities, an extensive list of models, and evaluation techniques. Let's peep at the tool by creating a machine model using the painted data we created earlier.
Orange has mainly four different tabs.

2.5.1 Data

It has around 26 different functions. One can extract data from different sources like files, SQL tables, and data tables. You can paint data, sample, merge, and select data. You can even construct features, detect outliers, and preprocess data. The list is long, and a plethora of data-related stuff is available at user's disposal.

2.5.2 Visualize

Around 15 different types of visualization are available, which can be used to view data across various dimensions. For our painted data, I created a quick scatter plot by connecting the **Paint Data** icon to the scatter plot diagram. In each of the visualizations, there is a handful of functions that can be used for creating marvelous plots. In our scatter plot shown below, I displayed a regression line using the **Show Regression Line** plot property. We can distinctly validate that, as there are two classes, c1 and c2, in our dataset, linear regression is not an appropriate technique. We painted the **Show Regression Line** in Orange in the below figure 3.



Axis Data
Axis x: Predictor
Axis y: Target
Jittering: 10 %
☐ Jitter continuous values

Points
Color: Class
Label: (No labels)
Shape: (Same shape)
Size: (Same size)

Plot Properties
☑ Show legend
☐ Show gridlines
☐ Show all data on mouse hover
☐ Show class density
☑ Show regression line
☐ Label only selected points

☑ Send Automatically

Figure 3. Regression Line

2.5.3 Model

There are ten supervised ML modeling functions. Let's create a decision tree model for the dataset we created earlier decision tree model for the dataset in Orange in the below figure 4.

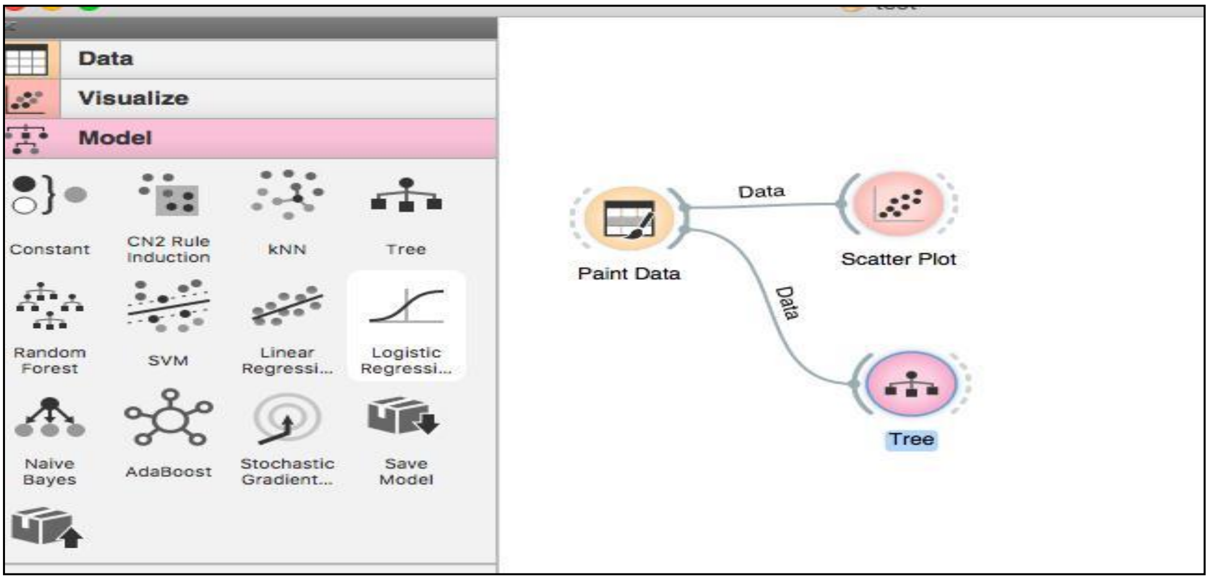
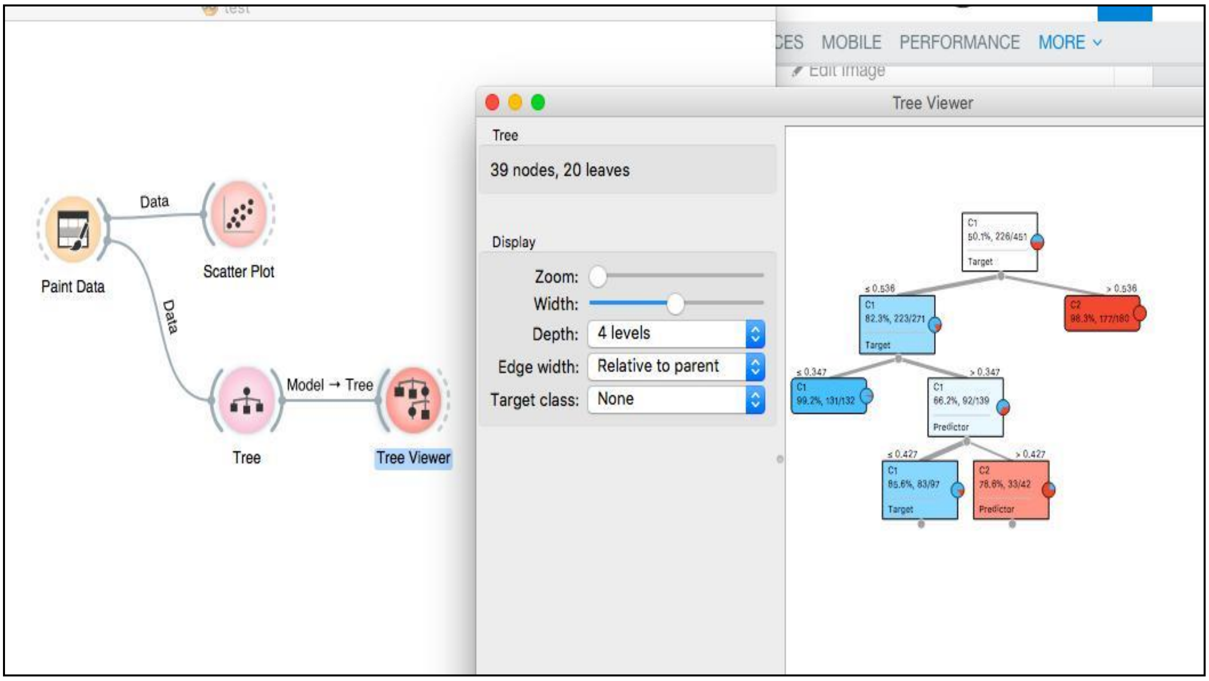


