ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)

https://doi.org/10.17559/TV-20170219114900
Original scientific paper

BGP Anomaly Detection with Balanced Datasets

Marijana ĆOSOVIĆ, Slobodan OBRADOVIĆ

Abstract: We use machine learning techniques to build predictive models for anomaly detection in the Border Gateway Protocol (BGP). Imbalanced datasets of network anomalies pose limitations to building predictive models for anomaly detection. In order to achieve better classification performance measures, we use resampling methods to balance classes in the datasets. We use undersampling, oversampling and combination techniques to change class distributions of the datasets. In this paper we build predictive models based on preprocessed network anomaly datasets of known Internet network anomalies and observe improvement in classifier performance measures compared to those reported in our previous work. We propose to use resampling combination techniques on datasets along with Decision Tree and Naïve Bayes classifiers in order to achieve the best trade-off between (1) the F-measure and the length of model training time, and (2) avoiding overfitting and loss of information.

Keywords: anomaly detection; BGP; classification; sampling techniques

1 INTRODUCTION

Machine learning techniques have been used by the research community to build predictive classifiers for detection of Border Gateway Protocol (BGP) anomalies [1-3]. Different classes, anomalous and regular, within datasets used to construct classifiers are usually not equally represented. Such imbalanced datasets of network anomalies pose limitations to building accurate predictive models [4]. Work [3] has shown improvements in performance measures using balanced datasets. In order to achieve better classification performance measures, we use known methods to balance classes in the datasets. We use algorithms implemented in the Python imbalanced-learn (v. 0.1.6) package [5] to create oversampled, undersampled and combination of oversampled and undersampled datasets based on datasets of known anomalies.

Oversampling techniques create additional instances of the anomalous class, whereas undersampling techniques remove some regular class instances from the datasets. We do not incur any loss of information in oversampling techniques, although the size of datasets and the time needed for building a classifier increases. On the other hand, loss of information could be present in undersampling techniques. Resampling combination techniques combine over- and undersampling methods to attain better classification accuracy [4]. We use classifiers [1] built on original imbalanced datasets and compare their performance with classifiers built on balanced datasets to determine the most reliable method for detecting network anomalies. Our overall purpose is to build more accurate predictive models than current ones for anomaly detection in the BGP.

We use automatic detection of data models, meta learning, implemented in the 3.9.0 version of Waikito Environment for Knowledge Analysis (Weka) [6]. Support vector machines (SVM), Decision Trees (J48), and Naïve Bayes (NB) are used as base classifiers within Filter and Wrapper classifiers implemented in Weka. We test the implementation framework of machine learning techniques on datasets of anomalies such as worms [7-9], large-scale power outages [10], and BGP router configuration errors [11].

This paper is organized as follows. In Section 2, we describe the BGP data. Data resampling techniques (oversampling, undersampling, and resampling combination techniques), classification models and their respective performance measures, and feature selection are discussed in Section 3. Classification results are described in Section 4. We draw conclusions in Section 5.

2 BGP DATA

Routing data are obtained from the Routing Information Service (RIS) project [12] originated by the Réseaux IP Européens (RIPE) Network Coordination Centre (NCC). These data have been collected since 2001. Chronological data collected regularly (every five minutes since 2003) can contribute to research of anomalous events. We are interested in BGP update messages since they carry network reachability information between autonomous systems that in times of intentional or unintentional changes in network topology respond with an increased number of routing updates. Update messages are stored in MRT [13] format and are transformed to ASCII by using the *bgpdump* library written in C and maintained by the NCC. We load BGP update messages of known network anomalies into a database, and by using SQL queries we extract features that are of interest to us. It is known that changes in network topology are highly correlated with volume and AS-PATH features. We extract the fifteen volume and AS-PATH features shown in Tab. 1 by querying the database on the minute-level. Hence, for every event considered, we created a feature matrix that is 7200×15; there are 7200 minute-level time instances over 5 days. Anomalous samples are recorded throughout the duration of the well documented anomalous events [7-11] and labeled accordingly. The rest of the samples belong to the regular class.

