SECURING BGP BY HANDLING DYNAMIC NETWORK BEHAVIOR AND UNBALANCED DATASETS

Rahul Deo Verma[1], Shefalika Ghosh Samaddar[2] and A. B. Samaddar[2]

[1]Ph.D. Scholar, Department of Computer Science and Engineering,
National Institute of Technology Sikkim, India
[2]Department of Computer Science and Engineering,
National Institute of Technology Sikkim, India

ABSTRACT

The Border Gateway Protocol (BGP) provides crucial routing information for the Internet infrastructure. A problem with abnormal routing behavior affects the stability and connectivity of the global Internet. The biggest hurdles in detecting BGP attacks are extremely unbalanced data set category distribution and the dynamic nature of the network. This unbalanced class distribution and dynamic nature of the network results in the classifier's inferior performance. In this paper we proposed an efficient approach to properly managing these problems, the proposed approach tackles the unbalanced classification of datasets by turning the problem of binary classification into a problem of multiclass classification. This is achieved by splitting the majority-class samples evenly into multiple segments using Affinity Propagation, where the number of segments is chosen so that the number of samples in any segment closely matches the minority-class samples. Such sections of the dataset together with the minor class are then viewed as different classes and used to train the Extreme Learning Machine (ELM). The RIPE and BCNET datasets are used to evaluate the performance of the proposed technique. When no feature selection is used, the proposed technique improves the F1 score by 1.9% compared to state-of-the-art techniques. With the Fischer feature selection algorithm, the proposed algorithm achieved the highest F1 score of 76.3%, which was a 1.7% improvement over the compared ones. Additionally, the MIQ feature selection technique improves the accuracy by 3.5%. For the BCNET dataset, the proposed technique improves the F1 score by 1.8% for the Fisher feature selection technique. The experimental findings support the substantial improvement in performance from previous approaches by the new technique.

KEYWORDS

Border Gateway Protocol (BGP), Extreme Learning Machine (ELM), Anomaly Detection.

1. INTRODUCTION

The Internet is a network without a center that is interconnected. It consists of thousands of autonomous (AS) systems. Border Gateway Protocol (BGP) is designed and implemented for the transmission of packets across the ASes. In BGP, the prefixes owned by each autonomous system will be announced and routing information learned from its neighbors will be propagated according to policy. The ASes must follow a path to the source of the prefix while propagating the prefix and will be able to choose between different paths. However, even being a core

component of the Internet infrastructure, BGP consists of several serious security vulnerabilities [1].

There are four types of messages sent over BGP: open, update, keep alive, and notification, which is defined by metrics such as the shortest path to the nearest next-hop router, and routing policies.While peer-to-peer messages are exchanged, a variety of events such as router misconfigurations, session resets, and link failures can trigger BGP anomalies. Any upgrade which does not represent a shift in the underlying BGP network or routing policy is the simplest concept of BGP anomalies. Such irregularities undermine the efficiency and performance of the network.

There have been numerous techniques used to detect BGP anomalies [2]. Unfortunately, these existing anomaly detection approaches perform poorly with highly unbalanced traffic characteristics (as malicious traffic amounts are very small compared to normal traffic), also, these approaches do not account for the network's dynamic nature. The unbalanced distribution of class and the dynamic nature of the network contribute to the classifier's inferior efficiency. We, therefore, need a technique to deal with the above-mentioned issue that not only learns from such unbalanced datasets but also preserves a margin between training samples and classifier boundaries so that we can deal with the network's dynamic behavior. To achieve this, a classifier based on Affinity Propagation and ELM is proposed in this work.

The proposed approach addresses the unbalanced classification of datasets by converting the issue of binary classification into a problem of multi-class classification. Using Affinity Propagation, the majority of class samples are clustered into multiple groups. With this, we divide the data samples of the majority of class into multiple classes each of which contains samples approximately equal to minority classes. Together with the minor class, these dataset clusters are used to train the classifier Extreme Learning Machine (ELM) to handle the problem of multiclass classification.

2. LITERATURE REVIEW

Several methods have been suggested by analyzing traffic patterns to detect anomalies. One of the early and common methods is to develop traffic behavior models based on statistical techniques [3, 4], identifying the anomalies as correlated abrupt changes that occur in the underlying distribution. The downside, however, is that with all possible cases, it is difficult to estimate the dimensional distributions. Clustering techniques [5, 6, 7] have also been suggested to identify all regular traffic data points belonging to one cluster while anomalous data points that belong to multiple clusters. Clustering techniques have the main disadvantage of being optimized to detect regular traffic, which is not the goal of detection methods. An alternative widely used approach is the rule-based technique [8, 9, 10], which builds classifiers based on specific rules. The downside is that a priori knowledge and a high degree of computations are required. Several machine learning techniques have been used to create traffic classification models [11, 12, 13, 14, 15, 16] to identify anomalies for both unsupervised and supervised machine learning models. Despite the ability of neural networks to detect the complex relationship between features, they have many disadvantages such as high computational complexity and high probability of overfitting. Support vector machine (SVM) techniques use nonlinear classification functions to identify anomaly patterns in data and classify that data point based on the value obtained by the classifier function. SVMs build a classification model that maximizes the difference between each class's data points. Several variants of SVM detection techniques are introduced and evaluated [17], but due to the quadratic optimization problem that needs to be solved, they have high computational complexity. Finally, due to their low time complexity, Bayesian networks (BN) techniques [18] are used in many real-time classification systems. BNs rely on two

