Performance Evaluation of BGP Anomaly Classifiers

Marijana Ćosović and Slobodan Obradović
University of East Sarajevo
East Sarajevo, Bosnia and Herzegovina
Email: {marijana.cosovic, slobo.obradovic}@gmail.com

Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia, Canada
Email: ljilja@sfu.ca

***Abstract*—Changes in the network topology such as large-scale power outages or Internet worm attacks are events that may induce routing information updates. Border Gateway Protocol (BGP) is by Autonomous Systems (ASes) to address these changes. Network reachability information, contained in BGP update messages, is stored in the Routing Information Base (RIB). Recent BGP anomaly detection systems employ machine learning techniques to mine network data. In this paper, we evaluated performance of several machine learning algorithms for detecting Internet anomalies using RIB. Naive Bayes (NB), Support Vector Machine (SVM), and Decision Tree (J48) classifiers are employed to detect network traffic anomalies. We evaluated feature discretization and feature selection using three data sets of known Internet anomalies.**

Keywords—BGP; machine learning; naïve Bayes; support vector machine; decision tree.

I. INTRODUCTION

No single learning algorithm performs the best on all given classification tasks [1]. Hence, in each case an appropriate algorithm should be selected by evaluating its performance based on various parameters. Statistics, machine learning, and data mining have been employed to evaluate and compare various algorithms [2], [3]. Meta learning, a subfield of machine learning, deals with automatic detection of data models. We used the Waikato Environment for Knowledge Analysis (Weka) [4], an open source software tool distributed under GNU General Public License. It is a framework for implementation of machine learning algorithms.

Machine learning techniques have been recently employed in designing BGP anomaly detection systems [5]–[7]. In this paper, we use Naïve Bayes (NB), Decision Tree (J48), and SVM Radial Basis Function (RBF) kernel classifiers implemented in Weka (v. 3.7.11) and test their ability to reliably detect network anomalies. Performance of anomaly classifiers depends on feature selection algorithms [8]. Weka J48 classifier is an implementation of the C4.5 algorithm, which is widely used in data mining to build decision trees using information entropy. Weka also provides a wrapper to the LibSVM [9] library for Support Vector Machines (SVM).

The paper is organized as follows. In Section II, we describe extraction of BGP features. Data transformation that consists of feature discretization and feature selection is described in Section III. Classification models and their performance measures are discussed in Section IV. We conclude with Section V.

II. BGP DATA

Routing Information Service (RIS) project [10] began in 2001 by the Réseaux IP Européens (RIPE) Network Coordination Centre (NCC) to collect and store Internet routing data. Having chronological routing data is beneficial when detecting anomalies. Prior to July 2003, update messages were collected every fifteen minutes. The interval between consecutive messages was later decreased to five minutes. BGP update messages are collected by the Remote Route Collectors (RRCs) and stored in the multi-threaded routing toolkit (MRT) format [11].

We use BGP update messages that originated from AS 513 (route collector rrc04) member of CERN Internet Exchange Point (CIXP). We only consider data collected during the periods of the Internet anomalies. *Bgpdump* is a C library maintained by the RIPE NCC and it is used for analysis of dump files. We transform BGP update messages from MRT into ASCII format by using the *bgpdump* library on a Linux platform. A *bash* script was used to process data files in batches. Concatenation of messages is performed to optimize loading the database.

BGP update messages, originally stored in files, are loaded into the database using the Structured Query Language (SQL) loader tool. Database tables are created for the three well-known Internet attacks: Slammer [12], Nimda [13], and Code Red I [14]. Data are sequentially imported into the database. Details of the three anomalies are listed in Table I.