Figure 4. Create a Decision Tree Model for The Dataset

So, our classification model is now ready. How convenient was it? Super easy for me. Let's quickly visualize the tree model. We can select the Tree Viewer from the Visualization section to view the model as shown in below figure 5.



Tree Viewer

Tree

39 nodes, 20 leaves

Display

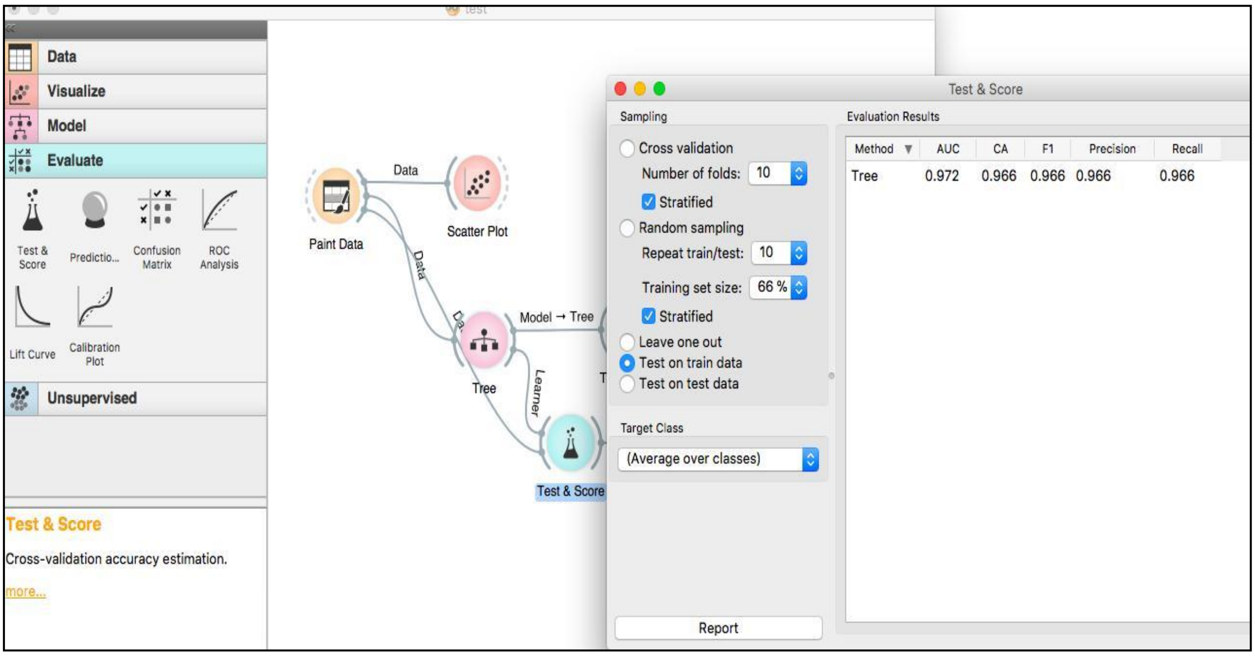
Zoom:
Width:
Depth: 4 levels
Edge width: Relative to parent
Target class: None

Figure 5. Decision Tree Model

Now that our model is ready, let's move to the next section to evaluate the accuracy of the model.

