Border Gateway Protocol Anomaly Detection Using Neural Network

Mohsen Karimi†, Ali Jahanshahi‡, Abbas Mazloumi‡, Hadi Zamani Sabzi‡
Department of Electrical and Computer Engineering†, Department of Computer Science and Engineering‡
University of California, Riverside
{mkari007, ajaha004, amazl001, hzama001}@ucr.edu

***Abstract*—Having reliable and stable connectivity to the Internet dramatically depends on how Border Gateway Protocol (BGP) can avoid bad-behaviour events by detecting them on time. Despite a lot of efforts have gone into detecting BGP anomalies during the last decade, it is still a challenging issue due to emerging new abnormal behaviours both from the attackers and network misconfigurations. In this work, we propose a Neural Network classifier to detect the abnormal BGP events caused by worm attacks in the network. The results show that our method outperforms the previous work in both generality and accuracy.**

***Index Terms*—BGP, Anomaly Detection, Neural Networks, Machine Learning**

I. Introduction

Border Gateway Protocol (BGP) is the Internets default inter-domain routing protocol designed to exchange routing and reachability information among autonomous systems (AS) on the Internet. An AS is a set of routers under a single technical administration. Each router in an AS uses an Interior Gateway Protocol (IGP) to communicate with other routers within the AS and an Exterior Gateway Protocol (EGP) such as Border Gateway Protocol (BGP) to communicate with other ASes. The Border Gateway Protocol makes routing decisions based on paths, network policies, or rule-sets configured by a network administrator and is involved in making core routing decisions. Various abnormal events such as power outage, misconfiguration, and different types of attacks can affect the stability and reachability of some parts of the network which can lead to delays and data loss over the Internet. We refer these abnormal behaviors of the BGP protocol as anomalies. These anomalies can range from single to thousands of wrong BGP updates which can change the behavior of the BGP traffic over the time [1].

In recent years, in conjunction with anomalies such as hijacking, misconfiguration, link failure, and DoS attacks, it is reported that malware attacks such as Nimda and Slammer can affect on BGP routing as well [2]. However, there are still several anomalies have not reported or even unnoticed. The rapid growth of Internet size can lead to decrease of the stability of the Internet. For example, in 2018, 4,739 routing incidents have been disclosed by BGPmon, a popular monitoring service [3]. Thus, the problem of anomalies is still an important issue in having a reliable and stable connectivity to Internet.

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

BGP anomaly detection problem has been addressed from various aspects. Anomaly detection techniques can be evaluated based on different metrics such as their ability to cover different types of BGP anomalies (generality), their source cause of anomalies, BGP features used, and their accuracy. Generally, anomaly detection methods try to model the normal behavior of the network and find the deviation from this normal behavior. The existing methods mainly focus on Statistical approaches [4], [5] and Machine Learning techniques [6], [7].

In [6] a framework is introduced to detect the anomalous events caused by worm attacks and network blackout. They use three separate Support Vector Machine (SVM) models to classify each anomaly event into normal and abnormal. In [8], the authors introduce a framework which uses a single decision tree for classification of all types of anomalies. The existing work on both categories, i.e. Statistical and Machine Learning, suffer from either relatively low accuracy or inability to detect multiple types of anomalies.

In this work, we present an anomaly detection framework for BGP protocol using Neural Network to address the aforementioned shortcomings. The proposed framework includes three main steps; first, we preprocess the raw data to remove unnecessary and redundant incidents, then we extract features, and finally we train a neural network and cross validate it to evaluate the performance. The results show that our method is more general and accurate than the similar existing ones.

II. Proposed Methodology

In this work, we analyze abnormal BGP behaviors caused by worm attacks using Neural Networks. The proposed method consists of three main processes: data preparation, feature extraction, and classification. First, we parse raw data sets gathered from two databases using *libbgpdump* tool and convert them to regular format to be processed in the next step. Then, the features are extracted from data sets using methods presented on [9] for every time slot, which is set to 60 seconds in this work. Next, we train a Multi-Layer Perception (MLP) Neural Network using a certain portion of data set to build the classifier. Finally, we evaluate the classifier's performance with the remaining data.

