Detecting BGP Anomalies Using Machine Learning Techniques

Qingye Ding, Zhida Li, Prerna Batta, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
Email: {qingyed, zhidal, pbatta, ljilja}@sfu.ca

***Abstract*—Border Gateway Protocol (BGP) anomalies affect network operations and, hence, their detection is of interest to researchers and practitioners. Various machine learning techniques have been applied for detection of such anomalies. In this paper, we first employ the minimum Redundancy Maximum Relevance (mRMR) feature selection algorithms to extract the most relevant features used for classifying BGP anomalies and then apply the Support Vector Machine (SVM) and Long Short-Term Memory (LSTM) algorithms for data classification. The SVM and LSTM algorithms are compared based on accuracy and F-score. Their performance was improved by choosing balanced data for model training.**

***Keywords*—Border gateway protocol; routing anomalies; machine learning; feature selection; support vector machine; long short-term memory.**

I. INTRODUCTION

Border Gateway Protocol (BGP) plays an essential role in routing data between Autonomous Systems (ASes) where an AS is a collection of BGP peers administrated by a single administrative domain [1]. The main function of BGP is to select the best routes between ASes based on routing algorithms and network policies enforced by network administrators. BGP anomalies may be caused by changes in network topologies, updated AS policies, or router misconfigurations. BGP anomalies affect Internet servers and hosts and are manifested by anomalous traffic behavior. Hence, they may be detected by analyzing collected traffic data and generating various classification models. A variety of techniques have been proposed [2]–[4] to detect BGP anomalies.

Machine learning techniques are the most common approaches for classifying BGP anomalies. While unsupervised learning techniques are often used for clustering, supervised learning is employed for anomaly classification when the input data are labeled based on various categories. Well-known supervised learning algorithms include Support Vector Machine (SVM) [5] and neural networks such as Hidden Markov Models (HMMs), Naive Bayes (NB), and Long Short-Term Memory (LSTM) [6]. SVM usually achieves the best accuracy and F-score compared to other machine learning algorithms. However, the SVM models have high computational complexity. LSTM is a recurrent neural network architecture that implements gradient-based deep learning algorithm. It outperforms other time-sequence learning algorithms because of its ability to learn from past experiences especially when long time intervals occur between events.

In this paper, we create the SVM and LSTM models to detect BGP anomalies. Since BGP events are sequential data streams, LSTM is a feasible classifier to identify BGP anomalies. We only consider BGP update messages because they contain the information about the BGP status and configuration that is sufficient for feature extraction. We extract BGP update messages from the collected data during the time periods when the Internet experienced known BGP anomalies. We select 10 features from the BGP datasets [7] using feature selection algorithms and then compare classification results generated by the SVM and LSTM algorithms. The minimum Redundancy Maximum Relevance (mRMR) [8] feature selection algorithms were employed to reduce the dimensionality of the dataset matrix. The SVM and LSTM classifiers are then used to detect anomalies.

This paper is organized as follows. In Section II, we describe the feature selection as well as the SVM and LSTM models used for anomaly detection. In Section III, we present the experimental procedure. Libraries and parameters used for feature selection and classification models are given in Section IV. Performance of the SVM and LSTM algorithms and their comparison are presented in Section V. We conclude with Section VI.

II. FEATURE SELECTION AND CLASSIFICATION ALGORITHMS

A. Feature Selection

Datasets containing BGP anomalies are collected from the Route Views project [9] while regular data are collected from the Réseaux IP Européens (RIPE) Network Coordination Centre (NCC) [7] and from BCNET [10]. We use 37 features extracted from BGP update messages that originated from AS 513. These features are collected per minute over five days: the day when the anomalies occurred, two days before, and two days after the anomalies, resulting in 7,200 data points. Three cases of well-known anomalies are considered: Slammer, Nimda, and Code Red I, as shown in Table I. For example, Slammer event occurred on January 25, 2003 and lasted 16 hours. Hence, BGP update messages collected between January 23, 2003 and January 27, 2003 are selected as samples for feature extraction.

TABLE I. BGP INTERNET ANOMALIES

Anomalies	Class	Date	Duration (h)
Slammer	Anomaly	January 25, 2003	16
Nimda	Anomaly	September 18, 2001	59
Code Red I	Anomaly	July 19, 2001	10

The high dimension of the dataset matrix increases the computational complexity and may lead to undesirable classification results. Hence, a subset of the original set of features is selected to create a new matrix. When training the SVM models, we employ minimum Redundancy Maximum Relevance (mRMR) [8] algorithms, which include Mutual Information Deference (MID), Mutual Information Quotient (MIQ), and Mutual Information Base (MIBASE). We select 10 features with the highest scores among the 37 features shown in Table II.