2.5.4 Evaluate

The Test & Score node when connected to the Tree model and Test data node provides the scores of various evaluation metrics. For our painted data model, the AUC is 0.972 and F1 are 0.966, which confirms that it is a sound model as figure 6.



Test & Score

Cross-validation accuracy estimation.

more...

Sampling

○ Cross validation
Number of folds: 10
☑ Stratified
○ Random sampling
Repeat train/test: 10
Training set size: 66 %
☑ Stratified
○ Leave one out
● Test on train data
○ Test on test data

Target Class

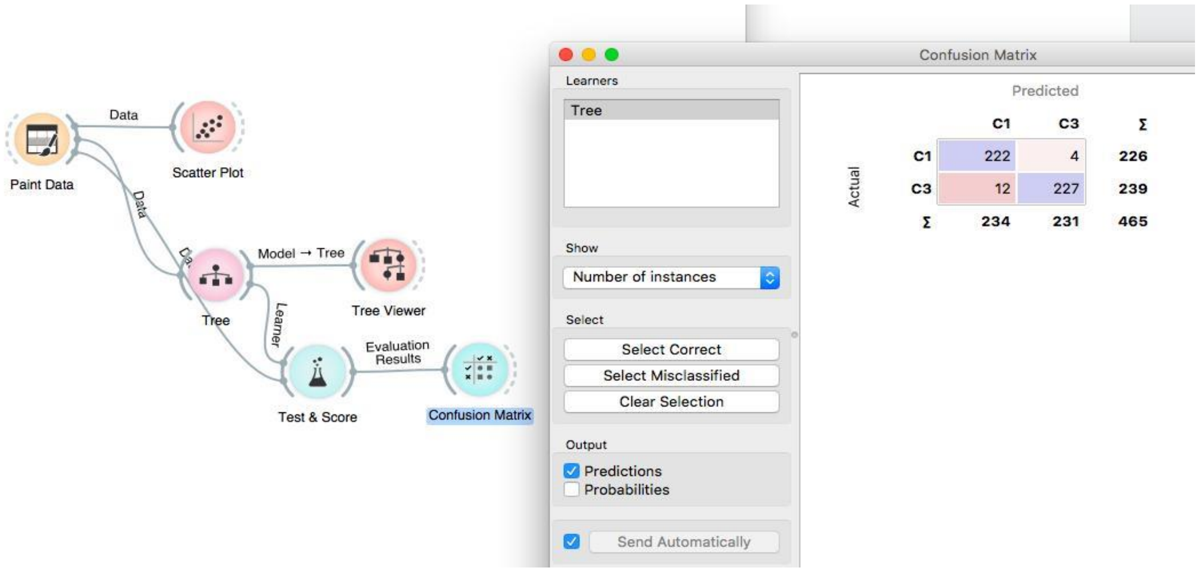
(Average over classes)

Evaluation Results

Method	AUC	CA	F1	Precision	Recall
Tree	0.972	0.966	0.966	0.966	0.966

Figure 6. The Test & Score node when connected to the Tree model

We can also view the confusion matrix by connecting the confusion matrix node to the Test & Score node. We can also view the confusion matrix in Orange in the below figure 7.



Confusion Matrix

Learners

Tree

Show

Number of instances

Output

☑ Predictions
□ Probabilities

☑ Send Automatically

		Predicted		
		C1	C3	Σ
Actual	C1	222	4	226
	C3	12	227	239
	Σ	234	231	465

Figure 7. The confusion matrix by connecting the confusion matrix node

It was a matter of 30 minutes to create a model using Orange. Without any prior experience on using a tool, the same kind of demo model takes more than an hour to be built using other open-source tools. In this way we have to implement a program in Orange machine learning. We have to use CTU-13 dataset and each datasets have four classifications where we can compare all the classifier. The four classes are- Normal, Background, Botnet and C&C. but we have a highest score target classes are Background. We have to use many classifiers but the result is best to Neural Network. In the four classifier the Neural Network have score highest in comparison of KNN, Navie Bayes and Logistic Regression. The dataset is used in these classifier is CTU-13-11 which has highest score in other datasets. The data's In P2P botnets the C&C server is concealed . Each bot can act as a client or server and the botmaster can perform its attacks from each computer . The main feature of P2P botnet is that all peers can play the role of C&C server. Because P2P botnets have decentralized C&C structure, they do not have the problem of centralized structure. In other words, if one bot is taken down, its effect on the whole botnet will be less and the botnet will remain under the control of other bots. On the other hand, management and maintenance of P2P botnets in comparison with centralized botnets is more complex. Fig. 8 shows, structure of the decentralized C&C.