Fig. 1 shows increase in number of announced messages due to BGP anomalies: Slammer and Code Red I worm, Moscow Power Blackout (MPB) and BGP router configuration error (AS 9121 Routing Table Leak (RTL)).

As we can observe in Tab. 2, in the case of the Slammer dataset, out of 7200 instances in the feature matrix only 869 instances belong to the anomalous class. The Slammer dataset has a class distribution in favor of regular events; the imbalance ratio (ratio of regular to anomalous instances) is more than 7:1. In the case of the MPB dataset, the ratio is more than 41:1. Imbalance ratios of the Code Red I and RTL datasets are in between those values.

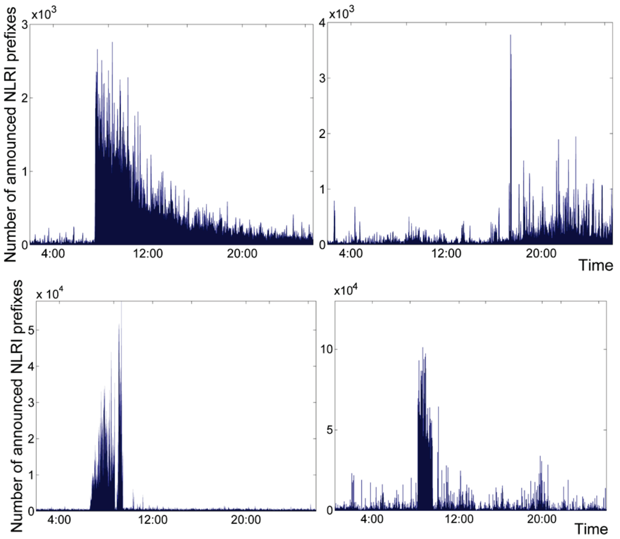

Figure 1 Number of announced Network Layer Reachability Information (NLRI) prefixes during anomalous events: Slammer worm (top left), Code Red I worm (top right), Moscow Power Blackout (bottom left), AS 9121 Routing Table Leak (bottom right).

Table 1 Extracted features from BGP update messages

Feature	
1	Number of announcements (*Ann No*)
2	Number of withdrawals (*With No*)
3	Number of announced NLRI prefixes (*Ann IP No*)
4	Number of withdrawn NLRI prefixes (*With IP No*)
5	Average AS-PATH length
6	Maximum AS-PATH length
7	Average unique AS-PATH length
8	Number of duplicate announcements (*Duplicate Ann*)
9	Number of duplicate withdrawals (*Duplicate With*)
10	Number of implicit withdrawals
11	Average edit distance (*AVG ED*)
12	Maximum edit distance (*MAX ED*)
13	Number of Exterior Gateway Protocol (EGP) packets (*EGP No*)
14	Number of Interior Gateway Protocol (IGP) packets (*IGP No*)
15	Number of incomplete packets (*Incomplete No*)

Table 2 BGP known network anomalies

Dataset	Regular Class	Anomaly Class	Number of features
Slammer	6331	869	15
Nimda	3679	3521	15
Code Red I	6600	600	15
MPB	7031	169	15
RTL	6900	300	15

Classifiers on both of these datasets are expected to perform much better in terms of performance measures after we balance the regular and anomalous events. On the other hand, the imbalance ratio of the Nimda dataset is close to 1; hence, we do not expect the resampled datasets to improve classifier performance greatly for Nimda.

3 RESEARCH METHODOLOGY

The flow chart of our research methodology is shown in Fig. 2. Primary goals of our project are: (1) decide whether to balance the datasets, (2) investigate oversampling, undersampling and resampling combination techniques, and (3) perform classification on balanced datasets using various classifiers in order to find classification methods with improved performance measures. Details of these steps are provided in subsequent sections.