TABLE II. 37 EXTRACTED BGP FEATURES

Feature	Definition	Category
1	Number of announcements	*volume*
2	Number of withdrawals	*volume*
3	Number of announced NLRI prefixes	*volume*
4	Number of withdrawn NLRI prefixes	*volume*
5	Average AS-PATH length	*AS-path*
6	Maximum AS-PATH length	*AS-path*
7	Average unique AS-PATH length	*AS-path*
8	Number of duplicate announcements	*volume*
9	Number of duplicate withdrawals	*volume*
10	Number of implicit withdrawals	*volume*
11	Average edit distance	*AS-path*
12	Maximum edit distance	*AS-path*
13	Interarrival time	*volume*
14-24	Maximum edit distance = n, where $n = (7, ..., 17)$	*AS-path*
25-33	Maximum AS-path length = n, where $n = (7, ..., 16)$	*AS-path*
34	Number of IGP packets	*volume*
35	Number of EGP packets	*volume*
36	Number of incomplete packets	*volume*
37	Packet size (B)	*volume*

B. Support Vector Machine (SVM)

Support Vector Machine is a supervised learning model for classification and regression tasks. Given a set of labeled training samples, the SVM algorithm learns a classification hyperplane (decision boundary) by maximizing the minimum distance between data points belonging to various classes. There are two types of SVM models: hard-margin and soft-margin SVMs [11]. The hard-margin SVMs require that each data point is correctly classified, while the soft-margin SVMs allow some data points to be misclassified. In this paper, the soft-margin SVMs are utilized. The hyperplane is acquired by solving a loss function (1) with constraints (2) [5]:

$$C \times \Sigma_{n=1}^{N} \zeta_n + \frac{1}{2}||w|| \tag{1}$$

$$t_n y(x_n) \geqslant 1 - \zeta_n, n = 1, ..., N, \tag{2}$$

where parameter $C > 0$ controls the trade-off between the margin and the penalty term ($\frac{1}{2}||w||$), N is the number of data points, and ζ_n is the slack variable. t_n denotes the target value, $y(x_n)$ is the training model, and x_n are data points.

An illustration of the soft margin is shown in Fig. 1. The solid line indicates the decision boundary while dashed lines indicate the margins. Data points with circles are support vectors. The maximum margin is the perpendicular distance between the decision boundary and the closest support vectors. Data points for which $\zeta = 0$ are correctly classified and are either on the margin or on its correct side. Data points for which $0 \leq \zeta < 1$ are also correctly classified because they lie inside the margin and on the correct side of the decision boundary. Data points for which $\zeta > 1$ lie on the wrong side of the decision boundary and are misclassified [5]. The outputs 1 and -1 correspond to anomaly and regular data, respectively.

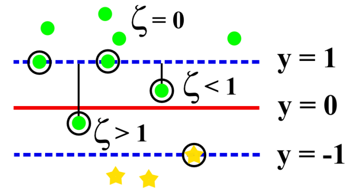

Fig. 1. Illustration of the soft margin SVM. Shown are correctly and incorrectly classified data points [5].

The SVM employs a kernel function to compute a non-linear separable function and maps the feature space into a linear space. We choose the Radial Basis Function because it creates a large function space and thus outperforms other types of SVM kernels :

$$K(u, v) = exp(-\lambda ||u - v||^2), \tag{3}$$

where u and v are dataset matrices and constant λ affects the number of support vectors.

Parameters C and λ are selected using 10-fold cross validation when generating the SVM models. We experiment with two libraries: libsvm-3.1 [12] and SVM^{light} [13]. The results presented in this paper are generated using SVM^{light}.

C. Long Short-Term Memory (LSTM) Neural Network

The LSTM approach employs a special form of the Recurrent Neural Networks (RNNs). Traditional RNNs are designed to store inputs in order to predict the outputs [14]. However, they perform poorly when several discrete time lags occur between the previous inputs and the present targets. Unlike the traditional RNNs, LSTM is capable of connecting time intervals to form a continuous memory [15]. The LSTM network is designed to overcome the vanishing gradient problem [16].

