International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

NetFlow based Cyber Threat Classification using J48 and Random Forest Machine Learning Algorithms

Rakesh Kumar, Rajeev Singh

Abstract*: In the field of information technology cyber security plays a vital role. Securing information is the biggest challenge now a days. As the word cyber security comes in our mind the fear of cybercrime comes in us at the same time. Cyber threats are nothing but an activity by which any targeted system can be compromised by altering the availability, integrity, and confidentiality of the system. To overcome such type of threats there are number of mechanisms available. Recently the Machine Learning (ML) approaches have proved to be a milestone for the classification of NetFlows. The NetFlow is a network protocol designed by CISCO which is used to collect the network traffic (NetFlows). In this paper J48 and Random Forest (RF) machine learning algorithms are used for classification of cyber threats using NetFlows. The results are obtained by applying classification algorithms on NetFlows using Weka ML tool and RStudio. A comparison is made in various perspectives like accuracy, true positive (TP), false positive (FP), etc.

Keywords* : Classification Algorithms, J48, Machine Learning, NetFlows, Random Forest.

I. INTRODUCTION

As the attackers become more and more smart the security of data decreases. Honeypots are the network enabled systems where the attacker attacks and system collect information like way of attacks, type of attack and other information related to attacker. The honeypot represents itself as a targeted system for hackers, and the hackers try to negotiate the honeypot installed system by the unauthorized access. The honeypots can be categorized based on the level of interaction between the intruder and system named as low-interaction honeypot, medium-interaction honeypot, and high-interaction honeypots. The low-interaction honeypot has the limited bandwidth between the user and external system, the medium-interaction honeypot lies between the low-interaction and high-interaction honeypots, while the high-interaction honeypot provides the better experience for the attackers and itself gathers more information of the specific attack [1]. After getting the information of attacker, the host can classify them by applying various techniques and in this research machine learning (ML) mechanisms have been used to categorize the recorded NetFlows. ML is a part of artificial intelligence (AI) by which we can obtain predictions or make classification as per the given data [2]. It is a technique by which we can analyze the large volume of data of NetFlows classification. In this paper we have used three ML techniques named as J48 and RF decision tree algorithms. Basically, there are three approaches of ML, supervised ML, Unsupervised ML and Reinforcement learning.

A. Supervised Learning

It is a kind of machine learning approach that makes use of a known dataset for training/learning and build classifier model which is later used for predicting class labels. The training data includes input features (X) and output class labels (Y). Using the data provided for training a supervised learning algorithm builds a model that can predict output class labels (Y) for a new dataset (testing data) which is used to evaluate the accuracy of the model. Examples of supervised machine learning algorithms are: Decision Tree classifier, Support Vector Machine, Random Forest ensembles classifier etc.

B. Unsupervised Learning

Most Machine Learning systems learn from labelled instances, nevertheless it is also possible to learn from unlabelled objects but it difficult to do so. Such an approach is called unsupervised learning. The most popular approach of generalizing unlabelled instances is conceptual clustering, where clustering is the task of grouping a set of objects on the basis of similarity of the objects. Examples of unsupervised machine learning algorithms are: K-means, Hierarchical clustering etc.

C. Reinforcement Learning

Machine is trained to make decisive actions. The machine is exposed to an environment where it trains itself indefinitely using trial and error, and learns which actions yield the best rewards. Examples of Reinforcement Learning machine algorithms are: Greedy optimization algorithm and LTV (lifetime values) optimization algorithms.

Among the various ML algorithms, we have selected of J48 and RF are used to classify the NetFlows. Both are the supervised decision tree algorithms.

The NetFlow is a protocol designed by the CISCO is used to collect and record all IP traffic to and from a router which is NetFlow enabled. This protocol permits to collect and examine data traffic via a program. It permits us to really drill into our network traffic to locate how the traffic is coming from the source address and where is the destination of that traffic. The NetFlow was made up of two components: NetFlow cache and NetFlow export.

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Rakesh Kumar*, Computer Engineering Dept, College of Technology Pantngar, U. S. Nagar, India. Email: rakesh.patel319@gmail.com

Rajeev Singh, Computer Engineering Dept, College of Technology Pantngar, U. S. Nagar, India. Email: rajeevpec@gmail.com

The NetFlow cache collects the IP flow info and NetFlow export forward data for analysis. The datasets are CTU-13 dataset generated at CTU University, Prague, Czech Republic [3], and the PantHoney dataset which was collected at Pantnagar University. The motive of this work is to classify the NetFlow by J48 and RF ML algorithms and check the performance of these algorithms on different platforms. The various stages of NetFlow classification are:

D. Data Acquisition

Data acquisition deals with obtaining and capturing information from two sources. The source is CTU-13 dataset generated in 2011 with the help of IDS and created 13 scenarios to represent the whole dataset. The second dataset is PantHoney dataset generated in 2019, through a HoneyNet installed at G.B.P.U.A&T.

