2016-06

IP prefix hijack detection using BGP connectivity monitoring

Alshamrani, H

10.1109/hpsr.2016.7525636
2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR)
IEEE

IP Prefix Hijack Detection Using BGP Connectivity Monitoring

Hussain Alshamrani
Centre for security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
hussain.alshamrani@plymouth.ac.uk

Bogdan Ghita
Centre for security, Communications and Network Research (CSCAN) Plymouth University
Plymouth, UK
bogdan.ghita@plymouth.ac.uk

Abstract— In spite of significant on-going research, the Border gateway protocol (BGP) still encompasses conceptual vulnerability issues regarding impersonating the ownership of IP prefixes for ASes (Autonomous Systems). In this context, a number of research studies focused on securing BGP through historical-based and statistical-based behavioural models. This paper suggests a novel method based on tracking the connectivity of suspicious ASes, which are received from a program tracing IP prefix hijacking signature. The paper uses Full Cross-Validation test to investigate the accuracy of the invented method and studies the similarity and differences between malicious and benign observations before they are classified. Classification might not be the appropriate technique to deal with IP prefix hijack detection on its own; therefore we propose to combine the two methods (signature and classification-based) in order to cover the limitations of both techniques. From a processing perspective, the outputs from signature-based method are used as inputs for the classification-based. The main features are extracted from the ASpath attributes of potentially suspicious ASes. The features are considered a mixture of the behavioural characteristics of connectivity among routers. The best five supervised classifiers were used in the previous researches and go with the characteristics of dataset will be used in this paper to evaluate the detection method. Under different learning algorithms, Random Forest and J48 classifiers, the detection method is able to detect the hijacks with 81% accuracy.

Keywords—BGP4; Machine learning; ASN; IP prefix hijack; features; RIRs Whois databases, route, MOAS, routes

I. INTRODUCTION

BGP remains the protocol of choice for core Internet interconnectivity. Although a number of BGP security issues have been identified for almost two decades, the protocol remains vulnerable to IP prefix attacks. This weakness leads to significant stability issues for the network, and may be used as a vehicle for black-hole traffic attackers [1], spamming [2], DDoS, and man-in-the-middle attacks [3]. In addition, hijackers may exploit redirecting BGP traffic for hijacking cryptocurrency transactions [4]. On April 2015 Schlamp pointed out to the reason that leads to hijacking of routes. For example, the main reason threatens the BGP security is emerging from abandoned Internet resources such as address blocks or AS numbers. In other words, when the DNS names expire, the attacker reregister domain names which are referenced by corresponding RIR (Regional Internet Registries) database objects [5]. 20% of the whole IPv4 address space is presently allocated but not above-board announced; this unused space is the ideal environment for such malicious BGP hijack events [3]. To solve this issue, our methods require organisations to announce their IP prefix at least once in order to advertise their ownership to the IP prefix block.

In a review of existing approaches, Goldberg indicated that the main reason BGP is taking so long to be secured is that, apart from its deployment challenges, the infrastructure lacks a central authority, as each organisation autonomously deploys its own solution, so a complete or mass deployment is unlikely to take place [6].

A traditional method employed by prior research has been to detect IP prefix hijacks based on anomaly detection and monitoring the stability of the encompassing routers. Nonetheless, such methods could not reliably distinguish IP prefix hijacks from normal events, such as power cut-off or submarine cable cuts [7]. Lastly, some detection methods analyse routing tables (table-based) in order to detect IP prefix hijacks, but organisations may refuse to provide their routing tables [8]. Vervier et al. noted that methods based on monitoring anomalies to detect IP prefix hijacks are still suffering from high false positive rates [3].

They also pointed out that prevention BGP hijack methods are still facing large-scale and deployment issues [3]. Due to several reasons, such as performance issues on large routing systems or impracticability of approaches like S-BGP [9], the threats still exist nowadays [10]. Wubbeling et al. pointed out security based on origin authentication and asymmetric encryption are not feasible nowadays, because it is not yet implemented in broadly used hardware and business processes of ASes [10]. In addition, RPKI (Resource Publication Infrastructure) system is one of IP prefix hijacking detection systems put in place to prevent BGP route hijacking. However the system had several false positives and negatives and needs further refinements. The system is based on tracing the hierarchical relationships of the address space were given by IANA, RIRs and big ISPs to customers. The Route Origin Authorizations (ROAs) is cryptographically signed and published in repositories. Every router has to upload the information [11].