hypotheses: features are conditionally independent given a target category and the resulting likelihood is the criterion for classification between any two data points. Variants of BNs have been introduced by several anomaly detection schemes [19, 20]. Most of the models described here are not intended for sequence classification and are not appropriate for time series anomaly detection, where only input instances are handled separately without taking into account the sequenced existence of traffic data. In fact, traffic data are multivariate time series and the patterns of the anomaly are gradually varying with time information. In [21], an ELM-based intrusion detection approach is presented that addresses the class imbalance problem. The results of the experiments show that the ELM outperforms the SVM. Knowing that, an ELM has the advantage of faster computation for both training and testing, the ELM emerges as a promising technique for classification problems. However, selecting the correct number of hidden nodes in an ELM is still a difficult task.

3. APPROACH

In the section, the details of Affinity Propagation and Extreme Learning Machine (ELM) are presented. Clustering algorithms such as affinity propagation are often used for unsupervised learning, while feedforward neural networks such as ELM use multiple layers of hidden nodes with no need to tune their parameters.

3.1. Affinity Propagation (AP)

The propagation of affinity (AP) has been suggested as a new and powerful exemplary learning technique. In short, the user must provide a complete matrix of similarities between the input data points as the initial input to the algorithm (for the selected metric(s)). First of all, all data points are seen as potential examples. As soon as information messages (i.e. responsibility and availability) are transmitted along the network edge (each data point acts as a node), identifying possible examples and clusters [23].

In the following sections, we explain the AP mathematical model in brief. AP starts with a set of real-valued similarities between data points as input. Given a N data point's dataset, x_i and x_j are two objects in it. The similarity $s(i,j)$ indicates how well x_j is suited to be the exemplar forx_i. For instance, it can be initialized to $s(i,j) = -\left\|x_i - x_j\right\|^2, i \neq j$. In [22], if no heuristic knowledge is present, self-similarities are referred to as preferences and are often inferred as constants. For instance, they could be set as:

$$s(l,l) = \frac{\sum_{i,j=1;i*j}^{N} s(i,j)}{N \times (N-1)}, 1 \leq l \leq N \tag{1}$$

Then, the AP method computes two types of messages which are then exchanged between data points. The first one identifies a "responsibility" $r(i,j)$ that is sent from the i to the j, a good indication of whether point j serves as a suitable exemplar for i. In the second message, called "availability" $a(i,j)$, candidate exemplar point j transmits his or her availability to point i and tells point i the accumulated evidence to determine whether or not point i should accept point j as its exemplar. The availabilities are initially set to zero, $a(i,j) = 0$. The update equations for $r(i,j)$ and $a(i,j)$ are written as:

$$sr(i,j) = s(i,j) - \max_{j \neq j} \{a(i,j') + s(i,j')\} \tag{2}$$

$$\begin{cases} min\left\{0, r(j,j) + \sum_{i \neq i,j} \max\{0, r(i,j)\}\right\} & i \neq j \\ \sum_{i \neq j} \max\{0, r(i,j)\} & i = j \end{cases} \quad (3)$$

To prevent numerical oscillations due to the exchange of messages between data points, a damping factor of $\lambda = [0,1\ ]$ is also applied:

$$\begin{aligned} R_{t+1} &= (1-\lambda)R_t + \lambda R_{t-1} \\ A_{t+1} &= (1-\lambda)A_t + \lambda A_{t-1} \end{aligned} \quad (4)$$

Where, $R = r(i,j)$ and $A = (i,j)$ represent the matrix of responsibility and the matrix of availability respectively; t indicates the time of iteration. For several iterations, the above two messages will be modified iteratively until they exceed certain specified values or the local decisions remain constant. At this point, it is then possible to combine availabilities and responsibilities to define exemplars:

$$c_i \leftarrow \arg\max_{1 \leq j \leq N} [r(i,j) + a(i,j)] \quad (5)$$

Affinity Propagation Algorithm

Input: a set of pairwise similarities, $\{s(i,k)\}_{(i,k)\in\{1,2,\dots,N\}^2, i\neq k}$where $s(i,k) \in \mathbb{R}$ indicates the suitability of data point k as an exemplar for data point i, and is calculated as:

$$s(i,k) = -\|x_i - x_k\|^2, i \neq k,$$

There is a real number s (k, k) for each point; this real number indicates that this point is preferred a priori (a cluster is a small cost to add).

$$s(k,k) = p\ \forall k \in \{1, \dots, N\}$$

Initialization: set availabilities to zero $\forall i, k: a(i,k) = 0$.