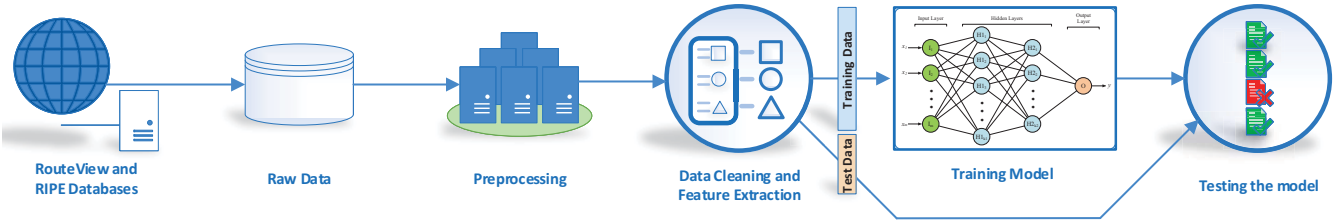

Fig. 1. The structure of the proposed framework

A. Data Structure

In this section the origin of the data sets, structure of data, and features are described. The routing data is selected from two different classes of normal and abnormal network. Since the main focus of this research is the effect of worm attacks on router update messages behavior, normal and abnormal classes of data are selected from the router's updates in presence and in absence of worm respectively. The data are gathered from [10] and [11]. Table I shows the virus name, source of data, date, and duration of the virus presence in the network. The virus presence duration is based on the information reported in [12].

TABLE I
WORM SPECIFICATION

Worm Name	Source of Data	Date	Duration
Nimda	rrc04, Geneva	Sept 18, 2001	24 hours
Slammer	Routeviews, Oregon	Jan 25, 2003	24 hours
Code-Red	rrc02, Paris	Jul 19, 2001	24 hours

To extract features, a Python script is written to parse the files that were generated using *libbgpdump*. The script calculates all the the parameters for a given time slot. Extracted features and a brief description of each feature have been shown in Table II.

TABLE II
LIST OF EXTRACTED FEATURES

BGP Features		
Feature		**Explanation**
1	**Updates**	# of update messages.
2	**A-Updates**	# of updates which only announce prefixes.
3	**W-Updates**	# of updates which withdraw at least one prefix.
4	**A-Prefix**	# of announced prefixes
5	**W-Prefix**	# of withdrawal prefixes
6	**A-Dup**	# of duplicated announced prefixes
7	**W-Dup**	# of duplicated withdrawal prefixes
8	**AW-Mix**	# of withdrawal after announced or vice versa.

B. The Classifier Framework

This section describes the architecture of the proposed classifier framework, its components, and their operations. As mentioned before, *libbgpdump* generated raw data is divided into 60 seconds slots/intervals, and the features introduced in Table II have been extracted for each of them.

The features and their corresponding classes, i.e. normal or abnormal, are divided into two portions: 90% of them are used for training phase and 10% are used for testing phase to evaluate the overall performance of the classifier. It is worth mentioning that for each portion, the data was selected randomly.

The classifier utilized in this research is based on MLP Neural Network which uses Back-Propagation technique for training. The structure of the whole framework is shown in Figure 1. For every set of extracted features, one output is considered to be labeled as normal or abnormal. The corresponding outputs for normal and abnormal feature sets are assigned -1 and +1, respectively.

In the training phase, the 90 percent of extracted feature vectors along with their corresponding outputs are fed to the network. The input layer of the network consists of input nodes of feature vector. Therefore, the number of neurons in the first layer is the number of features which here is 8. Since there are two classes in this work, i.e. normal and abnormal, the number of neurons in the last layer is 1 and a threshold unit is added afterward to classify between two predefined classes. Number of hidden layers are chosen 2 and the optimum number of nodes in each hidden layer is chosen based on (1) and (2) [13].

$$k1 = \sqrt{N * (p+2)} + 2\sqrt{\frac{N}{p+2}} \tag{1}$$

$$k2 = p\sqrt{\frac{N}{p+2}} \tag{2}$$

Where p, k1, and k2 are number of outputs, number of neurons in first hidden layer, and number of neurons in second hidden layer respectively.

The activation function used to calculate each layer's output from the summation of weighted inputs is based on *tansig* function which is the Hyperbolic Tangent Sigmoid. This activation function has output range from -1 to +1 that is very suitable in cases where there are only two classes.