E. Data Pre-Processing

The CTU-13 dataset is already labelled dataset whereas the PantHoney dataset is labelled with the help of alert files. Generally, binetflow files are not labelled files so the binetflow files can be labelled in Excel sheet.

F. Data Segmentation

Segmentation of the data means dividing the data into various parts according to need. Here we have taken two parts from CTU-13 dataset as scenario one and two respectively for the compilation of our research work respectively. Similarly, we have acquired two different parts from PantHoney dataset as scenario three and four respectively for the validation of proposed our work.

G. NetFlow Classification

Finally, classifiers are used for the training and testing of the datasets. These methods are used to classify NetFlows. Some prominent classifiers are J48, random forest, support vector machine (SVM), k-nearest neighbor, multi-layer perceptron classifier. This paper is further subdivided into V sections. Section II discusses about background work i.e. NetFlow classification, Section III methodology used for the classification of NetFlows. Section IV Evaluates the dataset and analyse the ML algorithms, Section V provides discussion and finally Section VI provides the conclusion.

II. RELATED WORK

A number of researchers have used the CTU-13 dataset for detection of botnets, the same dataset has been used in this research work. The attacks were classified by the RF feature extraction algorithm. Tomasz et al. (2014) proposed an anomaly detection system with the help of ARFIMA (Autoregressive Fractionally Integrated Moving Average) model. Dynamically growth of network security is required for protection against new threats. Intrusion Detection/Prevention Systems (IDS/IPS) use ARFIMA model for the detection of anomalies therefore IDS can be used for detection and protection against new cyber threats. The function of IDS and IPS is used to monitoring and detecting the attacks. The important function of IDS is not only monitor and detect bugs present in the system but also keeping the record of events trying to disrupt the security. The IDS is classified into two groups. The first group is used for detecting the known attacks with the use of determined and precise features called signatures while the second group uses a technique for monitoring the activities of the system in order to expose the anomalies for detection of attacks. The DARPA dataset was used by the authors and after applying the ARFIMA model they got a detection rate which lies between 95%-100% [4]. Wagner and Engel (2012) proposed a kernel function method for the detection of anomaly from the NetFlow records. In this, the first phase was based on the spatial aggregation technique which shows the NetFlow records and in second phase the authors have applied the kernel function on NetFlow data to detect the anomalies from the records. In this, authors collect the data from RESTENA Luxembourg which is acquired using five local ISP's. The data was used by the author for detection of UDP-flooding attack [5]. According to Duygu et al. (2017) the big data apache spark method can be used to classify the anomalies from NetFlow data. The continuous changes in the network can be easily analysed via big data because of its six V's characteristics. The Velocity i.e. how fast the data is processing, second Volume i.e. how much data is consumed, third Variety of data, fourth Veracity i.e. accuracy of data, fifth Vocabulary i.e. schema and structure of the data and sixth Value which dictates the importance of data. The authors have selected the 10th sample of CTU-13 botnet dataset. From this part of the data, outputs were generated to find the botnet attacks. After applying the big data technique for the classification of NetFlow data an accuracy of 96% is achieved [6]. Cynthia et al. (2011) proposed an OCSVM (One Class Support Vector Machine) ML approach for anomaly detection. The Lincoln dataset was used in this research work, which includes the Nachi scan, Netbios scan, DDoS UDP flood, DDoS TCP flood, Popup spam malicious activities. To overcome such type of anomalies the author used OCSVM and got a 92.8% [7]. Ahmad et al. (2016) developed a machine learning (ML) technique for botnet traffic classification using a C4.5 and Correlation-based Feature Selection algorithms. By using the Zeus botnet (Microsoft Windows) threats are collected such as spam, distributed denial-of-service (DDoS), and phishing. At present most of the botnet detection techniques are not feasible because the bots change their C&C server structure. To address this issue the author proposed a technique using CONFIA (Classification of Network Information Flow Analysis) and C4.5 algorithms. This work successfully achieved the target of detecting botnet attacks in the network [8]. Valentin et al. (2011) proposed a method to the classify network traffic by using machine learning (ML) approach such as C4.5 algorithm. For the classification of NetFlows the author collected the data form Universitat Politecnica de Catalunya (UPC) and used C4.5 ML technique for validation process. Then using the J4.5 decision tree algorithm the author achieved an accuracy of about 90% [9]. Bakshi and Ghita (2016) proposed a two-phase machine learning approach for the classification of network traffic. The flow accounting mechanisms like NetFlows are assumed insufficient for classification requiring more packet-level information like host behavior, and particular hardware. So, to overcome the classification problem of NetFlows, the two-phase ML classification mechanism is applied on NetFlow inputs. In this work the K-means is applied on every flow class and used for training in C5.0 decision tree classifier.