Repeat: update$s(i,k)$, and $r(i,k)$ until convergence achieved

$$\forall i, k: r(i,k) = s(i,k) - \max_{k', k' \neq k} [s(i,k') + a(i,k')]$$

$$\forall i, k: a(i,k) = \begin{cases} \sum_{i', i' \neq i} \max[0, r(i',k)], for\ k = i \\ \min\left[0, r(k,k) + \sum_{i', i' \neq i} max[0, r(i',k)]\right], for\ k \neq i \end{cases}$$

Output: assignments$\hat{c} = (\hat{c}_1, \hat{c}_2, \dots, \hat{c}_N)$, where $\hat{c}_i = argmax_k\ [a(i,k) + r(i,k)]$ and $\hat{c}_i$ indexes the cluster's exemplar to which point i is assigned.

Several comprehensive analyzes of the AP method are performed ([23], [24]) for different scale datasets. A comparison of Affinity Propagation clustering with standard approaches (like p-median analysis and vertex heuristic substitution) shows that there are only minor differences for both precision and speed on small datasets. For large datasets, however, AP offers notable benefits over existing methods [22, 24].

3.2. Extreme Learning Machine

An emerging algorithm in machine learning is called the Extreme Learning Machine (ELM) [25]. It is based on a single hidden layer feedforward neural network (SLFN), which trains quickly and provides performance similar to support vector machines (SVMs) [26]. ELM is a standard three-

layer feedforward design that was introduced in 2006 [27]. The model contains two layers: an input layer and a hidden layer (the sigmoid nonlinear neurons) projecting the input layer onto higher dimensions. The final layer serves as the output and is made up of linear input-output neurons. Fig. 1 shows the ELM structure.

$$y(p) = \sum_{j=1}^{m} \beta_j g\left(\sum_{i=1}^{n} w_{i,j} x_i + b_j\right) \tag{6}$$

Here, β_i and β_j represents the weights between the hidden layer and the input layer, and the output layer and the hidden layer respectively. The hidden layer neuron threshold value b_j and its activation function $g(.)$. Weights of the same input layer $(w_{i,j})$ and bias (b_j) are assigned randomly. In the beginning, the network is initialized by allocating the input layer neuron number (n) and hidden layer neuron number (m), and the activation function $(g(.))$. Now, based on this information, by combining and rearranging the parameters known in equilibrium, the output layer becomes as in equation (8) [28].

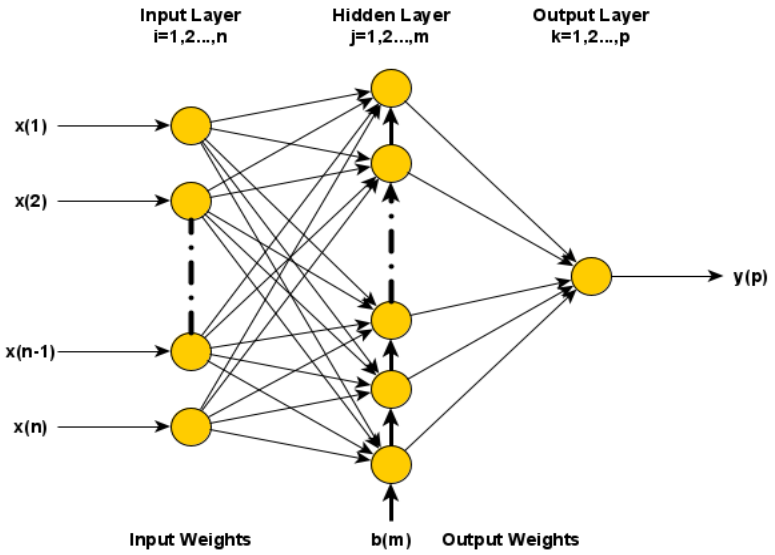


Figure 1. Feed-forward neural network with a single hidden layer in an ELM structure.