TABLE I. BGP DATASETS

	Number of events		Number of features	Number of classes
	Anomaly	Regular		
Slammer	869	6,331	15	2
Nimda	3,521	3,679	15	2
Code Red I	600	6,600	15	2

The SQL Slammer worm began infecting Microsoft SQL servers on January 25, 2003. The attack lasted 16 hours. Microsoft SQL servers were infected through a small piece of code that generated IP addresses at random. Furthermore, code replicated itself by infecting new machines through randomly generated targets. If the target was a Microsoft SQL server, it become infected and began infecting other servers. User PCs were infected if they had Microsoft SQL Server Data Engine

(MSDE) installed. The number of infected machines doubled approximately every nine seconds. Single infected machines have reported additional traffic of 50 Mb/sec [15] and that was a consequence of increased generation of update messages. The process caused a Denial of Service (DoS) attack. The Nimda worm was released on September 18, 2001. The attack lasted 59 hours. It exploited vulnerabilities in the Internet Information Services (IIS) web servers for the Internet Explorer 5. It used three main methods for propagation: email, network shares, and the web. The worm propagated by sending an infected attachment that was automatically downloaded after viewing email. A user could also download it from the website or access an infected file through the network. Although the Code Red I worm attacked Microsoft Internet Information Services (IIS) web servers earlier, the peak of infected computers was observed on July 19, 2001. The worm replicated itself by exploiting weakness of the IIS servers and, unlike the Slammer worm, Code Red I searched for vulnerable servers to infect. Rate of infection was doubling every 37 minutes, hence lower spreading rate.

These anomaly events affect performance of BGP [16]. We generated volume and AS-PATH features shown in Table II [5] by accessing the BGP update messages through querying the database. The AS-PATH features (5, 6, 7, 11, and 12) are derived from the AS-PATH attribute of BGP update messages. We filter data in the database to parse the ASCII files and generate feature statistics that are calculated every minute during a five-day period for each of the three attacks. More complex tasks required writing a PL/SQL code [17].

TABLE II. FEATURES EXTRACTED FROM BGP UPDATE MESSAGES

Feature	Name	Definition
1	*Number of announcements*	Number of routes that are available for delivery of the data.
2	*Number of withdrawals*	Number of routes that are no longer reachable.
3	*Number of announced Network Layer Reachability Information (NLRI) prefixes*	Number of announced NLRI prefixes within BGP update messages that have type field set to announcement during one-minute interval.
4	*Number of withdrawn Network Layer Reachability Information (NLRI) prefixes*	Number of withdrawn NLRI prefixes within BGP update messages that have type field set to withdrawal during one-minute interval.
5	*Average AS-PATH length*	Average length of AS-PATHs of all messages during one-minute interval.
6	*Maximum AS-PATH length*	Maximum length of AS-PATHs of all messages during one-minute interval.
7	*Average unique AS-PATH length*	Average of unique length of AS-PATHs of all messages during one-minute interval.
8	*Number of duplicate announcements*	Number of duplicate BGP update messages that have type field set to announcement during one-minute interval.
9	*Number of duplicate withdrawals*	Number of duplicate BGP update messages that have type field set to withdrawal during one-minute interval.
10	*Number of implicit withdrawals*	Number of BGP update messages that have type field set to announcement and different AS-PATH attribute for already announced NLRI prefixes during one-minute interval.
11	*Average edit distance*	Average of edit distances among all messages during one-minute interval.
12	*Maximum edit distance*	Average of edit distances among all messages during one-minute interval.
13	*Number of Exterior Gateway Protocol (EGP) packets*	Number of BGP update messages that are generated by EGP during one-minute interval.
14	*Number of Interior Gateway Protocol (IGP) packets*	Number of BGP update messages that are generated by IGP during one-minute interval.
15	*Number of incomplete packets*	Number of BGP update messages that are of unknown sources during one-minute interval.

III. DATA TRANSFORMATIONS

Several techniques may be used to make input data more responsive to machine learning algorithms. Feature discretization and feature selection are among the more effective approaches.

A. Feature Discretization

Discretization of numeric features is crucial if the classification involves numeric features but the chosen training algorithm may only handle categorical features [18]. Discretization is beneficial even for algorithms that may handle numeric features since they often produce better results or have faster convergence if features are pre-discretized.