The results obtained by using unsupervised ML techniques on 6.8 million NetFlow records the accuracy as 87.67%, and after comprising the 14 NetFlow attributes of dataset reported an average accuracy of 92.37% increasing to 96.67% with adaptive boosting [10].

Peichao et al. (2018) presented a work on NetFlow based data to investigate network behavior by using classification techniques. The information of entropy is used to outline the features of relations among the hosts and ports. Classification was used for analysing the network traffic. The author developed a setup for capturing NetFlows and used the obtained data of 17 days from 661 hosts. The whole dataset was divided into groups where every group has unique behavior patterns. Use of classification with NetFlow data is a better option because the dataset is already divided into several groups. The authors have used the random forest method for the classification of network based twelve featured datasets on Weka 3.8 version with 10-fold cross-validation. In 10-fold cross validation, the 9-folds are used as training and the 10th-fold is used for the validation [11].

Kieran et al. (2017) presented a cluster density mechanism for anomaly detection using NetFlow data. Analyzing malicious activities is a challenging task because of continuous changing of the property of attacks. Anomaly detection mechanisms such as cluster density mechanism are covered in this work because they are beneficial as they can analyse the change in the network and threats without any intervention. Even though anomaly detection mechanisms have significant potential, but there are still some limitations for example a number of anomaly detection methods are not suitable for the real time environment. The authors have used k-NN (k-nearest Neighbor) and MCOD (Micro Cluster Based Algorithm) algorithms for anomaly detection [12].

Bakhshandeh and Eskandari (2018) proposed a random forest (RF) machine learning (ML) approach for NetFlow analysis. A number of methods are there for user profiling using various data sources like logs of the web traffic or packet traffic of the network. These methods are not practically feasible because they have not the capacity to process a large amount of traffic. The authors have used data captured from their own setup and after applying the RF technique got the accuracy of 94.60% [13]. Jiangpan et al. (2018) proposed a machine learning approach for DDoS attack detection using NetFlow analysis. DDoS is a dangerous threat to the internet. The methods used for detection usually fail because of their limitations in real-time, difficulty or universality. For detecting various types of DDoS attacks traffic sampling database like NetFlows are used. The authors have applied the C4.5, SVM (Support Vector Machine), Adaboost, and Random Forest machine learning algorithms for the detection of DDoS attacks on NetFlows and the detection accuracy by these algorithms is 0.930%, 0.908%,0.981%, and 0.986% respectively, the average accuracy of all the classification mechanism is more than 99%. [14].

III. METHODOLOGY USED

The CTU-13 dataset and PantHoney dataset contain the NetFlow files and the NetFlow files are labelled using the alert files. Then the machine learning algorithms are applied on labelled NetFlow files. J48 and RF algorithms are executed in Weka and RStudio environment.

In this paper ML algorithms J48 and Random Forest (RF) are used for the classification of NetFlows.

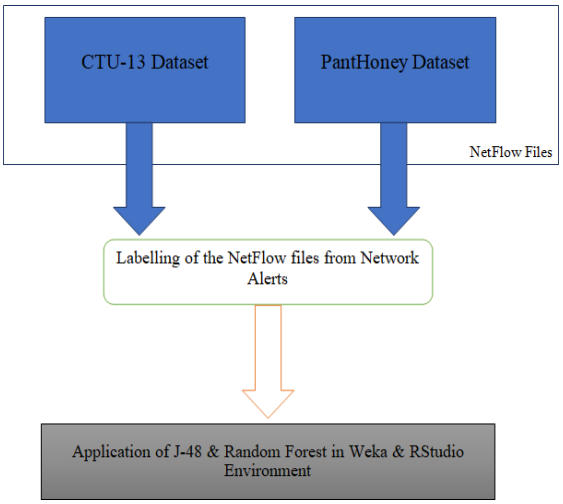

Fig. 1 General Architecture of NetFlow Analysis

The J48 decision tree classification algorithm uses labelled data in its beginning phase (training set). In training phase, the given data is partitioned into resulting nodes and follow the recursive divide and conquer mechanism. The RF emerges from decision tree classifier which is an ensemble method and it develops tree using CARET (Classification and REgression Trees) mechanism.