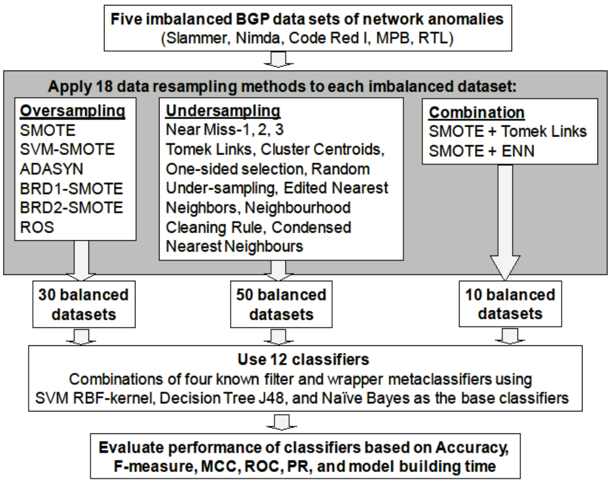
Five imbalanced BGP data sets of network anomalies
(Slammer, Nimda, Code Red I, MPB, RTL)
Apply 18 data resampling methods to each imbalanced dataset:
Oversampling
SMOTE
SVM-SMOTE
ADASYN
BRD1-SMOTE
BRD2-SMOTE
ROS
Undersampling
Near Miss-1, 2, 3
Tomek Links, Cluster Centroids, One-sided selection, Random Under-sampling, Edited Nearest Neighbors, Neighbourhood Cleaning Rule, Condensed Nearest Neighbours
Combination
SMOTE + Tomek Links
SMOTE + ENN
30 balanced datasets
50 balanced datasets
10 balanced datasets
Use 12 classifiers
Combinations of four known filter and wrapper metaclassifiers using SVM RBF-kernel, Decision Tree J48, and Naïve Bayes as the base classifiers
Evaluate performance of classifiers based on Accuracy, F-measure, MCC, ROC, PR, and model building time

Figure 2 Research Methodology

3.1 Data Resampling

In the case of imbalanced datasets, classifiers often misclassify instances of the anomaly class because the classifiers are biased towards the regular class. The research community has provided many solutions to address this issue. One of the methods of achieving better classifier performance for imbalanced datasets is data resampling [6].

Classifiers built on datasets that have balanced class distributions perform better than classifiers built on imbalanced datasets. In this paper, we use oversampling, undersampling, and resampling combination techniques to create 90 balanced datasets from the original Slammer,Nimda, Code Red I, Moscow Power Blackout, and AS 9121 Routing Table Leak datasets (Fig. 1, [1] and [2]).

We use several oversampling methods.We consider the Synthetic minority oversampling technique (SMOTE) [14], Support Vector Machine (SVM)-SMOTE [15], Adaptive Synthetic Sampling (ADASYN) [16], Borderline1 (BRD1) – SMOTE [17], Borderline2 (BRD2) – SMOTE [17], and Random oversampling (ROS) methods to create balanced datasets. In the case of the resampled Slammer datasets shown in Tab. 3, anomalous and regular classes are equal in the cases of SMOTE, BRD1-SMOTE, BRD2-SMOTE and ROS datasets, and almost equal in the case of two other Slammer datasets. The resampled Code Red Idatasets and their respective imbalance ratios are shown in Tab. 4.

ROS is a method that creates additional instances of the anomaly class needed for balancing the dataset randomly. Since the additional data instances in the ROS dataset are randomly copied instances from the original dataset, there is a great chance of overfitting.

SMOTE is an oversampling method that creates additional instances without causing overfitting. This is because the algorithm creates artificial anomaly class samples along the connections between the anomaly class samples and their nearest neighbours belonging to the same class.