Many discretization algorithms have been proposed in the literature [19]–[21]. They may process particular feature independently of others (univariate) or may process all features in the feature space (multivariate). The minimum description length (MDL) discretization method [22] is the most commonly used supervised discretization algorithm and has been implemented in Weka. Its advantage is clarity and superior performance [23].

B. Feature Selection

Feature selection is performed to minimize the number of features that a machine learning algorithm should consider by disregarding redundant or unrelated features. Performance of algorithms may be improved by using pre-selection of features [17]. Features may be selected manually or automatically.

Feature selection [24] is a process that reduces dimensionality of a design matrix. It is used to decrease computational complexity and memory usage. Feature selection reduces overfitting by minimizing redundant data, improving modeling accuracy, and decreasing training time. Weka *Attribute Selection* tool offers several feature selection algorithms. Feature selection process is comprised of two processes: *attribute subset evaluator* and *search method*. Feature subsets are evaluated using the evaluator while the search method facilitates search of all possible subsets.

IV. Performance Measures

Classification aims to identify a single class (anomaly). A classifier labels the instances as either anomaly or regular (not anomaly). A decision made by the classifier is represented by the confusion matrix shown in Table III:

- TP: number of anomalous training data points classified as anomaly
- FP: number of regular training data points classified as anomaly
- FN: number of anomalous training data points classified as regular
- TN: number of regular training data points classified as regular.

TABLE III. Confusion Matrix

		Predicted class	
		Anomaly	***Regular***
Actual class	***Anomaly (True)***	TP	FN
	Regular (False)	FP	TN

Classifier models are trained on a limited data set and tested on a separate set. Evaluation of a classifier is performed in order to check its ability to generalize. Accuracy and error rates are used to evaluate classifier's performance. Accuracy is the percent of correct classifications while the error rate is the percent of incorrect classifications.

We aim to classify anomaly class, which represents a smaller portion of training and testing datasets. Hence, we need to identify adequate performance indices that reflect the accuracy and precision of the classifier of anomaly testing data points. Note that accuracy as a performance measure assumes equal cost for misclassification and a relatively uniform class distribution. Therefore, using only accuracy as a measure may be misleading [25].

Performance of a classification model depends on a model's ability to correctly predict classes. A number of performance measures may be calculated:

- Recall is a ratio of identified anomalies (TP) and all labeled anomalies (true).
- Precision is a ratio of identified anomalies (TP) and all data points identified as anomalous.
- F-measure is often used as a performance index to compare performance of classification models. It is a harmonic mean of the recall and precision:

$$F-measure = 2 \times \frac{recall \times precision}{recall + precision}.$$

- The Matthews correlation coefficient (MCC) [26] is used for binary classification:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}.$$

  This is a balanced measure that may be used even if the classes are of a different size.

- Plots of True Positive Rate (TPR), as a function of False Positive Rate (FPR), for various parameters of a machine learning model are called receiver operating characteristics (ROCs). They may be misleading if the number of positive and negative instances greatly differ.

- Precision-recall (PR) curves are used in machine learning tasks if a class imbalance is present [27], [28].

In order to evaluate supervised discretization, we use Weka FilteredClassifier metaclassifier. It performs filtering on training data only while the test data are kept for evaluation. The process employs discretization intervals obtained from training data. The 10-fold stratified cross-validation, commonly used as a compromise to reduce the overfitting effect [29], is utilized for model evaluation. The ROC and PR curves for the NB classifier of Slammer anomaly with (NB-D) and without discretization (NB) are shown in Fig. 1 and Fig. 2, respectively. ROC and PR curves indicate that discretization of the features improves model performance measures.

The NB-1, J48-1, and SVM-1 models are NB, J48, and SVM classifiers trained on discretized data sets, respectively. The NB-2, J48-2, and SVM-2 models are classifiers trained on data sets with optimized F-measure. They are shown in Table IV.