$$H(w_{i,j}, b_j, x_i) = \begin{bmatrix} g(w_{1,1}x_1 + b_1) & \cdots & g(w_{1,m}x_m + b_m) \\ \vdots & \ddots & \vdots \\ g(w_{n,1}x_n + b_1) & \cdots & g(w_{n,m}x_m + b_m) \end{bmatrix} \tag{7}$$

$$y = H\beta \tag{8}$$

Like all training algorithm models, the objective should be to minimize errors as much as possible. The output y_p error function obtained from the real output value y_o in ELM is $\sum_k^s (y_o - y_p)$(with "s": training data number) and $\left\| \sum_k^s (y_o - y_p)^2 \right\|$ can be minimized. The output y_p obtained from the actual output value y_o must be equal to y_p for both of these functions. If this occurs, the unknown parameter in equation (8) will be a very low probability matrix (H). It

means that there will never be the same number of samples in the training set as there are features in each sample.Therefore, it will be a challenge to take the inverse of H and to find weights (b). To overcome this situation, the pseudo-inverse of the matrix H can be taken by using Moore-Penrose Inverse. The output weights can therefore be found through $\beta = y \times MPI(H)$.

3.3. Experimental Datasets

BGP raw data are collected from AS513 (RIPE RIS, rcc04, CIXP, Geneva) during worm attacks like Slammer [29], Nimda [30], and Code Red I [31]. for the same duration, we downloaded the standard BGP datasets from RIPE NCC [32] and the BCNET Network Operations Center [33] from Vancouver, Canada. To convert MRT [34] to ASCII format, the libBGPdump tool [35] is used. Based on the tools written in C #, we parse the ASCII file and extract 37 features sampled in five days each minute, generating 7,200 samples for each anomaly case.

Filtered collected traffic data for BGP update messages during the intervals when the internet experienced BGP anomalies is provided in [36]. Three anomalous traffic events and two regular traffic events, described in this paper, are listed in Table 1.

Table 1. Details of BGP datasets [36]

Dataset	Class	Date	Duration (h)	Training set data points	Testing set data points
Slammer	Anomaly	January 25, 2003	16	3212:4080	1:3211, 4081:7200
Nimda	Anomaly	September 18, 2001	59	3680:7200	1:3679
Code Red I	Anomaly	July 19, 2001	10	3681:4280	1:3680, 4281:7200
RIPE	Regular	July 14, 2001	24	None	1:1440
BCNET	Regular	December 20, 2011	24	None	1:1440

3.4. Features Analysis

Information and characteristics influence the model's classification output and determine the machine learning upper limit. The BGP raw data is used to generate 37 features set with obvious physical or statistical meaning. The feature set is obtained through the extraction process described in [37], and their values are calculated in one-minute intervals, which produces 7200 samples for each anomaly condition. The details of these features are presented in [36], these features are divided into three types (Continuous, Categorical, and Binary), and grouped into two categories named volume (how many BGP announcements are made) and AS-path (the maximum edit distance between the two ASs). For the complete details of the features set please refer to [36], as in this work the similar features set properties are adopted.

3.5. Feature selection

The feature vector's high dimensionality due to non-informative features is considered unnecessary because it increases computational complexity and the use of memory [38]. It also leads to poor accuracy in classification. To decrease dimensionality, it is appropriate to choose the most relevant subset of the original set of features. A Fisher [39, 40] and a minimal Redundancy Maximum Relevance (mRMR) [41] algorithm were used to identify the most significant features.

Table 2. Top 10 selected features by different algorithms

Selected features highest score (top) to lowest score (bottom)	Fisher	MID	MIQ	MIBASE
	11	34	34	34
	6	32	2	36
	25	33	8	2
	9	2	24	8
	2	31	9	9
	36	24	14	3
	37	8	1	1
	24	14	36	6
	8	30	3	12
	14	22	25	11

To select the top ten features, we use three variants of the mRMR algorithm: Mutual Information Difference (MID), Mutual Information Quotient (MIQ), and Mutual Information Base (MIBASE). These selected features according to the feature selection algorithm are presented in Table 2.

3.6. Proposed Methodology

The proposed BGP anomaly detection model is shown in Fig. 2. Classification involves categorizing test labels into predefined categories. In this work, four different classes Slammer, Nimda, Code Red, and Regular are defined as presented in Table 1. The steps of the classification process for the proposed model are shown in Fig. 2. Initially, raw data is processed to extract features and labels. The datasets are sorted by affinity propagation clustering into multiple groups and divided into two sets of training data and testing data, as shown in Fig. 2. In the training stage, the training set which consists set of predefined features and respective labels is used to train the extreme learning machine (ELM). The trained ELM is used as an anomaly detector on the test dataset. In the testing stage, the testing dataset is applied to the trained ELM model to generate classification labels. At the end of the process, a set of labels generated from the testing step is compared with the original labels from the testing step to evaluate the performance of the proposed model.