The decision tree is decision structure which uses a tree like representation of graph to make the decisions. The decision tree is also known as classification tree where each internal node denotes an attribute while the branches represents to the outcome of the test and in last every leaf node depicts to a class label [15] [16].

A. J48

J48 is an extension of ID3 (Iterative Dichotomiser 3) which was developed by Ross Quinlan and first time proposed in WEKA (Waikato Environment for Knowledge Analysis). The specializations of J48 are accounting for hidden values, decision tree pruning, continuous attribute value ranges, and derivation of rules [17]. WEKA uses the java library and accepts a greedy and top-down approach for the creation of decision tree. The J48 decision tree classification algorithm uses labelled data in its beginning phase (training set). In training phase, the given data is partitioned into resulting nodes and follow the recursive divide and conquer mechanism [18]. We have followed the 10-fold cross validation mechanism to divide the whole data set [19]. In 10-folds cross validation 9-folds assumed as training set and 1-folds as test set.

B. Random Forest

The idea of Random Forest (RF) first time was discussed by Ho in 1995. The RF emerges from decision tree classifier which is an ensemble method, it develops tree using CART (Classification And Regression Trees) mechanism for maximizing size without pruning. So, it is capable for both classification and regression tasks. RF provides the better results from growing ensemble of tress. As the name introduces it creates a forest with the number of decision tress. In general, the more trees in forest the more robust prediction and thus higher accuracy [20].

Random forest is one of the most prevailing supervised ML algorithm. Like the name says it has a lot of individual decision trees that operated as an ensemble classifier. Every tree of the forest serves a division by which we can classify the new instances.

C. Dataset Description

The datasets are used in this research named as CTU-13 and PantHoney (table 1). In table 1 case 1 and 2 considers data from CTU-13 dataset, corresponding to the data captured on 11-08-2011 and 15-08-2011. Case 3 and 4 considers data collected through PantHoney dataset captured on 05-02-2019 and 25-05-2019. Case 1 shows the 2nd scenario of the CTU_13 dataset while the second case represents 3rd scenario of CTU_13 dataset. Case 3 and 4 denotes 1st and 2nd scenario of PantHoney dataset.

Table I: Dataset Description

Dataset Description				
Cases	*Dataset Name*	*Duration*	*IP*	*No. of Instances*
1	CTU_13_1	00:50:59	147.32.84.165	2000
2	CTU_13_2	00:49:15	147.32.84.165	2000
3	PANT_File_1	00.58.08	14.139.250.XXX	2000
4	PANT_File_2	00:59:02	14.139.250.XXX	2000

Table II: Detailed Description of Dataset

Sr. No.	Attribute	Description
I	StartTime	Start time of the recorded NetFlow
II	Dur	Duration
III	Proto	IP protocol
IV	ScrAddr	Source address
V	Sport	Source port
VI	Dir	Direction of recorded communication
VII	DstAddr	Destination address
VIII	Dport	Destination port
IX	State	Protocol state
X	sTos	Source type of service
XI	dTos	destination type of service
XII	TotPkts	Total number of packets that have been exchanged between source and destination
XIII	TotBytes	Total bytes exchanged
XIV	SrcBytes	Number of bytes sent by source
XV	Label	Label assigned to this NetFlow (e.g., background, normal, and botnet)

Both the datasets have same attributes for evaluating the ML algorithms. As we have applied cross validation, so it divides the dataset into training and testing set. The data containing Binetflow file can be converted into CSV format easily. Both datasets have various rows and their attributes are listed in table 2. We calculated these parameters with the help of our dataset mentioned in table 1.

In this paper ML algorithms J48 and Random Forest (RF) are used for the classification of NetFlows. The J48 decision tree classification algorithm uses labelled data in its beginning phase (training set). In training phase, the given data is partitioned into resulting nodes and follow the recursive divide and conquer mechanism. The RF emerges from decision tree classifier which is an ensemble method and it develops tree using CARET (Classification and REgression Trees) mechanism.

IV. EXPERIMENTS AND RESULTS

In our setup we have used two datasets one to test and other to validate the experimental setup respectively. Various parameters mentioned in table 3 have been analyzed with the help of these two datasets:

A. True Positive (TP) Rate

The threats sample which are predicted as threat and they were really threat.

B. True Negative (TN) Rate

The threats sample which are predicted as non- threat and they were really non- threat.

C. False Positive (FP) Rate

The threats sample which are predicted as threat but they were not threat.

D. False Negative (FN) Rate

The threats sample which are predicted as non- threat but they were accurately threat.