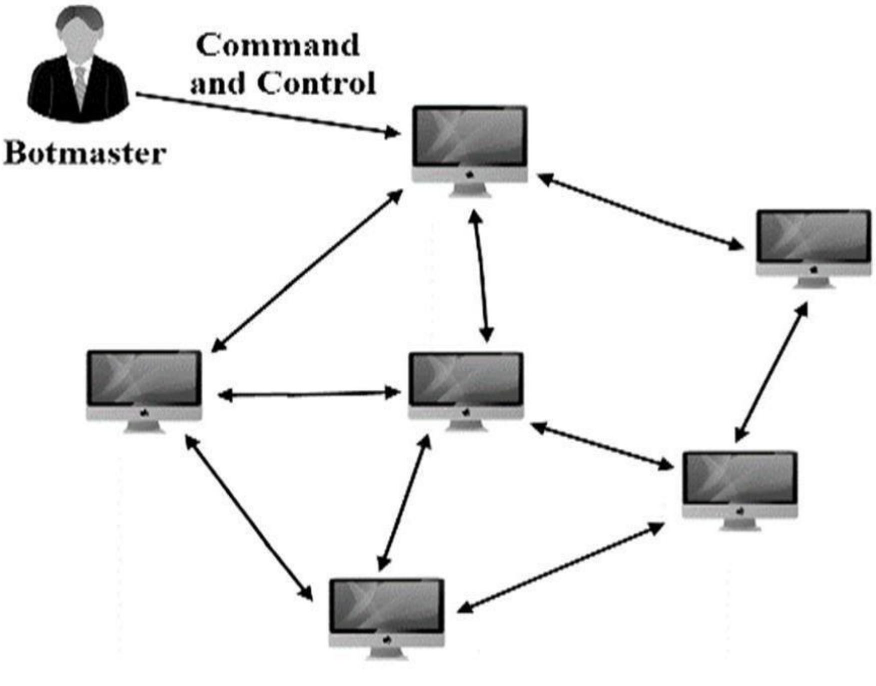


Figure 8. Decentralized C&C structure

3. Results / Applications

Highlight determination is an imperative issue that influences the precision of identification. The subject of distinguishing futile, less critical and genuinely helpful highlights is pertinent in light of the fact that the exactness of identification, the computational speed and the general execution of the recognition framework can be fundamentally upgraded by dispensing with pointless highlights.

In situations where there are no futile highlights, focusing on the most noteworthy ones perhaps will enhance the execution of the location component.

- We can use orange software for comparing the CTU-13 files. The three senariose compaire the each files like Background, Botnet and Normal. Now, the best results will come in CTU-13-11 dataset. The score is give below in table 1.

Beginning arrangement of stream level highlights has been chosen via deliberately contemplating C&C conduct of P2P botnets. In accordance with contemporary research in discovery of P2P botnets, eleven botnet stream and conduct trademark highlights have been considered for the present examination, as expressed here under:

Table 1. Settings

Sampling type: Stratified Shuffle split, 10 random samples with 66% data
Target class: Average over classes

Scores

Method	**AUC**	**CA**	**F1**	**Precision**	**Recall**
KNN	0.740	0.994	0.992	0.992	0.994
Neural Network	0.840	0.993	0.990	0.986	0.993
Naive Bayes	0.869	0.979	0.984	0.989	0.979
Logistic Regression	0.755	0.993	0.990	0.986	0.993

- In above Table 1 we can compare all the four classes. Where KNN is scoreless as compare to Neural Network then we can again compare result to Neural Network with Naïve Bayes, again compare to Neural Network with Logistic Regression. Now we can have good result score in Neural Network.

In table 2 we can show the Confusion matrix for Neural Network (showing number of instances). In this matrix we can show the Background, Botnet and Normal, whereas the Background number is Predicted number.

Table 2. Confusion matrix for Neural Network (showing number of instances)

		Predicted			
		Background	**Botnet**	**Normal**	∑
Actual	**Background**	318363	0	27	**318390**
	Botnet	70	0	0	**70**
	Normal	2140	0	0	**2140**
	∑	**320573**		**27**	**320600**

- In figure 9 & 10, we can AOC Analysis is defining the Neural Network is showing in green line, whereas KNN is showing in orange line after that Naive Bayes is showing in purple, at the last is Logistic Regression is showing in pink. So that the highest score is Neural Network.