The LSTM implementation consists of an input layer, an LSTM layer, and an output layer. The input layer consists of 37 nodes that are inputs to the LSTM layer, each node corresponds to one feature. The output layer has one node, which is connected to the output of the LSTM layer. The output is labeled by 1 (anomaly) or -1 (regular). The LSTM layer consists of one LSTM cell, called the "memory block" [17]. It is composed of: (a) forget gate f_n, (b) input gate i_n, and (c) output gate o_n. The forget gate discards the useless memories according to the cell state, the input gate controls the information that will be updated in the LSTM cell, and the output gate works as a filter to control the output. The logistic sigmoid and network output functions are denoted by σ and $tanh$, respectively. An LSTM module is shown in Fig. 2.

III. EXPERIMENTAL PROCEDURE

In this paper, we consider both unbalanced and balanced training datasets. In the unbalanced datasets, the number of samples in the regular datasets is larger than in the anomalous datasets. To create the balanced datasets, we use all anomalies in each set and randomly select the same number of regular entries. We use the unbalanced and balanced data to generate

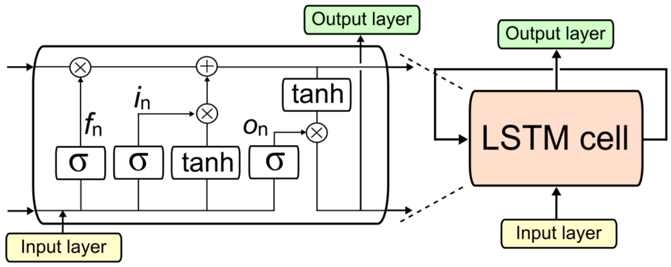

Fig. 2. Repeating modules for the LSTM neural network. Shown are the input layer, LSTM cell, and output layer.

the SVM and LSTM models and compare their performance. Unbalanced and balanced SVM (LSTM) datasets are denoted as SVM$_u$ (LSTM$_u$) and SVM$_b$ (LSTM$_b$), respectively. The three SVM and LSTM models are trained using datasets that contain anomalies. Datasets containing anomalies and regular data (BCNET and RIPE) are then used for testing the models. The classification procedure follows:

Step 1: Train and test the three SVM and LSTM models using 37 features.

Step 2: Select the 10 most relevant features using the three feature selection algorithms: MID, MIQ, and MIBASE. Train and test the three SVM models using datasets with and without anomalies. Skip this Step for generating the LSTM models.

Step 3: Evaluate the SVM and LSTM models using the accuracy and F-score measures.

Step 4: Tune the SVM and LSTM model parameters to achieve the best performance.

The three models are created using concatenations of two anomaly datasets, as shown in Table III. The concatenated training datasets consist of 14,400 ($2 \times 7,200$) data points represented by $14,400 \times 37$ and $14,400 \times 10$ matrices that correspond to 37 and 10 features, respectively. Rows correspond to data samples while columns represent features.

TABLE III. THE SVM AND LSTM TRAINING AND TESTING DATASETS

Model	**Training dataset**	**Testing dataset**
SVM_1 and LSTM_1	Slammer and Nimda	Code Red I
SVM_2 and LSTM_2	Slammer and Code Red I	Nimda
SVM_3 and LSTM_3	Nimda and Code Red I	Slammer

IV. CLASSIFICATION ENVIRONMENT

We use three feature selection algorithms (MID, MIQ, and MIBASE) [8], implemented in MATLAB, to minimize the dimension of the dataset matrix by selecting the 10 most relevant features.

We use the SVM^{light} [13] library developed in C language to classify BGP anomalies. SVM^{light} is an effective tool for classification, regression, and ranking when dealing with large training samples. The SVM^{light} library training and classification modules are used for training and testing SVM models. We tune the value of a parameter that controls the trade-off between the training error and the margin as well as the cost factor [14].

PyBrain [18], a modular Machine Learning Library for the Python language, is used as the LSTM classifier. The library is used for neural networks, unsupervised learning, and reinforcement machine learning. PyBrain [19] is used to generate LSTM models with 37-dimensional inputs, 1 hidden layer, and 1-dimensional outputs. We use 37 features because PyBrain already contains the feature selection function. We utilize the same combinations of datasets as in the case of SVM to generate three models: LSTM_1, LSTM_2, and LSTM_3. Models are trained using the BackpropTrainer built within the library. Parameters "momentum" and "learningrate" determine the direction and the step size of the learning movement in the gradient descent procedure, respectively. In addition to anomalous datasets, we also use regular datasets collected from RIPE [7] and BCNET [10] for testing.