As a case study, UPDATE messages were collected from the 24th of February 2008, using the Route View project of University of Oregon, when Pakistan Telecom intended to restrict local access to YouTube, but the advertised UPDATE messages blocked access to YouTube [12] for approximately two hours [13].

In this paper we implement a program to search for suspicious ASes and pass the result to another program to

978-1-4799-8950-8/16/$31.00 ©2016 IEEE

trace the behaviour of routers through their connectivity. From the behaviour we can extract several parameters such as direct and indirect neighbours, number of sender and receiver neighbours for both the victim and hijacker. These two programs form the structure of the detection method which is a combination of signature and connectivity-based. Zhang et al. pointed out the importance of signature-based and anomaly-based in modern intrusion detection together with their inherent drawbacks – uncertainty for signature-based methods and inability to detect new attacks for anomaly-based analysis [14]. Furthermore, connectivity model is a new approach used recently to trace the behaviour of opportunistic networks. Kathiravelu argues that a paradigm shift from mobility models connectivity model [15]. As a result, we decided to combine signature-based and anomaly-detection-based techniques to avoid their limitations when they work separately.

For the detection method validation purposes, we are going to use a number of supervised machine learning classifiers based on full cross-validation test technique. The highest accuracy of the hijack detection was achieved using J48 and RandomForest classifier where the accuracy reached 81%.

This paper is organised as follows: in section II we present the detection method of the IP prefix hijack. In section III we crosscheck the RIR Whois database with the outputs of validator to label incidents while in section IV we extract features based on the connectivity behaviour of suspicious routers. In section V we explore the similarity between suspicious and malicious observations before they are classified. Section VI discusses the methodology of the classification and testing the behaviour of suspicious ASes while VII evaluates the accuracy of the detection method based on the results of learning algorithms. The conclusions and future work are outlined in section VIII.

II. DETECTION METHOD

In this section, we talk about how to connect the detection method of new parts to the previous work. The detection method is going to add a novel features are proposed to use supervised machine learning algorithms to detect IP prefix hijacking. Thus, we need a supportive part to do labelling for data. Tracer and validator blocks are beyond the scope of this paper.

Machine leaning has different learning approaches to mine data such as supervised learning, semi-supervised learning, unsupervised learning, reinforcement learning and deep learning. However, the supervised-learning approach is more accurate and appropriate to the issue of impersonating others' IP prefixes issue; therefore, the dataset will be structured in supervised format.

The IP prefix hijack detection method is composed of five main parts as it is shown in figure 1: IP prefix hijack signature tracer, suspicious ASes validator, Labeller, Dataset Extractor and Organiser (DEO) and ML. However, this paper concern to only three blocks: the Model, ML and labeller part. Figure 1 shows the general structure of the detection method.

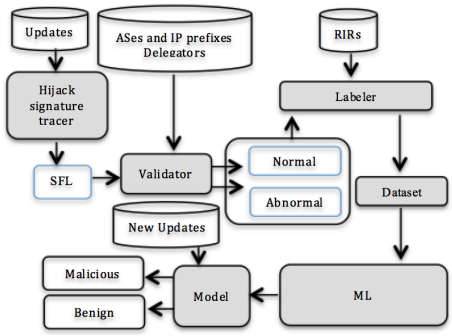

Fig 1. Detection method using combination of signature-based and connectivity-based

Tracer is signature-based algorithm receives update messages for specific period of time (15-minutes) and check them based on the IP prefix hijack signature. The algorithm uses two useful techniques data reduction and binary search algorithm to reduce search area of BGP messages. Table 1 shows the suspicious outputs the tracer caught. This table represents data were saved in the SFL (Suspicious Findings List), which exposes the output format of detected abnormal and suspicious routes.