Target class: Background
Costs: FP = 500, FN = 500
Target probability: 50.0 %

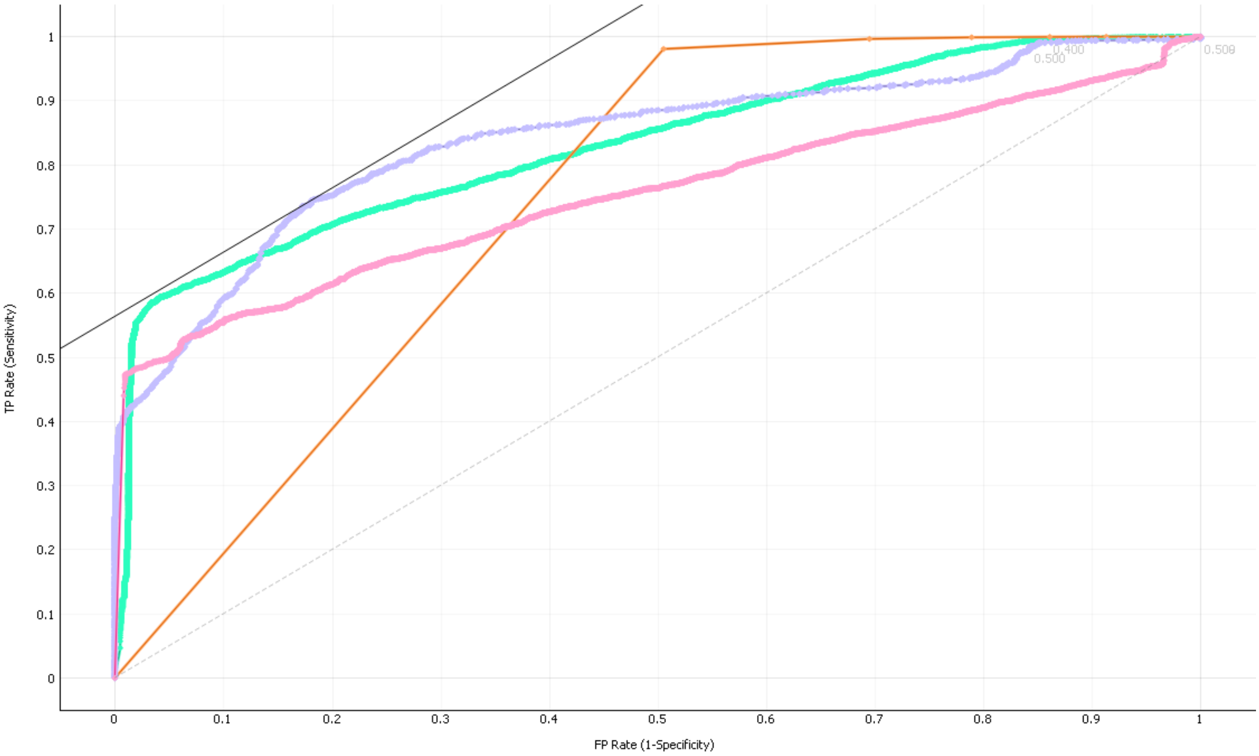


Figure 9. Analysis is defining the Neural Network, KNN, Naive Bayes and Logistic Regression