V. PERFORMANCE EVALUATION

Classification algorithms are evaluated based on accuracy and F-score:

$$\text{accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (4)$$

$$\text{F-score} = 2 \times \frac{\text{precision} \times \text{sensitivity}}{\text{precision} + \text{sensitivity}}, \quad (5)$$

where

$$\text{precision} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{sensitivity} = \frac{TP}{TP+FN}. \quad (7)$$

These performance metrics are calculated based on confusion matrix shown in Table IV. The true positive (TP) and the true negative (TN) are anomalous and regular data points that are correctly classified as anomaly and regular while the false negative (FN) and false positive (FP) are anomalous and regular data points that are misclassified as regular and anomaly, respectively.

TABLE IV. CONFUSION MATRIX

	Predicted class	
Actual class	Anomaly (positive)	Regular (negative)
Anomaly (positive)	TP	FN
Regular (negative)	FP	TN

Accuracy, as a performance measure, reflects the true prediction over the entire dataset. It is commonly used in evaluating the classification performance. It gives the same importance to the regular and anomalous data. However, accuracy may be misleading in the case of unbalanced datasets. The F-score is important for anomaly prediction because it is based on both precision and sensitivity, which consider the false predictions. Precision measures the discrimination ability of the classifier to identify classified and misclassified anomalies. Sensitivity identifies correctly classified anomalies in the dataset.

A. SVM Performance

We use the SVM_2 model to compare results for unbalanced and balanced training datasets, as shown in Table V. For unbalanced training datasets, the features selected by the MIBASE algorithm generated the best F-score (69.97 %). The best F-score (72.32 %) is achieved using balanced training datasets containing 37 features. The best classification accuracy for balanced training datasets may decrease because of the small size of the training datasets.

TABLE V. ACCURACY AND F-SCORE USING THE SVM_2 MODELS FOR UNBALANCED AND BALANCED DATASETS

Unbalanced Datasets					
		Accuracy			**F-score**
		Testing Dataset	RIPE	BCNET	Testing Dataset
SVM_u2	37 Features	67.46 %	52.85 %	46.39 %	68.60 %
SVM_u2	MID	70.79 %	58.40 %	50.69 %	69.45 %
SVM_u2	MIQ	65.33 %	58.54 %	51.81 %	64.88 %
SVM_u2	MIBASE	66.74 %	53.40 %	48.33 %	69.97 %
Balanced Datasets					
		Accuracy			**F-score**
		Testing Dataset	RIPE	BCNET	Testing Dataset
SVM_b2	37 Features	69.26 %	51.81 %	44.86 %	72.32 %
SVM_b2	MID	60.96 %	55.35 %	54.31 %	63.36 %
SVM_b2	MIQ	51.89 %	32.43 %	43.68 %	62.63 %
SVM_b2	MIBASE	67.10 %	65.14 %	55.00 %	63.84 %

B. LSTM Performance

Results of the LSTM classification are shown in Table VI. When using unbalanced data, the highest F-score (54.87 %) is achieved by LSTM_u2 trained by using the combined Slammer and Code Red I datasets. For balanced datasets, the LSTM_b2 model achieves the best performance (58.16 %).

TABLE VI. ACCURACY AND F-SCORE USING LSTM MODELS FOR UNBALANCED AND BALANCED DATASETS

Unbalanced Datasets				
	Accuracy			**F-score**
	Testing Dataset	RIPE	BCNET	Testing Dataset
LSTM_u1	89.58 %	65.49 %	57.30%	23.33 %
LSTM_u2	60.00 %	51.53 %	50.80 %	54.87 %
LSTM_u3	63.15 %	56.74 %	58.55 %	24.68 %
Balanced Datasets				
	Accuracy			**F-score**
	Testing Dataset	RIPE	BCNET	Testing Dataset
LSTM_b1	45.04 %	60.48 %	62.78 %	16.72 %
LSTM_b2	63.16 %	44.27 %	53.58 %	58.16 %
LSTM_b3	61.24 %	55.00 %	48.20 %	27.48 %

The F-scores using the LSTM_1 and LSTM_3 models with both unbalanced and balanced datasets are below 30 %. We obtained similar results as in the case of the SVM classifier when using the same training datasets. Thus, we suspect that the poor performance may be caused by noisy data.