Table III: Confusion Matrix

	Predicted: YES	**Predicted: NO**
Actual: YES	TP	FN
Actual: NO	FP	TN

Some of the important parameters required to evaluate the NetFlows are:

E) Precision

Precision is the measure of how often the model predicted yes and it is correct. It can be calculated by the given formula:

$$Precision = \frac{TP}{TP + FP}$$

F) Accuracy

The accuracy of a measurement is how close a result comes to the true value.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

G) Error rate

The error rate is calculated as the number of all incorrect classified instances divided by total number of instances. The best error rate is 0.0, while the worst is 1.0.

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN}$$

h) F1 measure

The f1 score can be understood as a biased average of the precision and recall, where an f1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the f1 score are equal. The formula for the f1 score is:

$$F - measure = 2 * \frac{Precision * Recall}{Precision + Recall}$$

A) Correctly Classified Instance

In figure 2, there are two types of points stars and squares on the classifier window. The star shows correctly classified instances while square displays incorrectly classified instances. This figure denotes to the information of 82 instance and it is classified as correctly.

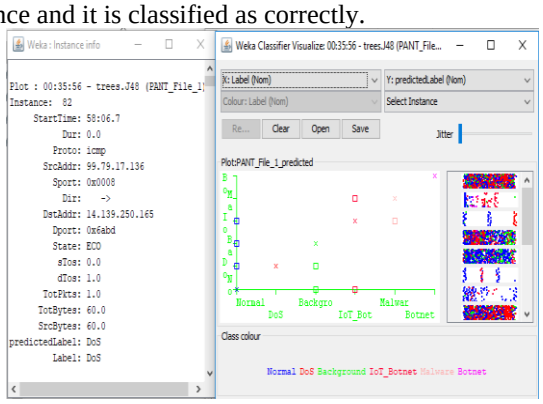

Fig. 2 Correctly Classified Instance

B) Incorrectly Classified Instance

Figure 3 shows the incredibly classified instance 568 and is derived from PANT_File_1.

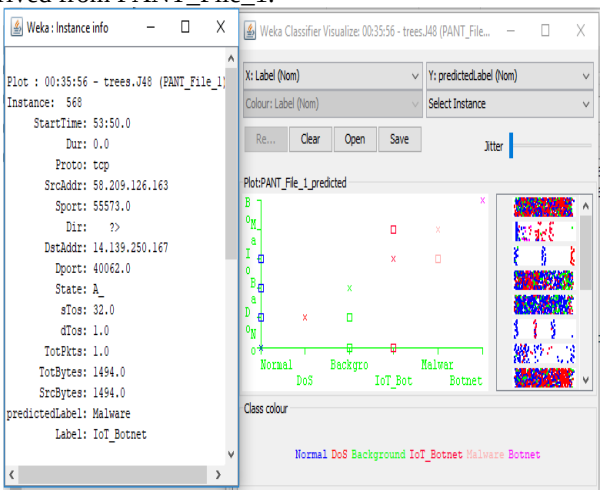

Fig. 3 Incorrectly Classified Instance

C) ROC Curve

Figure 4 shows the ROC (Receiver Operating Charateristics) cuve. ROC curve defines how the data is correctly classified. It is the classification between true positive (sensitivity) and false positive (specificity). As the figure 3 closer towards to left-hand edge and after that top edge represents a better classification. If any of the ROC curve derive to 45-degree diagonal space, it represents less accurate test.

D) Decision Tree

Figure 5 depicts the decision tree regarding the J48 classification ML algorithm. In figure 4 there are 9 leaves and 17 trees, the decision tree can be classified on the basis of Proto attribute as table 1 root. It makes two decisions according to the Proto attribute tcp or not tcp and further divides into ScrAddr and Dir attribute. Now both the attribute further classified into yes or no and so on. The leaves nodes elaborate to the classified label attribute.

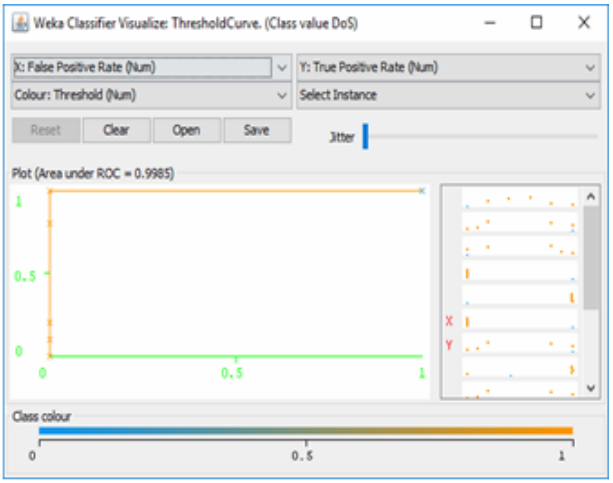

Fig. 4 ROC Curve

Table 4 shows the results obtained after executing J48 algorithm on Weka. In which 4 cases have been considered, first two cases are from the CTU-13 and last two from the PantHoney dataset. The average accuracy obtained after evaluating J48 algorithm on the CTU-13 dataset cases is 96.775% while the average accuracy of PantHoney dataset is 98.175%. The values of TPR, FPR, Precision are almost similar for all the cases.