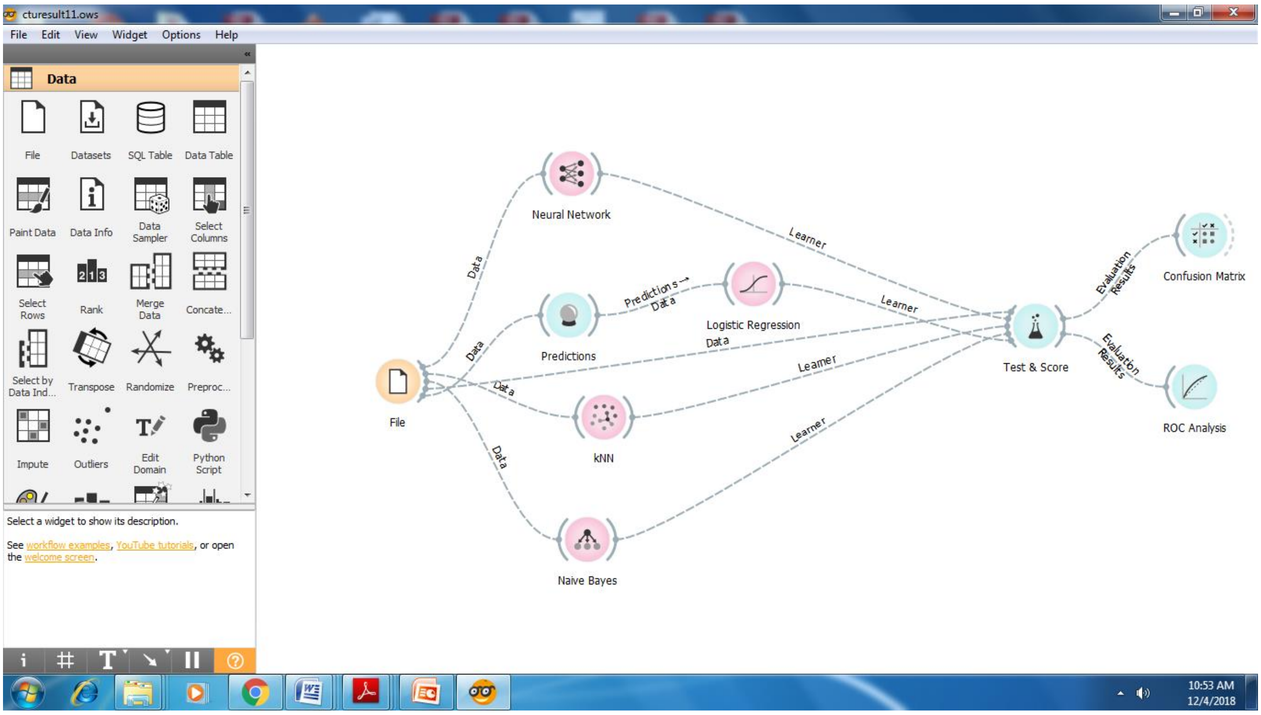


Figure 10. Test & Score Result on Orange

3.1 Data Set Preparation

The kindhearted movement tests were gathered arbitrarily from windows machines utilizing Wireshark. The kindhearted activity tests incorporate differed movement, for example, HTTP, FTP, SMTP and so forth including movement caught from authentic P2P applications. P2P record sharing by generous P2P applications includes rich website page exchanges and regularly conveys bundle to the extent of MTU.

Botnet C&C movement tests were gathered from the accompanying sources: The Nugache botnet C&C activity was acquired from Department of Computer Science, The University of Texas at Dallas. This is the equivalent botnet movement test utilized in the botnet related research works. So also, Waledac and P2P Zeus movement follows were acquired from Department of Computer Science, University of Georgia. These follows were likewise utilized in the botnet related research works. Bot parcels and amiable bundles are gathered from various systems. Be that as it may, parcels caught from every one of the systems are Ethernet bundles and has the equivalent MTU measure. This is the essential purpose behind which the factual highlights including stream payload over a period (transient qualities) are given slightest inclination while setting up the datasets. Besides, in the list of capabilities determination approach portrayed in Section 3.1, the transient qualities AIT and VIT having no effect on level of right order, genuine positive rate and false positive rate and subsequently are observed to be of slightest noteworthy in the discovery models.

During the time spent making the datasets, a Perl content is utilized to remove streams from the three P2P botnet follows and the typical web movement follows. A stream is characterized by blend of following 5-tuples: <source IP, goal IP, convention, source port, goal port>. The parcels having same incentive for these 5-tuples are totaled into one stream. Amid the stream extraction process certain classifications of streams were disposed of viz. (I) streams having single parcel and

(ii) streams that includes nearby communicated exercises in the system. Purposes behind disposing of these Streams are as per the following: (I) streams conveying single parcel does not convey any significant factual data, and if single bundle streams are not disposed of, the ‗proportion of biggest measured parcel' quality qualities in the dataset would move toward becoming ‗1', which would thusly antagonistically influence the order result. (ii) The bot tainted hosts may include in nearby communicates exercises. Be that as it may, the essential concentration here is to consider have to-have coordinated cooperation in the system and communicate movement is never part of bot C&C connection. In this way, such activity is labeled as undesirable for the grouping model.
Streams removed from every one of the three P2P botnet follows and typical web activity follows are additionally handled as pursues:

(i) Six datasets having 15000 streams in each are removed from six diverse P2P bot's C&C activity tests. Out of this six, two has a place with two diverse Nugache bots, another two has a place with two distinctive Waledac bots and the staying two has a place with P2P Zeus bot. These six datasets are then scaled somewhere in the range of 0 and 1 for each element.

(ii) Flows removed from typical web movement follows are assembled in to six a balance of having 5000 streams in each and are then scaled similarly somewhere in the range of 0 and 1 for each element.

(iii) One each from botnet dataset is joined with one each from typical part to make six composite datasets of 20000 streams in each.