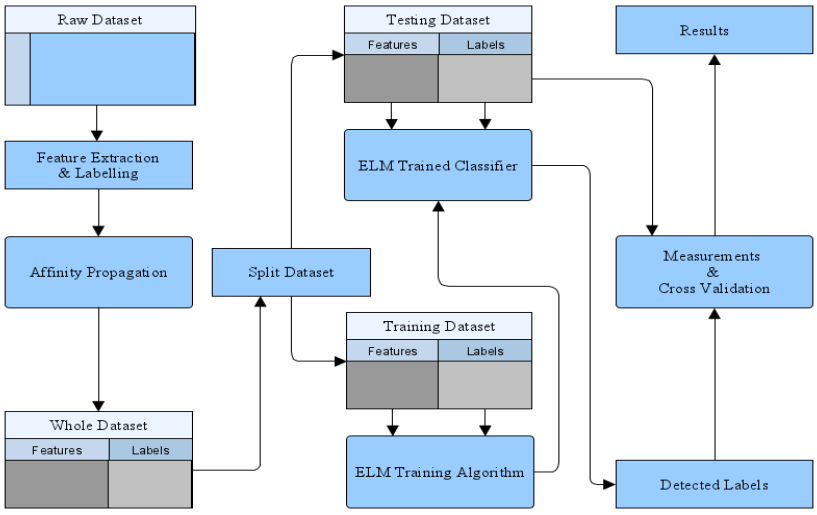


Figure 2. Proposed Classification Process

4. PERFORMANCE EVALUATION METRICS

The classifier is evaluated based on four common measures known as accuracy (Eq. 10), precision (Eq. 11), recall (Eq. 12), and F1 (Eq. 9) to estimate the efficiency of the methods. Accuracy determines the predictive ability of the classifier for normal and anomaly assessments. Accuracy defines the predicted accuracy of the label. Recall determines the completeness of the category. F1 is a dimensionless measure of precision and recall that can be used to balance accuracy and recall.

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

$$Accuracy = \frac{|TP + TN|}{|TP + TN + FP + FN\,|} \tag{10}$$

$$Precision = \frac{|TP|}{|TP + FP\,|} \tag{11}$$

$$Recall = \frac{|TP|}{|TP + FN|} \tag{12}$$

Where, $TP, TN,$ are denoting the true positive, true negative, while FP, FN are denoting the false positive and false negative. These terms can be defined as:

- TP: The number of anomalous instances classifier correctly identified as an anomaly.
- TN: The number of normal instances classifier correctly identified as normal.
- FP: The number of normal instances classifier incorrectly identified as an anomaly.
- FN: The number of anomalous instances classifier incorrectly identified as normal.

These can also be defined by the confusion matrix.

Table 3. Confusion Matrix

		Known Labels	
		True (Anomaly)	False (Normal)
Classifier's identification result	Positive (Anomaly)	TP	FP
	Negative (Normal)	FN	TN

5. RESULTS ANALYSIS

Table 4 shows the performance comparison of previous methods (SVM (Support Vector Machine), HMM (Hidden Markov Model) and, NB (Naïve Bayes)) with the proposed method. For the RIPE dataset when compared to previous methods, the proposed approach delivers better accuracy for MID and MIQ-based features. For MID-based features, it provides 75.3% accuracy which is 0.4% higher than the 74.9 the previous best provided by HMM. For MIQ-based features, we get 74.8% which is 3.5% higher than the previous best 71.3% provided by SVM. When comparing in terms of F1 score the proposed classifier works better for all features set except for the MIBASE. Since the F1 score presents the combined information of Precision and Recall the higher value of it even when the Accuracy measure is lesser shows the better classifier performance.

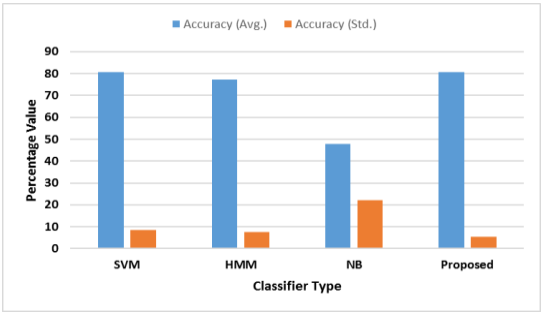
For the BCNET dataset when compared to previous methods, the proposed approach delivers better accuracy for MID-based features and gives 80.6% accuracy which is 1.7% higher than the 78.9 the previous best provided by HMM. When comparing in terms of F1 score the proposed classifier works better for the Fisher and MID feature set.