The results presented in table 5 are obtained after running the Random Forest algorithm using Weka tool on the datasets low accuracy is observed for all the cases of both the datasets. The average accuracy of the algorithm on CTU-13 and PantHoney dataset is 90.5% and 95.95% respectively. The accuracy of RF is less than the accuracy of RF algorithm.

Table 6 shows the results of using J48 algorithm on RStudio. The average accuracy of the J48 algorithm for all the cases CTU-13 dataset is 94.82% and PantHoney dataset is 93.2% which is less than the accuracy achieved on Weka classifier for the J48 algorithm.

V. DISCUSSION

The CTU-13 and PantHoney datasets have been used in this research. There are two samples taken from PantHoney dataset. The PANT_File_1 corresponds to Normal, DoS, IoT_Botnet, Botnet and Malware, while the PANT_File_2 consists Botnet, IoT_Botnet, Background and Malware cyber threat activities of NetFlows. The following points are worth mentioning:

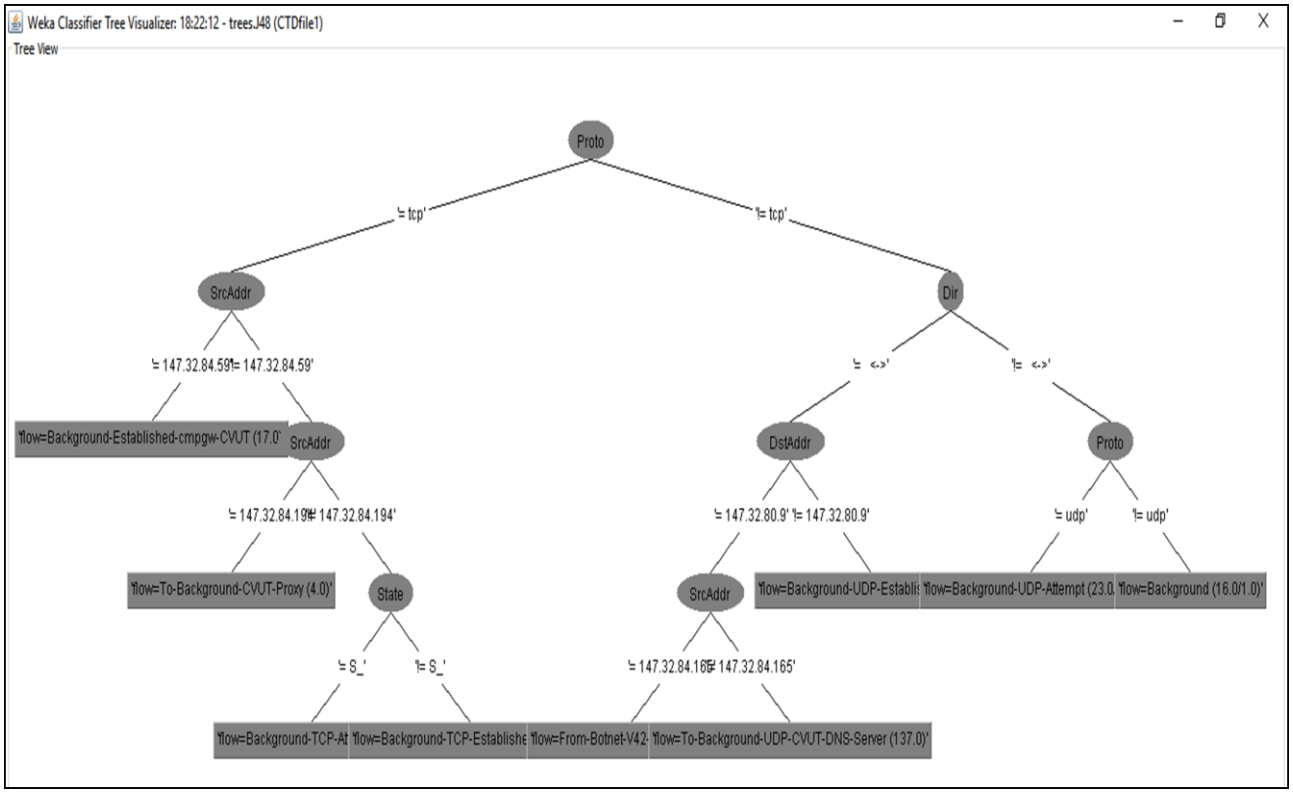

Fig. 5 Decision Tree

Table IV: Results of Running J48 Algorithm on Weka

CTU-13 Dataset and PantHoney						
Dataset/Algorithm	TPR	FPR	Precision	Accuracy	Error rate	F-measure
CTU_13_1-J48	0.958	0.013	0.947	0.9575	0.0425	0.952
CTU_13_2-J48	0.978	0.003	0.969	0.978	0.022	0.973
PANT_F_1-J48	0.967	0.019	0.968	0.967	0.033	0.966
PANT_F_2-J48	0.997	0.001	0.997	0.9965	0.035	0.989

Table V: Results of Running J48 Algorithm on Weka

CTU-13 Dataset and PantHoney						
Dataset/Algorithm	TPR	FPR	Precision	Accuracy	Error rate	F-measure
CTU_13_1-J48	0.90	0.058	0.891	0.90	0.10	0.889
CTU_13_2-J48	0.904	0.034	0.898	0.91	0.090	0.897
PANT_F_1-J48	0.904	0.034	0.898	0.934	0.066	0.897
PANT_F_2-J48	0.985	0.046	0.981	0.985	0.015	0.982

Table VI: Results of Running J48 Algorithm on RStudio

CTU-13 Dataset and PantHoney						
Dataset/Algorithm	TPR	FPR	Precision	Accuracy	Error rate	F-measure
CTU_13_1-J48	0.92	0.018	0.833	0.9284	0.072	0.911
CTU_13_2-J48	0.95	0.003	0.777	0.968	0.032	0.954
PANT_F_1-J48	0.91	0.003	1.00	0.8665	0.1335	0.899
PANT_F_2-J48	0.96	0.006	0.95	0.9975	0.025	0.901

After observing the results from the Weka and R Studio tools it is concluded that Weka is better compared to R Studio for classification of NetFlows.