3.2 Description of P2P Botnets used in Experimentation

An unadulterated shared botnet is a decentralized design permitting botmaster to utilize any companion indiscriminately to disperse order to other friend bots in the P2P organize. A portion of the outstanding P2P botnets are Nugache, Storm, Waledac and P2P Zeus . Nugache is the unadulterated P2P bot ancient rarity that does not rely upon any focal server including DNS. It handles C&C through encoded P2P Channel utilizing a variable piece length RSA key trade, which is utilized to seed symmetric Rijndael-256 session keys for each friend association. Each Nugache peer holds a rundown of up to 100 worker peers with the end goal to rejoin the system again on the off chance that it gets disengaged. Waledac botnets totally relies upon the utilization of HTTP correspondence and a quick motion based DNS organize for its C&C activities. Every last one of Waledac twofold conveys a rundown of IP delivers to make starting association with the waledac organize. Waledac paired likewise contains a hardcoded URL to get to the botnet, in the event that the bot neglects to distinguish a functioning hub in its location list. The hardcoded URL typically searches for an area which is a piece of the quick transition organizes made by the botnet. For C&C activities, each Waledac bot at first create an inner open authentication and sends to the head-end C&C server. The head-end C&C server scrambles the correspondence key (essential for the bot to associate with the botnet) utilizing the inside open endorsement. Following this, it sends the encoded key back to the bot. This key is decoded and utilized by the Waledac bot, for future correspondences with the botnet. The mainstream unified form of the Zeus botnet has been adjusted to make a stronger P2P variation known as P2P Zeus or Game Over. Numerous virtual sub-botnets are made by separating the primary P2P organize into a few sections by utilizing a hardcoded sub-botnet identifier present in each P2P Zeus' bot paired. These sub-botnets are freely controlled by a few botmasters, despite the fact that the primary P2P system of Zeus is kept up and refreshed as a solitary element. To reach the botnet, the bot double conveys a hardcoded list involving IP locations, ports and one of a kind identifiers of up to 50 Zeus bots. Companion list refreshing is done through a push-/pull-based friend list trade system. Zeus bot checks responsiveness of their neighbors at regular intervals. Each neighbor is reached thus and given 5 chances to answer. In the event that a neighbor does not answer inside 5 retries, it is esteemed inert and is expelled from the friend list. On the off chance that its whole neighbor ends up inert, a Zeus bot endeavors to re-bootstrap on to the system by reaching peers in its hardcoded peer list. On the off chance that this additionally comes up short, the bot utilizes a DGA reinforcement channel to recover a new RSA-2048 marked companion list.

4. Conclusions

The Neural Network strategy beat all other identification meth-ods with its execution as appeared in Table 2. The strategy requires an extensive number of information focuses and furthermore needs an equivalent dispersion of both pernicious and typical streams to prepare successfully. In pragmatic circumstances it isn't conceivable to acquire level with number of vindictive also, ordinary streams. Table 2 abridges the necessities of each recognition technique. The Botnet technique does not require a gigantic dataset and square with dissemination of streams in the two classes despite the fact that it has a lower execution. RNNs have the value of not requiring manual element designing yet to prepare the model it requires a bigger dataset. The CTU Dataset was not ready to give enough preparing precedents to prepare this model viably. Further, the performance of RNNs can be enhanced with a bigger dataset.

5. References:

1. H. L. T. Thu, H. Kim J. Kim, J. Kim, “Long Short Term Memory Recur-rent Neural Network Classifier for Intrusion Detection”, In Proceedings of IEEE International Conference on Platform Technology and Service. Jeju, Korea, 1–5, 2016.
2. T. Holz J. Goebel C. Kruegel E. Kirda P. Wurzinger, L. Bilge, “Automatically generating models for botnet detection”, In Proceedings of European Symposium on Research in Computer Security: ESORICS. 232–249, 2009.
3. G. Sebastian, Martin Grill, Jan Stiborek and Alejandro Zunino, “An empirical comparison of botnet detection methods”, Computers & Security 45, 100–123, 2014.
4. G. Sebastian, “Identifying, Modeling and Detecting Botnet Behaviors in the Network, Ph.D. Dissertation, Universidad Nacional del Centro de la Provincia de Buenos Aires, 2014.
5. M. D. Preda, M. Christodorescu, S. Jha, and S. Debray, “A semantics-based approach to malware detection”, ACM Transactions on Programming Languages and Systems" (TOPLAS), Vol. 30, 2007.
6. M. A. Siddiqui and Morgan Wang, "Data mining methods for malware detection", Ph.D. Thesis, College of Sciences at the University of Central Florida Orlando, Florida, 2008.
7. Nitya Nand Dwivedi and Zaidi, Taskeen, “Performance Analysis of Distributed Networks”, International Journal of Advanced Science and Technology, Vol. 29, No.3, pp.3283-3300, 2020.