Table 4. Performance Comparison

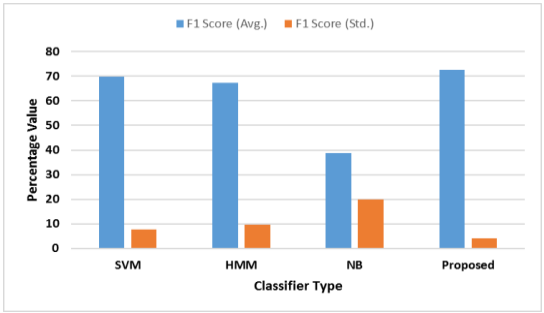
Dataset	Feature Set	Accuracy (%)				F1 (%)			
		SVM	HMM	NB	Proposed	SVM	HMM	NB	Proposed
RIPE	1-37	77.1	81.3	74.3	79.1	71.2	70.7	64.3	**73.1**
RIPE	Fisher	82.8	79.2	24.7	81.7	74.6	69.3	24.1	**76.3**
RIPE	MID	67.8	60.6	74.9	**75.3**	56.3	50.5	65.3	**70.6**
RIPE	MIQ	71.3	68.2	24.6	**74.8**	55.1	48.2	22.7	**69.5**
RIPE	MIBASE	72.8	74.8	75.4	71.2	68.9	67.7	60.5	63.4
BCNET	1-37	91.4	86.6	67.6	85.5	74.4	75.1	56.8	71.8
BCNET	Fisher	85.7	81.3	34.3	84.2	73.8	74.8	25.1	**76.6**
BCNET	MID	78.7	78.9	33.1	**80.6**	71.3	73.3	22.1	73.3
BCNET	MIQ	89.1	81.1	34.8	86.3	75.6	72.8	24.9	**76.7**
BCNET	MIBASE	90.2	81.4	33.1	87.8	75.4	71.5	21.8	75.1

The proposed algorithm gives an average 80.65% accuracy for all datasets with a standard deviation of 5.52%, whereas SVM provides 80.69% accuracy however with an 8.42% standard deviation. When analyzed for the F1 scores the proposed method gives a higher average F1 score of 72.64% with only 4.11% of standard deviation in comparison to the second-best SVM which provides 69.66% average accuracy with a 7.66% standard deviation. This validates that the proposed method gives much uniform performance for all the five features set in terms of accuracy and F1 score.

Looking at the performance of different feature selection algorithms Fisher provides better average accuracy of 77.95% with a 2.9% of standard deviation for the RIPE dataset however for the BCNET dataset MIBASE gives better accuracy of 73.12% with 26.94% of standard deviation for the BCNET dataset. For the F1 score, the Fisher feature selection algorithm gives better performance for both the datasets and achieves an average of 69.82% and 62.57% respectively.

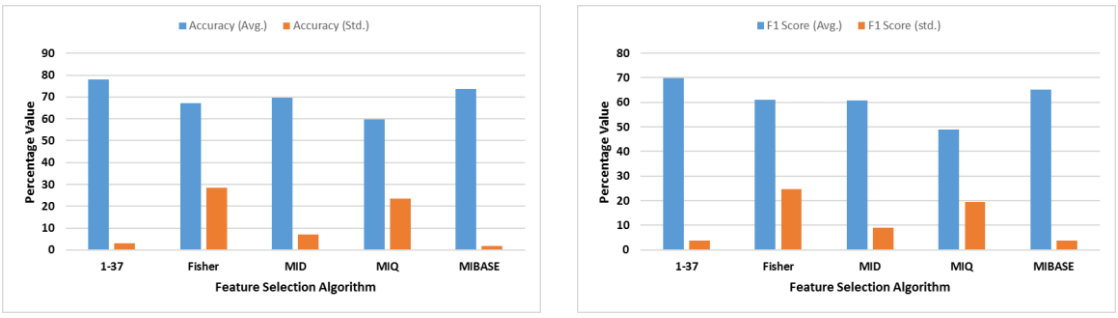


(a)



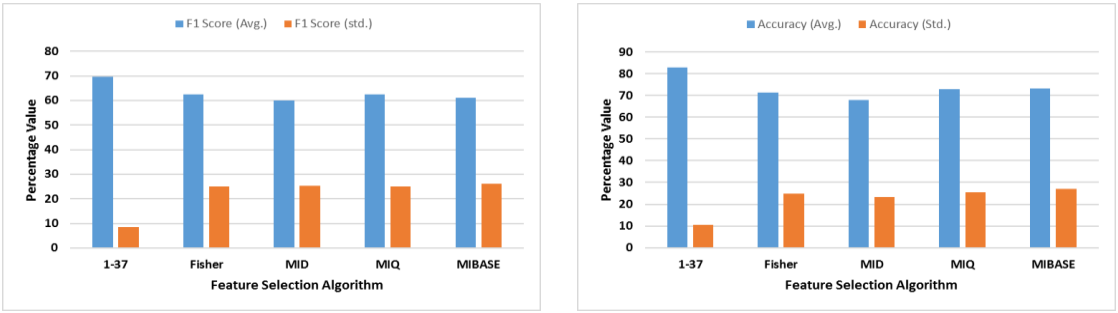
(b)