Table1. Suspicious finding list

Announcers	Neighbours	Routes
AS3	AS1239	128.30.0.0/15
		18.168.0.0/24
		18.168.1.0/24
AS3292	AS1299	158.173.176.0/20
	AS3549	
	AS8001	

Validator receives suspicious ASes as inputs and verifies them based on the database generated from RIRs Whois [16] and ASNs (Autonomous System Numbers) and IP prefixes delegators. We do that because BGP updates do not support organisation name data and the same signature of the hijack is showing up in the normal behaviours of routers, this conflict is called MOAS (Multiple Origin Autonomous System).

In case of the ASes and IP addresses ownership are not updated regularly in RIRs and their delegators, Labeller receives the inputs from validator and labels the outputs of the validator because the detection method is based on supervised learning approach. Since the RIRs operators do not save their old subscribers records, finding out the ownership history of some nominated suspicious routers make it very difficult to label some ASes. This method helps to decide and collect behaviour only from known ASes and ensure from and separate the benign and malicious ASes.

DEO is responsible for extracting anomaly detection features, organising data and classifying the behaviour of nominated benign and suspicious ASes. The DEO has 9 features extracted based on the suspicious routers connectivity. It categorises suspicious routes into two classes either normal or abnormal. The outputs of the DEO are passed as inputs to the ML (Machine Learning) block.

In ML (Machine Learning) we use five learning algorithms to evaluate the proposed features. In this part we use full-cross validation test option for training and testing dataset. The ML will give the final result accuracy of the detection method and use the detection model for detection new hijacks.

III. LABELLING INCIDENTS

Sine RIRs (Regional Internet Registries) do not keep records of old Whois registrations details, this section intends to label the outputs of the validator in order to specify the ASes we are going to trace their behaviour during the hijacking history and then structure a very high accurate supervised learning dataset. Labeller still uses RIRs to build the dataset but it needs to filter confusing events that appear in the up to date Whois RIRs databases. Based on that, some nominated ASes were received from hijack signature tracer will be excluded from the outputs of the Validator as their ownership to the victim routes are ambiguous. Table 2 describes validator outputs before they are labelled.

Table 2. Validator outputs before labelling

AS1	AS2	IP prefix
3	27930	'190.14.196.0/24'
3	27930	'190.14.197.0/24'
37	27064	'198.91.71.0/24'
100	14807	'63.115.54.0/24'
100	14807	'65.204.11.0/24'
209	7018	'24.32.114.0/24'
209	2711	'64.53.21.0/24'
209	2711	'64.53.40.0/22'
209	6395	'66.212.81.0/24'

Each nominated suspicious AS is investigated based on the five regional registries: AfriNIC, APNIC, ARIN, LACNIC and RIPE NCC. The strategy of labelling the events is based on three main aspects:

- If both of suspicious ASes own the route, we mark them with OWNER, and then the event is benign.
- If one of suspicious ASes owns the route, it marks with ONER and HIJACKER, and then the event is malicious.
- If none of suspicious AS origins owns the route, we tag them with NOTSURE, and then the event is not labelled.

Table 3. Suspicious ASes investigator dataset

AS1	AS2	AS1 STATUS	AS2 STATUS	LABEL
100	250	OWNER	OWNER	BENIGN
200	10	ATTACKER	OWNER	MALICIOUS
300	50	NOT SURE	NOT SURE	AMBIGUOUS

AMBIGUOS events will be removed from the dataset and we only keep records were labelled as BENIGN or MALICIOUS as it shown in table 3. After extracting features as it is going to be in next section, each feature pattern will be given the class of its ASes event label.