Weka *Threshold Selector* metaclassifier has been used to optimize F-measure evaluation metrics by selecting a mid-point threshold of the probability output by the base classifiers (NB, J48, and SVM). A good F-measure requires correctly set threshold for the class probabilities. *Threshold Selector* relies on the probability estimates from base classifier so that the threshold of these probabilities may be optimized. In case of most Weka learning algorithms, probability estimates are generated by default. In case of LibSVM, a wrapper classifier that allows a third-party implementation of Support Vector Machines, the option to generate probability estimates for classification should be disabled. We measure performance by 10-fold stratified cross-validation.

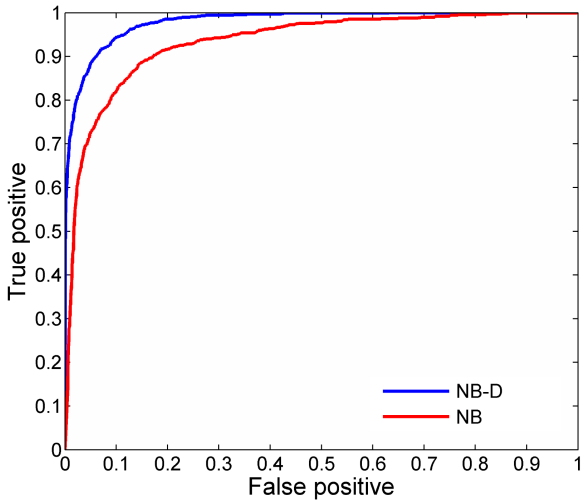

Fig. 1. Slammer: ROC curve for NB classifier with and without discretization indicate that discretization improves the shape of the ROC curve. An ideal ROC curve passes through coordinates (0, 0), (0, 1), and (1, 1).

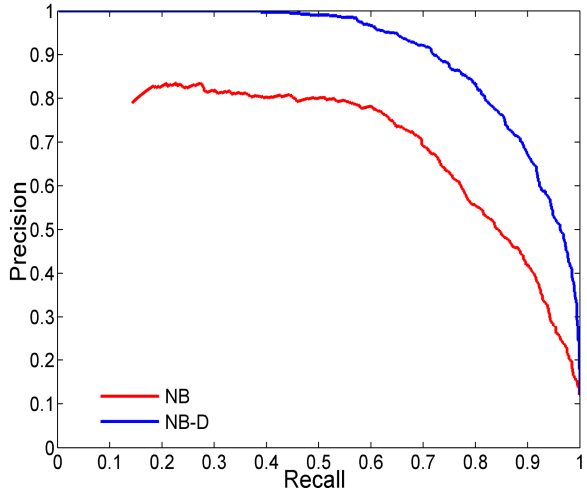

Fig. 2. Slammer: PR curve for the NB classifier with and without discretization indicate that discretization improves the shape of the PR curve. An ideal PR curve passes through coordinates (0, 1), (1, 1), and (1, 0).

Weka *AttributeSelectedClassifier* metaclassifier has been used to specify feature selection method and a learning algorithm as a part of the classification scheme. Two filter methods are used: *CfsSubsetEval* and *GainRatioAttributeEval*. We use the *GreedyStepWise* and *Ranker* search methods. *CfsSubsetEval* assesses feature subsets in a way to increase power of the individual features while at the same time minimizes redundancy between them. The feature subsets that have larger correlation factor with the class while at the same time having lower intercorrelation are desired.

GainRatioAttributeEval feature selection method selects features based on the information gain score. The higher the information gain score, the better is their discriminative power for classification. Hence, features are evaluated individually and ranked based on the scores returned by the evaluator. This is performed by a ranker search method.

Weka wrapper feature selection methods *ClassifierSubsetEval* and *WrapperSubsetEval* were used within *AttributeSelectedClassifier* metaclassifier to evaluate sets of relevant features. Both wrapper methods use a classifier to generate and evaluate sets of features from the training data. In case of *ClassifierSubsetEval*, a classifier is used as a parameter while *WrapperSubsetEval* employs 5-fold cross-validation to estimate the accuracy of the learning scheme for a subset of features [30]. These feature selection models are also shown in Table IV.