The J48 and RF algorithms were used on CTU-13 and PantHoney dataset for classification. The results are validated using the PantHoney dataset using the same classification techniques which have been applied on CTU-13 dataset.

In the Weka tool, the accuracy of J48 is more than the RF algorithm because J48 creates the tree on the basis of single parameter while the RF is an ensemble classifier and creates a number of trees on the basis of multiple parameters.

The J48 decision tree classification algorithm is applied in Weka and RStudio environment, in terms of accuracy the results observed from the Weka classifier tool are better in comparison to R Studio for J48 algorithm.

J48 classification algorithm works in R Studio for both the datasets but the Random Forest works only in Weka tool but not in RStudio. Upon executing RF in RStudio it has 53 categorical/labels problem. The RF algorithm in RStudio has a limitation which prevents it for classifying data having greater than 35 categorical values. If any of the dataset have such type of problem, then there is no solution of this categorical problem because it is the limitation of Random Forest in RStudio.

VI. CONCLUSION

In this research work we have used CTU-13 dataset and is validated using PantHoney dataset. For classifying the cyber threats, J48 and Random Forest machine learning decision tree classifier techniques have been used. The CTU-13 dataset and PantHoney dataset have been analyzed on different platforms and it has been found that the overall accuracy on the Weka classifier is more than the RStudio platform. The average accuracy of the J48 decision tree algorithm achieved on Weka and RStudio platforms are 97.475% and 94.01% respectively. The Random Forest classifier gives 93.225% on Weka tool. It is evident that J48 and RF have different accuracy on different platforms. It is found that the Weka tool provides better performance than RStudio: precision value and f1 Score are also high. While executing RF on RStudio, RStudio did not execute due the number of categories exceeding 53 (53 categorical problem).