IV. FEATURES EXTRACTION

In order to see the pollution of the internet when an edge router impersonates the ownership of a route is possessed by another router, and the connectivity between suspicious routers during the hijacking, we use Network Analysis and Visualization [17] to plot the topology of suspicious ASes. Based on the behaviour of suspicious ASes we extract 9 features from their connectivity. The behaviour of each suspicious AS can be calculated separately. However, we interested in the event of two suspicious ASes impersonating same IP prefix; therefore we need to take the absolute value of the differences between calculated suspicious ASes behaviours from equation 1. For example, finding the number of receiving neighbours is calculated in two separate column vectors, one for AS1 and another for AS2, and we need to apply the equation 1 in order to put them in one vector column. This vector column represents the behaviour of both ASes whether the event is malicious or benign. S_{AS1} and S_{AS2} indicate two sates either benign with benign or benign with malicious.

$$S_r = |S_{AS1} - S_{AS2}| \quad (1)$$

All features in table 4 were extracted from the behaviour of suspicious ASes (edge routers) are hidden in the ASPATH attribute. We briefly explain the purposes of these features. Since the innocent hijack does not occur for multiple different ASes, we extract the number of repeated incident in order to detect unintentional hijacks such as hijacks that occurred due to misconfiguration.

Table 4. Features of suspicious ASes

NO	Features
1.	# of repeated incidents
2.	# of receiver neighbours
3.	# of sender neighbours
4.	# of first propagators of suspicious routes
5.	# of shared receiver neighbours
6.	# of shared sender neighbours
7.	#of shared first propagators of suspicious routes
8.	# of connections between suspicious ASes
9.	Are they neighbours?

Generally, features 2-7 are based on the neighbourhood connectivity of suspicious ASes. Specifically, Features 2-4 concern about the direct neighbours of suspicious ASes while features 5-7 interests with shared direct neighbours between suspicious ASes. Feature eight and nine focus on direct and indirect connections between the suspicious ASes themselves. These features should reveal the similar and different patterns of suspicious ASes behaviours.

Table 5 shows a sample of the values of proposed features with their classes to detect the IP prefix hijacks. Each instance is labelled either with 0 if it is suspicious or 1 if it is benign. The type of pattern is represented by the whole of the features. F1-F9 represents features and C represents the two categorical classes of the behaviour. In terms of feature organization and calculation, each feature is saved in a separate column vector after being calculating based on the connectivity of suspicious edge routers. These column vectors are concatenated to give a dataset composed of 10 columns, including classes, and 340 examples. Since the registration details of some suspicious ASes are not recorded and are not given in any of RIRs, we

omit about 133 instances from the main dataset including malicious and benign samples. The dataset is built based on the rule explained in section III. The new size of the dataset will be dropped to have only 207 instances.

Table 5. Features after labelling

F1	F2	F3	F4	F5	F6	F7	F8	F9	C
2	22	1	665	0	1	0	0	0	0
7	0	1	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	1

Another important rule has to be taken into account is that getting rid of redundant instances, which means all repeated hijacks will be removed from the dataset because there is no need to similar events it. We observe that the size of the dataset is decreasing but with an increase in the accuracy of the data we are working on and getting rid of the redundancy. After labelling instances based on the RIRs Whois and removing repeated suspicious observation rules, the new size of the dataset will be limited to 113 instances. If the learning algorithms can distinguish the patterns of malicious and benign observations that mean the detection method was built in a high efficient way. Based on the results of the classifications we will evaluate the method.

V. CALCULATE DATA SIMILARITY

In this section we calculate the percentage of similarity and differences among benign and malicious observations. We invented our own algorithm to compute the similarity and differences of benign and malicious route patterns, which based on the XOR logical operator concept; the output is true if inputs are not alike otherwise the output is false.

Malicious and benign patterns are previously saved in one matrix. Malicious row observations are compared bit-by-bit against every benign sequence. X_b represents benign matrix row vectors and Y_m represents malicious matrix row vectors in formula 1 and 2.

$$X_b = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \quad (2)$$

$$Y_m = [f1, f2, f3, f4, f5, f6, f7, f8, f9] \quad (3)$$

Based on these two vectors formulas we compare the behaviour of benign and malicious observations. The output of this comparison is stored in a matrix with either zero or one, zeroes represent similarity and ones represent differences. We calculate every benign pattern and by the end we come up with several matrices for one benign vector, the number of matrices is as same as the size of malicious dataset.