Borderline SMOTE is an oversampling method based on SMOTE in which artificial anomaly class instances are created only in the borderline area. The BRD1-SMOTE algorithm creates artificial samples along the borderline between anomaly class samples and the nearest neighbours belonging to the same class while the BRD2-SMOTE algorithm in addition creates artificial samples along the borderline between anomaly class samples and positive nearest neighbours in the anomaly class, as well as negative nearest neighbours in the regular class.

Table 3 Imbalance Ratio for Oversampling Methods (OSM) – Slammer dataset

Dataset	Regular Class	Anomaly Class	Imbalance Ratio (IR)
Original	6331	869	7,285
SMOTE	6331	6331	1
SVM-SMOTE	6330	6331	0,999
ADASYN	6331	6343	0,998
BRD1-SMOTE	6331	6331	1
BRD2-SMOTE	6331	6331	1
ROS	6331	6331	1

Table 4 Imbalance Ratio for Oversampling Methods (OSM) – Code Red I dataset

Dataset	Regular Class	Anomaly Class	Imbalance Ratio (IR)
Original	6600	600	11
SMOTE	6600	6600	1
SVM-SMOTE	6660	6600	1
ADASYN	6600	6676	0,988
BRD1-SMOTE	6600	6599	0,999
BRD2-SMOTE	6600	6599	0,999
ROS	6600	6600	1

ADASYN is a sampling method that adaptively learns from imbalance in the dataset. The algorithm creates more synthetic instances of the anomaly class for those samples that are more difficult to learn and fewer instances for samples that are easier to learn. Hence, it has a flexible decision boundary.

Undersampling methods we consider in this paper are Near Miss-1, Near Miss-2, Near Miss-3 [18], Tomek Links [19], Cluster Centroids [20], One-sided selection [21], Random undersampling, Edited Nearest Neighbours [22], Neighbourhood Cleaning Rule [23], and Condensed Nearest Neighbours [24].

The undersampling methods NearMiss-1, NearMiss-2, and NearMiss-3 use the K-nearest neighbor (KNN) algorithm to create undersampled datasets. The NearMiss-1 method picks the regular class samples with the smallest average distance to the three closest anomaly class samples. Hence, it selects regular class samples that are close to some of the anomaly class samples. The NearMiss-2 method picks the regular class samples with the smallest average distance to the three furthest anomaly class samples. Hence, it selects regular class samples that are close to all of the anomaly class samples. The NearMiss-3 method picks a certain number of regular class samples for each of the anomaly class samples in order to have the anomaly class samples surrounded by regular class samples. In the case of the NearMiss-1 and NearMiss-2 methods applied to the Slammer dataset, we obtain a balanced dataset with 869 each of regular and anomaly data samples, while in the case of the NearMiss-3 method, the number of regular data samples drops down to 664.

The Tomek links method is a cleaning undersampling method in which a pair of samples that form a Tomek link (samples must belong to different classes and they are each other's nearest neighbour) are either near the border or one of the samples is noise. Hence, this method is used to remove noisy and borderline samples where clean clusters of classes are formed by removing all Tomek links that have nearest neighbour pairs with minimum distance between them belonging to different classes. In the case of Tomek links applied to the Slammer dataset, only 51 samples belonging to the regular class have been removed.

The Cluster centroid undersampling method separates data into two sets of samples based on class, namely, regular and anomaly samples. The regular class samples are clustered into K clusters and are replaced by the cluster centroid of the K-Means algorithm. Each cluster is combined with the anomaly class samples to form a K combination of datasets. The dataset that has the highest accuracy is chosen. In the case of the Cluster centroid applied to the Slammer dataset, we derive a regular-to-anomaly class ratio of 1:1.

The One-sided selection undersampling method is a special case of the Tomek links method in which only samples from the regular class are removed while the samples from the anomaly class are left intact.