8. M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “Amulti faceted approach to understanding the botnet phenomenon”, SIGCOMM conference on Internet measurement, ACM, pp. 41-52, 2006.
9. H. Choi, H. Lee, and H. Kim, "Bot GAD: detecting botnets by capturing group activities in network traffic", International ICST Conference on Communication System software and middleware, ACM, p. 21-28, 2009.
10. G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee, "Active botnet probing to identify obscure command and control channels", Annual Computer Security Applications Conference, IEEE, pp. 241-253, 2009.
11. M. J. Elhalabi, S. Manickam, L. B. Melhim, M. Anbar, and H. Alhalabi, "A Review Of Peer-To-Peer Botnet Detection Techniques", Journal of Computer Science, Science Publications, Vol. 10, pp. 169-177, 2014.
12. M. Feily, A. Shahrestani, and S. Ramadass, "A survey of botnet and botnet detection", International Conference on Emerging Security Information, Systems and Technologies, IEEE, pp. 268-273, 2009.
13. M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir,"A survey of botnet technology and defenses", Cybersecurity Applications & Technology Conference for Homeland Security, IEEE, pp. 299-304, 2009.
14. Evan Cooke, Farnam Jahanian and Danny McPherson, “The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets”, USENIX Association, SRUTI ’05: Steps to Reducing Unwanted Traffic on the Internet Workshop pp. 39-44, 2005.
15. W. Tarng, L.-Z. Den, K.-L. Ou, and M. Chen, "The Analysis and Identification of P2P Botnet’s Traffic Flows", International Journal of Communication Networks and Information Security (IJCNIS), vol. 3, pp. 138.148, 2011.
16. K. Muthumanickam and E. Ilavarasan, "P2P Botnet detection: Combined host-and network-level analysis", International Conference on Computing Communication & Networking Technologies (ICCCNT), IEEE, pp. 1-5, 2012.
17. S. S. Silva, R. M. Silva, R. C. Pinto, and R. M. Salles, "Botnets: A survey", Computer Networks Journal, Elsevier, vol. 57, pp. 378-403, 2013.
18. S. Khattak, N. Ramay, K. Khan, A. Syed, and S. Khayam, "A Taxonomy of Botnet Behavior, Detection and Defense", Communications Surveys & Tutorials, IEEE, Vol. 16, pp. 898-924, 2013.
19. J. Govil and J. Govil, "Criminology of botnets and their detection and defense methods", International Conference on Electro/Information Technology, IEEE, pp. 215-220, 2007.
20. H. R. Zeidanloo, M. J. Z. Shooshtari, P. V. Amoli, M. Safari, and M. Zamani, "A taxonomy of botnet detection techniques", International Conference on Computer Science and Information Technology (ICCSIT), IEEE, pp. 158-162, 2010.
21. Kumar, S. (2022). Strategic management of carbon footprint using carbon collectible non-fungible tokens (NFTS) on blockchain. Academy of Strategic Management Journal, 21(S3), 1-10
22. Kumar, S. (2021). Review of geothermal energy as an alternate energy source for Bitcoin mining. Journal of Economics and Economic Education Research, 23(1), 1-12

23. Dr. Naveen Nandal, Dr. Aarushi Kataria, Dr. Meenakshi Dhingra. (2020). Measuring Innovation: Challenges and Best Practices. International Journal of Advanced Science and Technology, 29(5s), 1275 - 1285.

24. Nandal, N. Impact of product innovation on the financial performance of the selected organizations: A study in indian context. Psychol. Educ. J. 2021, 58, 5152–5163.

25. P. Barthakur, M. Dahal, and M. K. Ghose, "An Efficient Machine Learning Based Classification Scheme for Detecting Distributed Command & Control Traffic of P2P Botnets", International Journal of Modern Education & Computer Science, MECS Publications, vol. 5, pp. 9-18, 2013.
26. J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets", International Conference on Availability, Reliability and Security, IEEE, pp. 917-922, 2009.
27. S. Garg, A. K. Sarje, and S. K. Peddoju "Improved Detection of P2P Botnets through Network Behavior Analysis", Recent Trends in Computer Networks and Distributed Systems Security, Springer, pp. 334-345, 2014.
28. P. Narang, S. Ray, C. Hota, and V. Venkatakrishnan, "Peer Shark: Detecting Peer-to-Peer Botnets by Tracking Conversations", IEEE Security and Privacy Workshops, IEEE, pp. 108-115, 2014.

Authors

Dr. Nitya Nand Dwivedi,
Assistant Professor, Department of Computer Application
Swami Vivekanand Subharti University,
Meerut, Uttar Pradesh, 250005
Email: nityananddwivedi29@gmail.com,

Dr. Shashiraj Teotia
Associate Professor, Department of Computer Application
Swami Vivekanand Subharti University
Meerut, Uttar Pradesh, 250005
Email:shashirajt@gmail.com

Mr. Ankur Chaudhary
Assistant Professor, Department of Computer Application,
Swami Vivekanand Subharti University,
Meerut, Uttar Pradesh, 250005
Email:ankurchaudhary849@gmail.com