Formula 4 show the general computation of similarity and difference of each benign pattern to all malicious patterns, where for is the loop starts from the first observation in the benign dataset and ends at the size of it. X_{b_i} is benign observations and Y_m is the malicious observations. $\sum 0$ is the summation of similar cases and $\sum 1$ is the summation of different cases.

$$_{i=1}^{n} for\{X_{b_i}\ xor\ Y_m\}_{(\sum 1)}^{(\sum 0)} \quad (4)$$

We calculate similarities and differences means of benign and malicious patterns from below two formulas 5 and 6, where n is the number of number of similarities and differences, S_i represents the similarities and D_i represents the differences of every benign observation to all malicious observations in the dataset. $\bar{S}$ gives the mean of similarity for all benign observations and $\bar{D}$ returns with the mean of differences of all benign observations. Both similarity and differences patterns of malicious and benign patterns are calculated to only ensure that the quality of data has been calculated correctly. In other words, one operation either calculating the behavioural similarity or difference between benign and malicious is enough to show the quality of data. Symmetric graph of similarity and differences shows that the calculation of one operation is carried out properly as it shown in figure 3. Based on observation of the calculation we either use similarity or difference calculation for studying the quality of the dataset. This dataset has malicious and benign announcement patterns.

$$\bar{S} = \frac{\sum_{i=1}^{n} S_i}{n} \quad (5)$$

$$\bar{D} = \frac{\sum_{i=1}^{n} D_i}{n} \quad (6)$$

From the graph in figure 3 we realise that the range of differences of malicious and benign observations is limited between 4.9 and 8.9. Correspondingly, the similarity among malicious and benign observations is limited between 2.9 and 6.1. If the value of both calculations is subtracted, the result will equal 4, which represents the range of similarities and differences. This value is probably is not very big but enough to differentiate between malicious and benign behaviours.

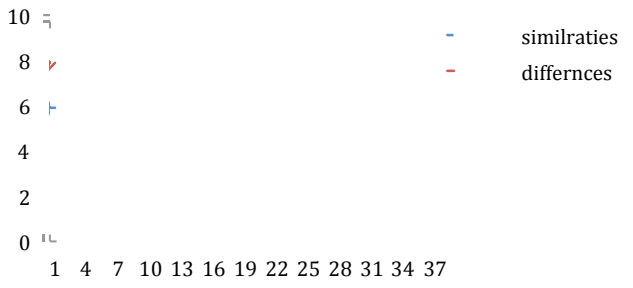

Fig 3. Similarity and differences between benign and malicious observations

Formula 7 computes the similarity percentage between malicious and benign behaviour, where total observations is equal to malicious observations plus benign observations. NS is the number of similarity was found in the whole dataset of benign and malicious observations. TO is the total observation of the dataset, which is 113. According to the formula, the percentage of the similarity is 0.07.

$$Similarity\ \text{percentage} = \frac{NS}{TO} \quad (7)$$

The more similarity behaviours are greater than the difference behaviours, the more confusion could happen to learning classifiers. Since the similarity between malicious and benign datasets is good, we can use different classifiers in section IX to differentiate two patterns.

VI. CLASSIFICATION

In this section we discuss the method is going to be used to apply machine learning with cross-validation test to detect the IP prefix hijacks. After building the dataset, which is based on the connectivity of the routers, we are going to classify suspicious ASes based on the following steps:

a) The detection method firstly determines the appropriate method of Cross-Validation test is going to be used.
b) Since data is few we use Full Cross-Validation with all proposed learning algorithms as in figure 2.
c) Each algorithm repeats the classification with different parameters for many times in order to observe the efficiency of the features and then the classifiers.
d) The best result of each classifier is saved to be compared to other classifiers' results.
e) Based on the offset of the number of false positives and false negatives, the best result among tried classifiers is determined.