The Edited Nearest Neighbour (ENN) method removes the samples from the dataset that do not agree with the majority of their k nearest neighbours. Usually, k is chosen to be three; hence, the ENN method of data undersampling removes samples that are different from two (the majority) of their three nearest neighbours.

The Neighbourhood Cleaning Rule (NLC) method is based on ENN. The original dataset is divided into regular and anomaly classes. Two sets of misclassified samples are formed and removed from the original dataset: (1) noisy data in the regular class are identified using ENN, and (2) samples that are different from two (the majority) of their three nearest neighbours, as well as different from samples that are misclassified using their three nearest neighbours and that belong to the anomaly class.

Condensed Nearest Neighbours (CNN) rule is based on reducing the size of a training set by finding a subset such that every sample in the subset is closer to a sample of the same class than it is to a sample of a different class. CNN is used to remove samples from the regular class that are far away from the decision boundary.

The combination of over- and undersampling methods for balancing datasets that we use in this paper are SMOTE + Tomek Links [25] and SMOTE+ENN [26].

3.2 Classification Models and Performance Measures

We use four models of Filter and Wrapper classifiers developed in Weka (v. 3.9.0), and for each of these classifiers we use SVM RBF-kernel, Decision Tree J48, and Naïve Bayes as a base classifier, thereby creating twelve different classifiers. *AttributeSelectedClassifier* is a metaclassifier implemented in Weka that internally performs three procedures to perform classification: searching, evaluating, and classifying selected samples. The first two processes perform feature selection; feature subsets are evaluated and the space of all possible subsets is searched. Feature selection is done in general to reduce training time, reduce overfitting, and increase accuracy by removing redundant features. The two filter methods we use for feature selection are *CfsSubsetEval* with the *GreedyStepWise* search method and *GainRatioAttributeEval* combined with the *Ranker* search method. The first method chooses subsets that have a large correlation with the class and low correlation with each other. The second method selects features in accordance with the information gain score; a higher score implies better discrimination ability for classification. The two wrapper methods used within metaclassifiers for evaluation of relevant subsets are Weka's *ClassifierSubsetEval* and *WrapperSubsetEval.*

Accuracy as defined in the equation below can be a misleading performance measure when assessing classifiers applied to imbalanced datasets. However, imbalance ratios in datasets considered in this paper are greatly improved by using resampling methods. Accuracy, F-measure, the Matthews correlation coefficient (MCC) [27], area under a receiver operating characteristic curve (ROC), and area under a precision-recall (PR) curve [28] are used to evaluate the performance of the classifiers. Because the resampled datasets are balanced, accuracy and F-measure values are very similar for a given classifier. The differences that occur are assumed to be attributed to datasets that are not perfectly balanced and rounding errors. During the classification process, a classifier identifies samples as either regular or anomalous. A confusion matrix defines the number of anomalous training data points classified as anomalous as true positive (*TP*), the number of regular training data points classified as anomalous as false positive (*FP*), the number of anomalous training data points classified as regular as false negative (*FN*), and the number of regular training data points classified as regular as true negative (*TN*).

Accuracy is defined as:

$$Acc = \frac{TP + TN}{TP + FN + FP + TN}$$

The F-measure is defined as the harmonic mean of recall and precision:

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

where recall is the ratio of identified anomalies to all labeled anomalies (*TP*/(*TP*+*FN*)), and precision is the ratio of identified anomalies to all anomalous data points (*TP*/(*TP*+*FP*)).