Figure 3. Mean and standard deviation of (a) Accuracy, and (b) F1 Score, of different classification techniques for the RIPE+BCNET dataset



(a) (b)

Figure 4. Mean and standard deviation in (a) Accuracy, and (b) F1 Score, of different feature selection techniques for only the RIPE dataset



(a) (b)

Figure 5. Mean and standard deviation in (a) Accuracy, and (b) F1 Score, of different feature selection techniques for only the BCNET dataset

6. CONCLUSION

In this paper, we presented an Affinity Propagation and Extreme Learning Machine (ELM) based approach for anomaly detection in the BGP network. Affinity Propagation-based clustering used the datasets processing phase, while ELM during the classification phase. Finally, the performance of the proposed algorithm is evaluated with two different datasets named RIPE and BCNET with four different feature selection algorithms. The experimental results reveal that the proposed algorithm performs better than SVM, HMM, and NB algorithms and provides much stable performance throughout the datasets and feature selection algorithms. The experimental results show that for both datasets and balanced and non-balanced class distributions, the proposed algorithm provides significantly improved performance over previous algorithms. Although the proposed algorithm performs better than compared algorithms, the algorithm may require several repetitions to get the best solution. This happens because with AP it is difficult to get the optimal parameter values, also AP may involve oscillations. In the future, these limitations may be addressed using enhanced versions of the AP algorithm.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] S. Murphy, "BGP security vulnerabilities analysis," Tech. Rep., 2005.

[2] Al-Musawi B, Branch P, Armitage G. BGP anomaly detection techniques: a survey. IEEE CommunSurv Tut 2017; 19: 377-396.

[3] Chandola V, Banerjee A, Kumar V. Anomaly detection: a survey. ACM ComputSurv 2009; 41: 1-58.

[4] Hajji H. Statistical analysis of network traffic for adaptive faults detection. IEEE T Neural Networ 2005; 16: 1053-1063.

[5] Thottan M, Liu G. Anomaly detection approaches for communication networks. In: Cormode G, Thottan M, editors. Algorithms for Next Generation Networks. London, UK: Springer, 2010. pp. 239-261.

[6] Wang, Bingming, Shi Ying, Guoli Cheng, Rui Wang, Zhe Yang, and Bo Dong. "Log-based anomaly detection with the improved K-nearest neighbor." International Journal of Software Engineering and Knowledge Engineering 30, no. 02 (2020): 239-262.

[7] Putina, Andrian, and Dario Rossi. "Online anomaly detection leveraging stream-based clustering and real-time telemetry." IEEE Transactions on Network and Service Management (2020).

[8] Tan P, Steinbach M, Kumar V. Introduction to Data Mining. 1st ed. Boston, MA, USA: Addison-Wesley, 2005.

[9] Protogerou, Aikaterini, Stavros Papadopoulos, Anastasios Drosou, Dimitrios Tzovaras, and Ioannis Refanidis. "A graph neural network method for distributed anomaly detection in IoT." Evolving Systems (2020): 1-18.

[10] Li, Huichun, Chengli Zhao, Yangyang Liu, and Xue Zhang. "Anomaly detection by discovering bipartite structure on complex networks." Computer Networks (2021): 107899.

[11] Augusteijn M, Folkert B. Neural network classification and novelty detection. Int J Remote Sens 2002; 23: 2891-2902.

[12] Diaz I, Hollmen J. Residual generation and visualization for understanding novel process conditions. In: IEEE IJCNN'02 Neural Networks Conference; 12–17 May 2002; Honolulu, HI, USA. New York, NY, USA: IEEE. pp. 2070-2075.

[13] Xu, Mengying, and Xing Li. "BGP Anomaly Detection Based on Automatic Feature Extraction by Neural Network." In 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), pp. 46-50. IEEE, 2020.

[14] Takhar, Hardeep Kaur. "Machine Learning Techniques for Detecting BGP Anomalies." PhD diss., SIMON FRASER UNIVERSITY, 2020.

[15] Al-Akhras, Mousa, Mohammed Alawairdhi, Ali Alkoudari, and Samer Atawneh. "Using machine learning to build a classification model for iot networks to detect attack signatures." (2020).*International journal of Computer Networks & Communications. 12. 99-116. 10.5121*.

[16] Hai, Tran Hoang, and Eui nam Huh. "Network Anomaly Detection Based On Late Fusion Of Several Machine Learning Algorithms." *International Journal of Computer Networks and Communications* 12, no. 6 (2020): 117-131.

[17] Sharma O, Girolami M, Sventek J. Detecting worm variants using machine learning. In: Proceedings of CoNEXT Conference; 10–13 December 2007; New York, NY, USA. New York, NY, USA: ACM. pp. 1-12.