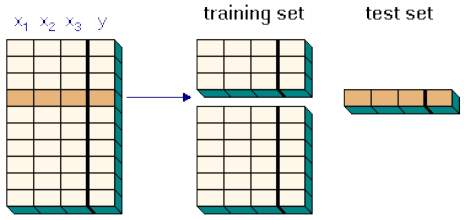

Fig 2. Full cross-validation technique

In 2008, Xindong Wu et al. had studded the best algorithms are using in data mining. Based on the research community it is found that the most influential data-mining algorithms are allocated in 10 top algorithms [18]. The study was about general types of learning algorithms such as classification, clustering, statistical learning, association analysis and link mining. However, we only interested in supervised-learning algorithms as we can have a prior picture of the percentage of benign and malicious data before they are given to the algorithms. On the other hand, another study has taken place for investigating the best learning classifiers in 2014. The study compared 179 classifiers for 17 families and over 121 different databases and found the best classifiers are Random Forest versions. RandomForest belongs to rule-based family and is considered supervised learning [19]. In addition, In 2014 Kaur and Chhabra claimed that improved J48 used recently to increase the accuracy rate of classification [20]. We have different deep studies of the best algorithms in data mining; first study was based on the research community and which the best algorithms were used widely in different area in data mining while the second study was empirical study performed by some experts in data mining. Both of studies are important because they cover each other limitations. Based on these studies we are going to test the detection method using five supervised learning algorithms: J48 which is the improved version of C4.5 and C5.0, k-NN (k-Nearest Neighbour), NB (Naïve Bayes), CART and RF (Random Forest); and based on the features of the algorithms such as accuracy, fastness and offset of false positive and negatives, the classifier is chosen. We also can observe that the most of the best classifiers belong to supervised not unsupervised learning. Although Adaboost is a supervised classifier one of the best learning algorithms, it is going to be excluded because of its dependability on other classifier. Adaboost strength is acquired from other classifiers, which means the algorithm gives the same accuracy result of the classifier it based on therefore it will be ignored.

A. Testing

Proposed algorithms use full cross-validation technique, which also called leave one out cross-validation. In full cross-validation, we choose the largest fold (113), which is the size of the dataset, in order to enlarge training dataset and minimize the size of the testing dataset, as the original dataset is not big. Every single instance will be used as a test set and remaining data as training dataset. This idea helps to avoid the possibility that folds (testing datasets) have one or more instances have not been trained in other folds (training datasets).

For instance, suppose we have 100 instances and we use 10 cross-validation, the dataset will be divided into 10 chunks because 100 divided by 10 is equal 10; so each one has 10 instances but probably the tested route malicious behaviour in the same fold of testing dataset. That means we might omit some hijacks if we do not maximize training dataset and minimize the test data set as much as we can. The smallest testing dataset and the largest training dataset we have, the more accurate evaluation of the detection method we receive. According to the changing of algorithms parameters the accuracy of the classification is registered as it shown in table 6. All algorithms parameters need to be adopted to suit the aim of the extracted features.

Table 6. Results based on Rule and Tree machine learning algorithms

Algorithm	Training dataset	Accuracy
J48	Full cross-validation	81%
KNN		79%
NB		76%
CART		81%
RF		81%

B. Error False Positive and Negatives Calculation

Confusion matrix in table 7 shows the accuracy of the detection method for both classes, malicious and benign. A and B represent correctly and incorrectly instances. Zero is the class of malicious instances and one is the class of benign instances. It is notable that the algorithms have difficulty to

classify benign observations more than classifying malicious observations in the whole algorithms with slightly better in k-Nearest Neighbours and Random forest.

For instance, for malicious classification, J48 classified data and came up with 3 incorrectly classified malicious observations and 73 correctly classified observations. The case repeats itself in k-Nearest Neighbours, Naïve Bayes and Classification and Regression Tree algorithms. However, Random Forest has 7 incorrectly classified malicious instances and 69 correctly classified malicious instances. In terms of benign classification, Random Forest is considered the best algorithm of detecting benign instances because it detected 23 benign instances correctly among 37 unique cases and the worst one is Naïve Bayes. Based on these notes and the offset of false positives and false negatives, the best algorithm will be elected.