TABLE IV. CLASSIFIER MODELS

Classifiers	Methods	Models	Description
NB J48 SVM		NB-1 J48-1 SVM-1	Classifier trained on discretized data sets
		NB-2 J48-2 SVM-2	Classifier trained on data sets with F-measure optimized
	Filter	NB-3 J48-3 SVM-3	Correlation based feature subset evaluator (CfsSubsetEval) with Greedy Stepwise search method
		NB-4 J48-4 SVM-4	Gain ratio based feature evaluator with ranker for individual features
	Wrapper	NB-5 J48-5 SVM-5	Classifier subset evaluator using a classifier as a parameter for evaluation of sets of features on training data
		NB-6 J48-6 SVM-6	Wrapper subset evaluator using 5-folds cross-validation internally to estimate the accuracy of the learning scheme for a set of features

Performance measures for NB, J48, and SVM classifiers are shown in Table V, Table VI, and Table VII, respectively. NB-2 model shows improvements over NB-1 model in all performance measures for Slammer, Nimda, and Core Red I data sets. J48-1 classifier performs better in all performance measures for Slammer, Nimda and Code Red I data sets. SVM-1 classifier performs better on Slammer data set while SVM-2 performs better on Nimda and Code Red I data sets.

TABLE V. PERFORMANCE OF THE NB CLASSIFIER WITH DISCRETIZATION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	NB-1	0.767	0.741	0.980	0.907
	NB-2	0.807	0.781	0.980	0.906
Nimda	NB-1	0.745	0.493	0.826	0.817
	NB-2	0.758	0.483	0.826	0.816
Code Red I	NB-1	0.541	0.509	0.900	0.600
	NB-2	0.585	0.548	0.900	0.596

TABLE VI. PERFORMANCE OF THE J48 CLASSIFIER WITH DISCRETIZATION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	J48-1	0.844	0.825	0.967	0.879
	J48-2	0.826	0.802	0.966	0.876
Nimda	J48-1	0.755	0.518	0.815	0.774
	J48-2	0.753	0.485	0.814	0.773
Code Red I	J48-1	0.628	0.608	0.866	0.562
	J48-2	0.626	0.594	0.871	0.560

TABLE VII. PERFORMANCE OF THE SVM CLASSIFIER WITH DISCRETIZATION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	SVM-1	0.862	0.845	0.906	0.765
	SVM-2	0.855	0.837	0.980	0.926
Nimda	SVM-1	0.762	0.526	0.763	0.690
	SVM-2	0.767	0.506	0.844	0.825
Code Red I	SVM-1	0.564	0.542	0.729	0.372
	SVM-2	0.618	0.584	0.804	0.559

Subsets of features are evaluated based on NB-3, J48-3, and SVM-3 models are:

- Slammer: Features 1, 2, 3, 8, 11, 13, 14, and 15
- Nimda: Features 1, 3, 4, 14, and 15
- Code Red I: Features 1, 3, 4, 10, 13, and 14.

These volume features confirm that they are more relevant for identifying the anomaly class than AS-PATH features. A known effect of BGP anomalies is indeed a large number of BGP announcements.

Features according to NB-4, J48-4, and SVM-4 models are ranked as:

- Slammer: 1, 14, 3, 4, 15, 9, 8, 10, 12, 2, 6, 13, 7, 11, and 5
- Nimda: 3, 14, 1, 4, 9, 15, 2, 10, 8, 6, 12, 13, 7, 5, and 11
- Code Red I: 13, 1, 14, 3, 4, 10, 15, 2, 8, 6, 12, 5, 11, 7, and 9.

Selected features with wrapper models are shown in Table VIII.