[18] Gray, Caitlin, Clemens Mosig, Randy Bush, Cristel Pelsser, Matthew Roughan, Thomas C. Schmidt, and Matthias Wahlisch. "BGP Beacons, Network Tomography, and Bayesian Computation to Locate Route Flap Damping." In Proceedings of the ACM Internet Measurement Conference, pp. 492-505. 2020.

[19] Moore A, Zuev D. Internet traffic classification using Bayesian analysis techniques. In: Proceedings Conference on Measurement and Modeling of Computer Systems; 6–10 June 2005; Alberta, Canada. New York, NY, USA: ACM. pp. 50-60.

[20] El-Arini K, Killourhy K. Bayesian detection of router configuration anomalies. In: Proceedings of Workshop on Mining Network Data; 26 August 2005; Philadelphia, PA, USA. New York, NY, USA: ACM. pp.221-222.

[21] Awad, Mohammed, and AlaeddinAlabdallah. "Addressing Imbalanced Classes Problem of Intrusion Detection System Using Weighted Extreme Learning Machine." *International Journal of Computer Networks & Communications (IJCNC) Vol* 11 (2019).

[22] Dueck, Delbert. Affinity propagation: clustering data by passing messages. Toronto: University of Toronto, 2009.

[23] Taneja, Shweta, Bhawna Suri, Himanshu Narwal, Anchit Jain, Akshay Kathuria, and Sachin Gupta. "A new approach for data classification using Fuzzy logic." In Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference, pp. 22-27. IEEE, 2016.
[24] Song, Qinbao, Jingjie Ni, and Guangtao Wang. "A fast clustering-based feature subset selection algorithm for high-dimensional data." IEEE transactions on knowledge and data engineering 25, no. 1 (2013): 1-14.
[25] Huang, G.-B., Wang, D.H., Lan, Y.: Extreme Learning Machines: A Survey. International Journal of Machine Learning and Cybernetics 2, 107–122 (2011).
[26] Tapson, J., van Schaik, A.: Learning the pseudoinverse solution to network weights. Neural Networks 45, 94–100 (2013).
[27] Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme Learning Machine: Theory and applications. Neurocomputing 70, 489–501 (2006).
[28] Toprak, Abdullah. "Extreme Learning Machine (ELM)-Based Classification of Benign and Malignant Cells in Breast Cancer." Medical science monitor: international medical journal of experimental and clinical research 24 (2018): 6537.
[29] D. Moore, V. Paxson, and S. Savage. Inside the slammer worm [J]. IEEE Security & Privacy, 2003, 99(4): 33-39.
[30] A. Machie, J. Roculan, and R. Russellu. Nimda worm analysis[R]. Tech. Rep., Incident Analysis, Security Focus, 2001.
[31] D. Moore, and C. Shannon. Code-Red: a case study on the spread and victims of an Internet worm[C]//Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurement. ACM, 2002: 273-284.
[32] RIPE NCC [Online]. Available: http://www.routeviews.org/. Accessed: Feb. 28, 2018
[33] S. Lally, T. Farah, and R. Gill. Collection and characterization of BCNET BGP traffic. Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on. IEEE, 2011: 830-835.
[34] libBGPdump.[Online]. Available: http://bitbucket.org/ripencc wiki/Home. Accessed: Sep. 17, 2018
[35] T. Manderson. Multi-threaded routing toolkit (MRT) border gateway protocol (BGP) routing information export format with geo-location extensions. RFC 6397, IETF, Oct. 2011 [Online]. Available: https://tools.ietf.org/html/rfc6397.txt
[36] Al-Rousan, Nabil M., and Ljiljana Trajković. "Machine learning models for classification of BGP anomalies." In 2012 IEEE 13th International Conference on High Performance Switching and Routing, pp. 103-108. IEEE, 2012.
[37] N. Al-Rousan, and Lj. Trajkovic. Machine learning models for classification of BGP anomalies. IEEE, International Conference on High Performance Switching and Routing. IEEE, 2013:103-108.
[38] P. Winter, E. Hermann, and M. Zeilinger, \Inductive intrusion detection in flow-based network data using one-class support vector machines," in New Technologies, Mobility and Security (NTMS), 2011 4th IFIP Int. Conf., Paris, France, Feb. 2011, pp. 1-5.
[39] Y.-W. Chen and C.-J. Lin, \Combining SVMs with various feature selection strategies," in Feature Extraction: Foundations and Applications, London, Springer, June 2006, pp. 317-328.
[40] Q. Gu, Z. Li, and J. Han, \Generalized fisher score for feature selection," in Proc. Conf. on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 2011, pp. 266-273.
[41] H. Peng, F. Long, and C. Ding, \Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1226{1238, Aug. 2005.