REFERENCES

1. Matri Shukla and Pranav Verma, "Honeypot: Concepts, Types and Working", IJEDR, issue 4, vol. 3, 2015, pp. 596-598.
2. Virendra Kumar, Munesh Chandra Trivedi, and B. M. Mehtre, "DDL: An Approach to Handle DDoS (Ping Flood) attack", proceedings of International Conference on ICT for Sustainable Development, Advances in Intelligent Systems and 408, 2016, pp. 11-23.
3. Rafal Kazil and Michal Choras, "Pattern Extraction Algorithm for Netflow-Based Botnet Activities Detection", Security and Communication Networks, 2017, pp.1-10.
4. Tomasz Andrysiak, Lukasz Saganowski, Michal Choras, and Rafal Kozik "Network Traffic Prediction and Anomaly Detection Based on ARFIMA Model", Springer International Publishing Switzerland 2014, pp. 545-554.
5. Cynthia Wagner and Thomas Engel., "Detecting anomalies in Netflow Record Time Series by Using a Kernel Function", IFIP International Federation for Information Processing, 2012, pp. 122-125.
6. Duygu Sinanc Terzi, Ramazan Terzi, and Seref Sargiroglu, "Big Data Analytics for Network Anomaly Detection from Netflow Data", 2nd International Conference on Computer Science Engineering, 2017, pp. 592-597.
7. Cynthia Wagner, Jerome Francois, Radu State, and Thomas Engel, "Machine Learning Approach for IP-Flow Record Anomaly Detection", IFIP International Federation for Information Processing, 2011, pp. 28-39.
8. Ahmad Azab, Mamoun Alazab, and Mahdi Aiash, "Machine Learning based Botnet Identification Traffic", IEEE TrustCom-BigDataSE-ISPA, 2016.
9. Valentin Carela-Espanol, Pere Barlet-Ros, Albert Cabellos-Aparicio, and Josep Sole-Pareta, "Analysis of the impact of sampling on NetFlow traffic classification", Computer Networks, 2011, pp. 1083-1099.
10. Taimur Bakshi and Bogdan Ghita, "On Internet Traffic Classification: A Two-phased Machine Learning Approach", Journal of Computer Networks and Communications, 2016, pp. 1-21.
11. Peichao Wang, Yun Zhou, Cheng Zhu, and Ruiqi Yue, "Role of Classification with Netflow Data in Internet", Tenth International Conference on Advanced Computational Intelligence (ICACI), 2018, pp. 29-31.
12. Kieran Flanagan, Enda Fallon, Abir Awad, and Paul Connolly, "Self-Configuring NetFlow anomaly Detection using Cluster Density Analysis", ICACAT, 2017, pp. 421-427.
13. Atieh Bakhshandeh and Zahra Eskandari., "An Efficient user Identification Approach Based on Netflow Analysis", 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC), 2018, pp. 1-5.
14. Jiangpan Hou, Peipei Fu, Zigang Cao, and Anlin Xu, "Machine Learning Based DDoS Detection Through NetFlow", in proceedings at Military Communication Conference (MILCOM), 2018.
15. Sujata Joshi and S. R. Priyanka Shetty, "Performance Analysis of Different Classification Methods in Data Mining for Diabetes Dataset Using WEKA Tool", International Journal in Recent and Innovation in Computing and Communication, issue 3, vol. 3, March 2015, pp. 1168-1173.
16. Neeraj Bhargava, Girja Sharma, Ritu Bhargawa, and Manish Mathuria, "Decision Tree Analysis on J48 Algorithm for Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering, issue 6, vol. 3, June 2013, pp. 1114-1119.
17. Gaganjot Kaur and Amit Chhabra, "Improved J48 Classification Algorithm for the Prediction of Diabetes", International Journal of Computer Applications, issue 22, vol. 98, July 2014, pp. 13-17.
18. Uzair Bashir and Manzoor Chachoo, "Performance Evaluation of J48 and Bayes Algorithms for Intrusion Detection System", International Journal of Network Security & Its Applications (IJNSA), issue 4, vol. 9, July 2017, pp. 1-11.
19. Akhilesh Kumar Shrivas and Prabhat Kumar Mishra, "Intrusion Detection System for Classification of Attacks with Cross Validation", International Journal of Engineering Science Innovation, issue 9, vol. 5, September 2016, pp. 21-24.
20. Eesha Goel and Abhilasha, "Random Forest: A Review", International Journal of Advanced Research in Computer Science and Software Engineering, issue 1, vol. 7, Janunary 2017, pp. 521-257.

AUTHOR PROFILE

Rakesh Kumar graduated from Rakshpal Bahadur College of Engineering and Technology, Bareilly in 2013 in Computer Science Engineering. He worked as Assistant Professor at Surajmal College of Engineering and Management, Kichha, Uttarakhand (India). He is pursuing his M.Tech. Degree from G. B. Pant University of Ag. & Technology, Uttarakhand. He is currently working as a Assistant Professor at Rajshree Institute of Management and Technology, Bareilly. His research area is Cyber Security.

Rajeev Singh is currently working as Associate Professor in the Department of Computer Engineering, G. B. Pant University, Uttarakhand (India). He received his Ph.D. Degree from N. I. T. Hamirpur (H. P.) and M. Tech. Degree from Indian Institute of Technology, Roorkee (India), both in Computer Science and Engineering. His research interest includes information systems, computer networks and network security. He has published several book chapters and research papers in journals/conferences of repute.