TABLE VIII. FEATURES SELECTED USING WRAPPER MODELS

Data set	Models	Features selected
Slammer	NB-5	1, 2, 5, 11
	J48-5	1, 2, 3, 5, 6, 7, 8, 10, 11, 12, 14
	SVM-5	1, 2, 5, 7, 11, 15
	NB-6	1, 3, 5, 6, 7, 8, 11, 12, 14, 15
	J48-6	1, 2, 5, 10, 11
	SVM-6	1, 2, 5, 7, 11, 14, 15
Nimda	NB-5	1,4,7,13
	J48-5	1, 2, 3, 4, 5, 8, 9, 10, 11, 13, 14
	SVM-5	1, 3, 4, 5, 7, 11, 14, 15
	NB-6	1, 4, 11, 13
	J48-6	1, 2, 4, 10
	SVM-6	1
Code Red I	NB-5	4,5,13
	J48-5	2, 3, 4, 10, 11, 13
	SVM-5	1, 2, 3, 4, 5, 8, 11, 13, 15
	NB-6	1, 4
	J48-6	2, 4, 5, 7, 11, 13
	SVM-6	1, 2, 3, 4, 5, 8, 11, 13

Performance measures of NB, J48, and SVM classifiers trained on Slammer, Nimda, and Code Red I data sets are shown in Tables IX, Table X, and Table XI, respectively. The recall rates evaluated for the classifiers are shown in Table XII. Four tests were performed for each classifier on three different data sets. Wrapper methods for feature selection provide better results than filter methods for Slammer and Code Red I data sets. They measure the quality of feature subsets by building and evaluating an actual classification model, hence their better performance. In case of Nimda data set, wrapper methods for feature selection slightly outperform filter methods. Performance margin is smaller than in case of Slammer and Code Red I data sets.

The NB-5 classifier model achieves the best performance on Nimda data set with four features selected. The J48-6 classifier model performs the best on Code Red I data set with six features selected. SVM-6 classifier model achieves the highest performance measures (F-measure, recall rate, and MCC) on Slammer data set with the reduced number (from fifteen to eight) of selected features.

TABLE IX. PERFORMANCE OF NB CLASSIFIER FEATURE SELECTION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	NB-3	0.780	0.754	0.980	0.898
	NB-4	0.767	0.741	0.980	0.907
	NB-5	0.830	0.808	0.981	0.910
	NB-6	0.829	0.807	0.981	0.908
Nimda	NB-3	0.752	0.509	0.834	0.824
	NB-4	0.745	0.493	0.826	0.817
	NB-5	0.754	0.514	0.836	0.820
	NB-6	0.754	0.516	0.832	0.818
Code Red I	NB-3	0.571	0.534	0.897	0.608
	NB-4	0.541	0.509	0.900	0.600
	NB-5	0.625	0.599	0.902	0.624
	NB-6	0.634	0.605	0.902	0.631

TABLE X. PERFORMANCE OF J48 CLASSIFIER FEATURE SELECTION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	J48-3	0.847	0.827	0.940	0.854
	J48-4	0.853	0.834	0.921	0.816
	J48-5	0.848	0.828	0.932	0.805
	J48-6	0.856	0.839	0.929	0.850
Nimda	J48-3	0.760	0.518	0.810	0.767
	J48-4	0.731	0.476	0.756	0.696
	J48-5	0.734	0.482	0.758	0.695
	J48-6	0.752	0.514	0.812	0.776
Code Red I	J48-3	0.587	0.570	0.809	0.541
	J48-4	0.631	0.613	0.799	0.505
	J48-5	0.635	0.617	0.789	0.502
	J48-6	0.656	0.639	0.795	0.551

TABLE XI. PERFORMANCE OF SVM CLASSIFIER FEATURE SELECTION

Data set	Model	F-measure	MCC	ROC	PR
Slammer	SVM-3	0.828	0.812	0.873	0.721
	SVM-4	0.856	0.843	0.886	0.766
	SVM-5	0.878	0.866	0.904	0.799
	SVM-6	0.880	0.867	0.907	0.800
Nimda	SVM-3	0.667	0.444	0.713	0.660
	SVM-4	0.684	0.423	0.709	0.646
	SVM-5	0.660	0.434	0.708	0.655
	SVM-6	0.661	0.435	0.709	0.655
Code Red I	SVM-3	0.568	0.566	0.716	0.396
	SVM-4	0.622	0.629	0.738	0.465
	SVM-5	0.623	0.624	0.743	0.459
	SVM-6	0.632	0.631	0.748	0.468