Table 7. Confusion matrix testing for best classifiers

Algorithm	Train dataset	A	B	Classified as
J48	Full cross-validation	73	3	A=0
		18	19	B=1
KNN		73	3	A=0
		21	16	B=1
NB		73	3	A=0
		24	13	B=1
CART		73	3	A=0
		19	18	B=1
RF		69	7	A=0
		14	23	B=1

From two following equations, 8 and 9, we can compute the percentage error of the false positives and false negatives for the whole algorithms. NC_p Stands for Abnormal Confusion Percentage while AC_P represents Normal Confusion percentage.

$$NC_P = \frac{incorectly\ classified\ instnces}{correctly\ classified\ instnces} \quad (8)$$

$$AC_p = \frac{incorectly\ classified\ instnces}{correctly\ classified\ instnces} \quad (9)$$

If we take the percentage of false negatives and false positives for each algorithm and then the average of the whole algorithms, we come up 0.05 false negative and 1.15 false positives. Figure 3 shows that the false negative is less than the false positive in total but that does not explore the best algorithm; therefore, we judge the best algorithm based on taking the less false negative among malicious confusion computations and the less false positive in benign confusion.

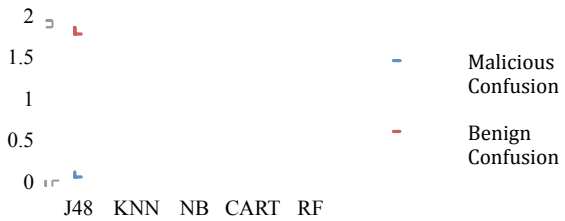

Fig 3. The best algorithm findings

From that rule we find the J48 is the best algorithm in terms of having less false negatives, if we take false positives of equal algorithms in to account. On the other hand, Random Forest is considered the best algorithm of detecting benign observations. If we take the offset of false positive and false negative of all classifiers results, Random Forest would be the best algorithm works with the features. However, the detection method would work better if we can combine these two algorithms to avoid learning implications for both algorithms. Formula 10 and 11 can compute the highest accuracy of the detection method when J48 and RF are combined, where HAM_m represents the highest accuracy of detecting malicious observations and HAM_b for detecting benign observations while $ICMO_{J48}$ and $ICBO_{RF}$ represent the incorrectly classified malicious and benign observations.

$$HAM_m = \left|\frac{ICMO_{J48} - ICMO_{RF}}{datset\ size}\right| \quad (10)$$

$$HAM_b = \left|\frac{ICBO_{J48} - ICBO_{RF}}{datset\ size}\right| \quad (11)$$

VII. RESULTS AND EVALUATION

Initially, the detection method was consisted of 12 features; these features are mixed of stability and connectivity observations of suspicious ASes were caught in tracing hijack signature phase. However, features were extracted based on the stability of edge routers are deleted as they make the detection very bad. As a result, the total number of features becomes 9. BGP packets are going to be classified and evaluated based on the remaining 9 features. Generally, the five classifiers are suggested to be used can work with the extracted features in a high efficiency although all of them have false positives and false negatives but in low percentage.

The detection method result supports the studies that have been investigated in 2014, and said that the Random Forest versions and J48 are the best algorithms among classifiers [19] [20]. The false negative if we use J48 is 0.02 while the false positive is 0.15. On the other hand, if we use Random Forest as the classifier of the detection method, the false negative will be 0.06 and false positive 0.12, which means J48 is better than Random Forest since the number of false negative in J48 is less than the number of false negative in RF. If we want the detection method to be in the highest efficiency, it needs to work with J48 and RF integrally. For example, based on equation 10 and 11, false negative will be 0.04 and 0.03 false positive accuracy, which means its accuracy will be increased from 81% to 93%.

VIII. CONCLUSION AND FUTURE WORK

In conclusion, this paper discussed a novel method to detect IP prefix hijacks in BGP. The method uses the extracted behaviour of suspicious ASes as inputs to the connectivity-based method, which in turn classify new BGP updates. Based on the suspicious ASes detected by the IP prefix hijacks Tracer data are classified into two classes, benign and malicious. Usually, researchers concern about the amount of

data in their datasets. However, we interest in the uniqueness of suspicious and abnormal patterns, therefore the amount of data was few. Another reason for making dataset small is that the algorithm excludes obvious normal observations from the dataset is going to be used for tracing routers connectivity. Moreover, there is no tool to give labelled accurate data of the historical incidents. As a result, we created our own accurate dataset by checking the ownership of suspicious ASes and IP prefixes through RIRs. Full Cross-Validation test method solves the issue of the size of the dataset because it is small. Five different learning algorithms and the best classifier works with the extracted features are picked. The result of the detection method is encouraging and very good as the percentage of false positive and false negative is less than 20% and the detection accuracy of the IP prefix hijacks is 81%.