TABLE III
OVERVIEW OF NEURAL NETWORK SPECIFICATION

Neural Network Specification	
Number of Samples (N)	5535 Samples
Input Layer	8 Neuron
Output Layer (p)	1 Neuron
First Hidden Layer (k1)	129 Neurons
Second Hidden Layer (k2)	43 Neurons
Activation Function	Sigmoid
Maximum Number of Epochs	10000
Simulation Framework	MATLAB

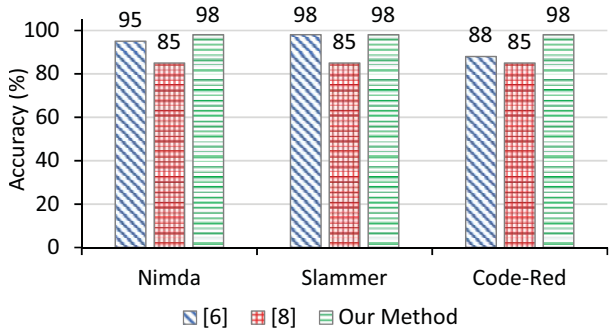

Fig. 2. Anomaly detection accuracy comparison

Other settings of the network are set using trial and error to achieve the best performance. Specifications of the Neural Network used in this project are summarized in Table III.

III. Experimental Results

In the first step of experiments, a Neural Network model is built for 90% of combined data set of three BGP events. For every abnormal day, one day is also used as normal day to build the whole data set. The normal day for each data set is considered as ten days before each anomaly happened. For example, Sept 8, 2001 is used as a normal day for Nimda event. After training the network, the test set, containing 10% of unseen normal data and 10% of unseen abnormal data are passed to the model to verify its ability to classify the behaviour of the network to normal or abnormal.

Fig. 2 shows the comparison of anomaly detection accuracy for our approach versus [6] and [8]. We used data sets from Slammer, Nimda, and Code-Red worms that affected performance of the global Internet BGP for training and testing the proposed classifier, which is the same data set used by [6] and [8].

According to the results, our classifier outperforms [8] in detecting all types of anomalies. Although our classifier accuracy is slightly better or equal to that of [6], we use a single classifier to detect all the anomalies which adds more generality to our classifier.

IV. Conclusion

In this project we proposed a method to classify abnormal behaviour of network by designing a Neural network classifier. To classify data we used 8 features based on BGP update messages. The data of three different data sets from three different types of worms, each of which were presented in different dates, were used in this work to extract features. The features were fed to an MLP Neural Network to build the classifier. Finally, a set of test data were used to evaluate the performance of the network. We could achieve 98% accuracy by using only 8 features and training a single classifier for all three types of data sets which is the best performance in this area to the best of our knowledge.

One of the possible future works could be adding more features to the feature vector that possibly make the trained network more effective and therefore would lead to better performance of the trained classifier. A more precise data set that separates the data more accurately would also help to improve the performance of the classifier.

References

[1] B. Al-Musawi, P. Branch, and G. Armitage, "Detecting bgp instability using recurrence quantification analysis (rqa)," in *2015 IPCCC*, Dec 2015.

[2] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang, *Analysis of BGP Update Surge during Slammer Worm Attack*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 66–79.

[3] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "Bgp hijacking classification," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, June 2019, pp. 25–32.

[4] M. C. Ganiz, S. Kanitkar, M. C. Chuah, and W. M. Pottenger, "Detection of interdomain routing anomalies based on higher-order path analysis," in *Sixth International Conference on Data Mining (ICDM'06)*, Dec 2006.

[5] S. Deshpande, M. Thottan, T. K. Ho, and B. Sikdar, "An online mechanism for bgp instability detection and analysis," *IEEE Transactions on Computers*, vol. 58, no. 11, pp. 1470–1484, Nov 2009.

[6] A. Allahdadi, R. Morla, and R. Prior, "A framework for BGP abnormal events detection," *CoRR*, vol. abs/1708.03453, 2017. [Online]. Available: http://arxiv.org/abs/1708.03453

[7] N. M. Al-Rousan and L. Trajković, "Machine learning models for classification of bgp anomalies," in *2012 IEEE 13th International Conference on High Performance Switching and Routing*, June 2012.

[8] Q. Wu and M. Wang, "Abnormal bgp routing dynamics detection by sampling approach in decision tree," in *2009 First International Workshop on Database Technology and Applications*, April 2009, pp. 170–173.

[9] A. Moreira, "Anomaly detection in enterprise networks," Master's thesis, Faculty of Engineering, University of Porto, 2011.

[10] U. of Oregon. (2015, jul) University of oregon route views project. [Online]. Available: Available:http://www.routeviews.org/

[11] R. I. E. N. C. Center. (2015, june) University of oregon route views project. [Online]. Available: http://www.ripe.net/

[12] N. y. Liang, G. b. Huang, P. Saratchandran, and N. Sundararajan, "A fast and accurate online sequential learning algorithm for feedforward networks," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1411–1423, Nov 2006.

[13] D. Stathakis, "How many hidden layers and nodes?" *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 2133–2147, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1080/01431160802549278