TABLE XII. RECALL RATES FOR NB, J48, AND SVM CLASSIFIERS

Data set	Model	Recall	Model	Recall	Model	Recall
Slammer	NB-1	0.902	J48-1	0.808	SVM-1	0.825
	NB-2	0.793	J48-2	0.852	SVM-2	0.824
	NB-3	0.898	J48-3	0.823	SVM-3	0.756
	NB-4	0.902	J48-4	0.834	SVM-4	0.777
	NB-5	0.806	J48-5	0.826	SVM-5	0.812
	NB-6	0.817	J48-6	0.817	SVM-6	0.820
Nimda	NB-1	0.757	J48-1	0.758	SVM-1	0.777
	NB-2	0.857	J48-2	0.814	SVM-2	0.858
	NB-3	0.763	J48-3	0.781	SVM-3	0.580
	NB-4	0.757	J48-4	0.726	SVM-4	0.640
	NB-5	0.763	J48-5	0.731	SVM-5	0.574
	NB-6	0.761	J48-6	0.754	SVM-6	0.574
Code Red I	NB-1	0.732	J48-1	0.548	SVM-1	0.477
	NB-2	0.588	J48-2	0.603	SVM-2	0.613
	NB-3	0.690	J48-3	0.492	SVM-3	0.443
	NB-4	0.732	J48-4	0.542	SVM-4	0.483
	NB-5	0.563	J48-5	0.545	SVM-5	0.493
	NB-6	0.595	J48-6	0.565	SVM-6	0.503

V. CONCLUSION

In this paper, we have investigated performance measures of BGP detection models based on Naïve Bayes, SVM-RBF kernel, and Decision Tree J48 classifiers. Slammer, Nimda and Code Red I data sets are examples of known anomalies that confirmed useful for developing of anomaly detection algorithms. We have analyzed effects of feature discretization and feature selection using both filter and wrapper methods on given data sets in order to improve classification performance. *Threshold Selector* metaclassifier was used for optimization of F-measure. Performance of the classifiers is influenced by the data set employed. No single classifier performs the best across all given data sets. Experiments are performed using Weka and the stratified 10-fold cross validation. The best performance measures are achieved using the SVM-RBF kernel classifier with wrapper feature selection method on the Slammer data set.

REFERENCES

[1] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural Computation*, vol. 8, no. 7, pp. 1341–1390, Oct. 1996.

[2] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1924, Oct. 1998.

[3] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learning Res.*, vol. 7, pp. 1–30, Jan. 2006.

[4] (2015, Jan.) Weka 3: Data Mining Software in Java [Online]. Available: http://www.cs.waikato.ac.nz/ml/weka/index.html.

[5] N. Al-Rousan and Lj. Trajković, "Machine learning models for classification of BGP anomalies," in *Proc. 13th IEEE Int. Conf. High Performance Switching and Routing*, Belgrade, Serbia, June 2012, pp. 103–108.

[6] N. Al-Rousan, S. Haeri, and Lj. Trajković, "Feature selection for classification of BGP anomalies using Bayes models," in *Proc. Int. Conf. Mach. Learning Cybern.*, Xi'an, China, July 2012, pp. 140–147.

[7] 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 Int. Conf. Syst., Man, Cybern.*, San Diego, CA, USA, Oct. 2014, pp. 1331–1336.

[8] G. H. John, R. Kohavi, and K. Peger, "Irrelevant features and the subset selection problem," in *Proc. 11th Int. Conf. Mach. Learning*, New Brunswick, NJ, USA, July 1994, pp. 121–129.

[9] (2015, Jan.) LibSVM - a library for support vector machines [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[10] (2015, Jan.) RIPE RIS raw data [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data.