REFRENCES

[1] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the internet," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, p. 265, Oct. 2007.

[2] P.-A. Vervier, Q. Jacquemart, J. Schlamp, O. Thonnard, G. Carle, G. Urvoy-Keller, E. Biersack, and M. Dacier, "Malicious BGP hijacks: Appearances can be deceiving," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 884–889.

[3] P. Vervier, O. Thonnard, and M. Dacier, "Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks," in *Proceedings 2015 Network and Distributed System Security Symposium*, 2015, no. February, pp. 8–11.

[4] G. Valadon and N. Vivet, "Detecting BGP hijacks in 2014 BGP Hijacking for Cryptocurrency Profit Reported by Dell SecureWorks on August 7 2014," 2014. [Online]. Available: http://www.nosuchcon.org/talks/2014/D3_04_Guillaume_Valadon_Nicolas_Vivet_detecting_BGP_hijacks.pdf.

[5] J. Schlamp, J. Gustafsson, M. Wählisch, T. C. Schmidt, and G. Carle, "The Abandoned Side of the Internet: Hijacking Internet Resources When Domain Names Expire," in *arXiv preprint arXiv: ...*, 2015, pp. 188–201.

[6] S. Goldberg, "Why is it taking so long to secure internet routing?," *Commun. ACM*, vol. 57, no. 10, pp. 56–63, Sep. 2014.

[7] I. O. de Urbina Cazenave, E. Kosluk, and M. C. Ganiz, "An anomaly detection framework for BGP," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.

[8] H. Cao, M. Wang, X. Wang, and P. Zhu, "A Packet-Based Anomaly Detection Model for Inter-domain Routing," in *2009 IEEE International Conference on Networking, Architecture, and Storage*, 2009, pp. 192–195.

[9] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 4, pp. 582–592, 2000.

[10] M. Wubbeling, T. Elsner, and M. Meier, "Inter-AS routing anomalies: Improved detection and classification," in *2014 6th International Conference On Cyber Conflict (CyCon 2014)*, 2014, pp. 223–238.

[11] M. Wählisch, O. Maennel, and T. C. Schmidt, "Towards detecting BGP route hijacking using the RPKI," in *ACM SIGCOMM Computer Communication Review*, 24-Sep-2012, vol. 42, no. 4, p. 103.

[12] H. Balakrishnan, *How YouTube was " Hijacked ,"* no. May. 2009.

[13] C. D. Marsan, "Six worst Internet routing attacks." [Online]. Available: http://www.networkworld.com/news/2009/011509-bgp-attacks.html. [Accessed: 28-Jan-2014].

[14] K. Zhang, A. Yen, X. Zhao, D. Massey, S. Felix Wu, and L. Zhang, "On Detection of Anomalous Routing Dynamics in BGP," in *Proceedings of the International IFIP-TC6 Networking Conference 2004*, vol. 5, 2004, pp. 259–270.

[15] T. Kathiravelu, A. Pears, and N. Ranasinghe, "Connectivity Models : A New Approach to Modeling Contacts in Opportunistic Networks," *Proc. Eighth Int. Inf. Technol. Conf. 2006*, p. 185, 2006.

[16] ARIN, "Regional Internet Registries," 2015. [Online]. Available: https://www.arin.net/knowledge/rirs.html. [Accessed: 08-Apr-2015].

[17] MathWorks, "Network Analysis and Visualisation," 2015. [Online]. Available: http://uk.mathworks.com/help/bioinfo/network-analysis-and-visualization.html. [Accessed: 10-Nov-2015].

[18] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, Jan. 2008.

[19] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?," *J. Mach. Learn. Res.*, vol. 15, pp. 3133–3181, 2014.

[20] G. Kaur and A. Chhabra, "Improved J48 Classification Algorithm for the Prediction of Diabetes," *Int. J. Comput. Appl.*, vol. 98, no. 22, pp. 13–17, 2014.