C. Performance Comparison

Performance of the SVM_2 and LSTM_2 models using unbalanced and balanced datasets is shown in Table VII.

TABLE VII. THE BEST ACCURACY AND F-SCORE OF SVM AND LSTM MODELS

	SVM_u2	**SVM_b2**	**LSTM_u1**	**LSTM_b2**
Accuracy	70.79 %	69.26 %	89.58 %	63.16 %
	SVM_u2	**SVM_b2**	**LSTM_u2**	**LSTM_b2**
F-score	69.97 %	72.32 %	54.87 %	58.16 %

The two best F-scores are achieved by SVM_b2 (72.32 %) and LSTM_b2 (58.16 %) that are trained using balanced datasets. Both SVM_2 models achieve higher F-scores than the LSTM_2 models. Using balanced datasets to train the SVM models leads to better F-scores than the results previously reported [3] that used unbalanced datasets (accuracy = 68.60 % and F-score = 22.20 %). This improvement is due to careful extraction of features from the BGP datasets as well as the use of balanced training datasets.

VI. CONCLUSION

We created the SVM and LSTM models for detecting BGP anomalies. The SVM_2 models based on the combination of the Slammer and Code Red I training datasets achieve better accuracy and F-score than results reported in the literature. The SVM classifier achieved the highest F-score using balanced datasets. In case of the unbalanced datasets, the accuracy is higher due to the large number of the regular testing data. Using the SVM classifier may be a feasible approach for detecting BGP anomalies in communication networks.

REFERENCES

[1] D. P. Watson and D. H. Scheidt, "Autonomous systems," *Johns Hopkins APL Technical Digest*, vol. 26, no. 4, pp. 368–376, Oct.–Dec. 2005.

[2] Y. Li, H. J. Xing, Q. Hua, X. Z. Wang, P. Batta, S. Haeri, and Lj. Trajković, "Classification of BGP anomalies using decision trees and fuzzy rough sets," in *Proc. IEEE Trans. Syst., Man, Cybern.*, San Diego, CA, USA, Oct. 2014, pp. 1331–1336.

[3] N. Al-Rousan and Lj. Trajković, "Machine learning models for classification of BGP anomalies," in *Proc. IEEE Conf. on High Performance Switching and Routing, HPSR 2012*, Belgrade, Serbia, June 2012, pp. 103–108.

[4] N. Al-Rousan, S. Haeri, and Lj. Trajković, "Feature selection for classification of BGP anomalies using Bayesian models," in *Proc. ICMLC 2012*, Xi'an, China, July 2012, pp. 140–147.

[5] C. M. Bishop, *Pattern Recognition and Machine Learning*. Secaucus, NJ, USA: Springer-Verlag, 2006.

[6] D. N. T. How, K. S. M. Sahari, Y. Hu, and C. K. Loo, "Multiple sequence behavior recognition on humanoid robot using long short-term memory (LSTM)," in *Proc. ICRA 2014*, Hong Kong, China, Dec. 2014, pp. 109–114.

[7] RIPE NCC: RIPE Network Coordination Center. [Online]. Available: https://www.ripe.net/ .

[8] mRMR (minimum Redundancy Maximum Relevance Feature Selection). [Online]. Available: http://penglab.janelia.org/proj/mRMR/ .

[9] University of Oregon Route Views Project. [Online]. Available: http://www.routeviews.org/ .

[10] BCNET. [Online]. Available: https://www.bc.net/ .

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] Libsvm. [Online]. Available: http://www.csie.ntu.edu.tw/ cjlin/libsvm/ .

[13] SVM^{light}: Support Vector Machines. [Online]. Available: http://svmlight.joachims.org .

[14] K. Morik, P. Brockhausen, and T. Joachims, "Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring," in *Proc. Int. Conf. on Machine Learning, ICML 1999*, Bled, Slovenia, June 1999, pp. 268–277.

[15] S. Hochreiter and J. Schmidhuber, "Long short–term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Oct. 1997.

[16] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992.

[17] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. INTERSPEECH*, Singapore, Sept. 2014, pp. 338–342.

[18] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber, "PyBrain," *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 743–746, Oct. 2010.

[19] PyBrain: The Python Machine Learning Library. [Online]. Available: http://pybrain.org/ .