[11] T. Manderson, "Multi-threaded routing toolkit (MRT) Border Gateway Protocol (BGP) routing information export format with geo-location extensions," RFC 6397, *IETF*, [Online]. Available: http://www.ietf.org/rfc/rfc6397.txt.

[12] (2015, Jan.) Center for Applied Internet Data Analysis. The Spread of the Sapphire/Slammer Worm [Online]. Available: http://www.caida.org/publications/papers/2003/sapphire/.

[13] (2015, Jan.) Sans Institute. Nimda worm—why is it different? [Online]. Available: http://www.sans.org/reading-room/whitepapers/malicious/nimda-worm-different-98.

[14] (2015, Jan.) Sans Institute. The mechanisms and effects of the Code Red worm [Online]. Available: https://www.sans.org/reading-room/whitepapers/dlp/mechanisms-effects-code-red-worm-87.

[15] (2015, Jan.) Sans Institute. Malware FAQ: MS-SQL Slammer [Online]. Available: https://www.sans.org/security-resources/malwarefaq/ms-sql-exploit.php.

[16] M. Ćosović, S. Obradović and Lj. Trajković, "Algorithms for investigation of abnormal BGP events," in *Proc. Int. Scientific Conf. UNITECH,* Gabrovo, Bulgaria, Nov. 2013, no. 2, pp. 253–257.

[17] M. Ćosović, S. Obradović, and Lj. Trajković, "Using databases for a BGP data analysis," in *Proc. Int. Scientific Conf. UNITECH,* Gabrovo, Bulgaria, Nov. 2014, no. 2, pp. 367–370.

[18] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011, pp. 314–322.

[19] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. 12th Int. Conf. Mach. Learning*, Tahoe City, CA, USA, July 1995, pp. 194–202.

[20] H. Liu, F. Hussain, C. L. Tan, and M. Dash, "Discretization: An enabling technique," *Data Mining and Knowledge Discovery,* vol. 6, no. 4, pp. 393–423, Oct. 2002.

[21] Y. Yang and G. I. Webb, "A comparative study of discretization methods for naive-Bayes classifiers," in *Proc. Pacific Rim Knowledge Acquisition Workshop*, Tokyo, Japan, Aug. 2002, pp. 159–173.

[22] U. M. Fayyad and K. B. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proc. 13th Int. Joint Conf. Artificial Intell.*, San Francisco, CA, USA, Sept. 1993, pp. 1022–1027.

[23] J. Kujala and T. Elomaa, "Improved algorithms for univariate discretization of continuous features," in *Proc. 11th European Conf. on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Poland, Sept. 2007, pp. 188–199.

[24] M. Ćosović, S. Obradović, and Lj. Trajković, "Feature selection techniques for machine learning," in *Proc. Int. Scientific Conf. UNITECH,* Gabrovo, Bulgaria, Nov. 2013, no. 1, pp. 85–89.

[25] F. Provost, T. Fawcett, and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," in *Proc. 15th Int. Conf. Mach. Learning*, Madison, WI, USA, July 1998, pp. 445–453.

[26] B. W. Matthews, "Comparison of the predicted and observed secondary structure of T4 phage lysozyme," *Biochimica et Biophysica Acta* (*BBA*) - *Protein Structure,* vol. 405, no. 2, pp. 442–451, Oct. 1975.

[27] P. Singla and P. Domingos, "Discriminative training of Markov logic networks," in *Proc. 20th Nat. Conf. Artificial Intell.*, Pittsburgh, PA, USA, July 2005, pp. 868–873.

[28] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proc.23rd Int. Conf. Mach. Learning*, Pittsburgh, PA, USA, June 2005, pp. 233–240.

[29] G. J. McLachlan, K. A. Do, and C. Ambroise, *Analyzing Microarray Gene Expression Data*. Hoboken, NJ, USA: John Wiley & Sons, 2004, pp. 31–61.

[30] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intell.*, vol. 97, no. 1–2, pp. 273–324, Dec. 1997.