The Matthews correlation coefficient (*MCC*) used for binary classification is defined as:

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

Table 5 Features (Table 1) selected for the best performing classifier models with their respective oversampled datasets

Dataset	Model	Features selected
Slammer ROS	J48-4	1, 14, 3, 4, 15, 8, 10, 2, 12, 6, 9, 11, 5, 13, 7.
	J48-5	1, 3, 5, 6, 7, 8, 10, 11.
	J48-6	1, 2, 5, 7, 8, 10, 11, 15.
Slammer BRD2 SMOTE	NB-6	1, 2, 5, 11, 12, 15.
Code Red I ROS	J48-4	13, 1, 4, 3, 9, 10, 15, 2, 11, 8, 12, 5, 6, 7, 14.
	J48-5	1, 3, 5, 6, 11, 14.
Code Red I BRD2 SMOTE	NB-3	1, 2, 3, 4, 6, 8, 10, 11, 13, 14, 15.
	NB-6	4, 6, 11, 13, 15.
MPB ROS	J48-3	1, 3, 5, 8, 10, 13, 14, 15.
	J48-4	3, 14, 1, 8, 15, 10, 5, 13, 7, 6, 4, 9, 2, 11, 12.
	J48-5	3, 4, 5, 8.
	J48-6	3,4,5,7,8,10.
RTL BR2 SMOTE	NB-5	1, 10, 11, 15.
	NB-6	1, 2, 11, 15.
RTL ROS	J48-5	1, 8, 10, 11.
	J48-6	2, 8, 11.

3.3 Feature Selection

Features in Tab. 5 and Tab. 6 are shown for the models that produce the best performance based on F-measure with their respective oversampled and undersampled datasets. In cases where the F-measure was the same for more than one model, we choose a model with shorter model training time. Models J48-4, and SVM-4 rank all of the features, which are listed in Tab. 5 and 6 in order from highest to lowest gain ratio, whereas other models select a subset of relevant features according to criteria introduced in subsection 3.2, Classification models.

Table 6 Features selected for the best performing classifier models with their respective undersampled datasets

Dataset	Model	Features selected
Slammer RUS	J48-3	1, 2, 7, 12, 14, 15.
	NB-5	1, 11.
	SVM-6	1, 2, 7.
	NB-6	1, 2, 7, 11.
Code Red I Near Miss-1	SVM-3	1, 2, 3, 4, 7, 8, 12, 13, 14, 15.
	SVM-4	1, 13, 3, 4, 2, 15, 10, 14, 8, 6, 9, 12, 7, 5,11.
	SVM-6	1, 2, 3, 4, 5, 6, 8, 11, 15.
	NB-6	3, 4, 5, 6, 7, 8, 11, 13, 15.
MPB Near Miss-1	SVM-3	3, 4, 5, 6, 7, 8, 12.
	SVM-4	8, 3, 7, 14, 1, 10, 15, 5, 13, 6, 2, 9, 4, 11, 12.
	SVM-5	1, 2, 7.
	SVM-6	1, 2, 6.
RTL RUS	NB-3	1,2,3,8,10.
	NB-4	2, 10, 1, 14, 8, 15, 7, 4, 5, 6, 3, 12, 11, 9, 13.
	NB-5	2
	NB-6	1

Fig. 3 and 4 show the feature gain ratio [28] (used as a feature evaluator) of the original Slammer and Code Red I datasets, respectively, in comparison to all oversampling methods. The ranker search method employed within filter classifiers gives priority in ranking to volume features over AS-PATH features (Tab. 5 and 6), which is another confirmation that volume features could be used for anomaly prediction.

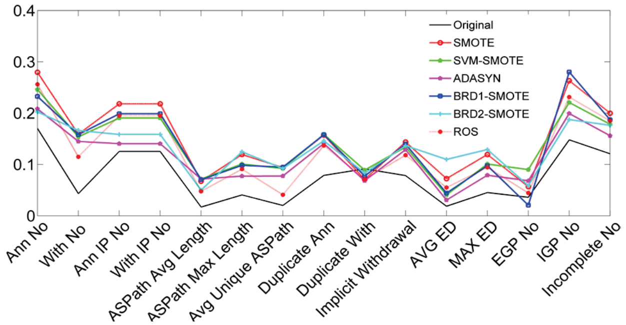

Figure 3 Gain ratio feature evaluator – Slammer dataset (Original) vs. oversampled datasets

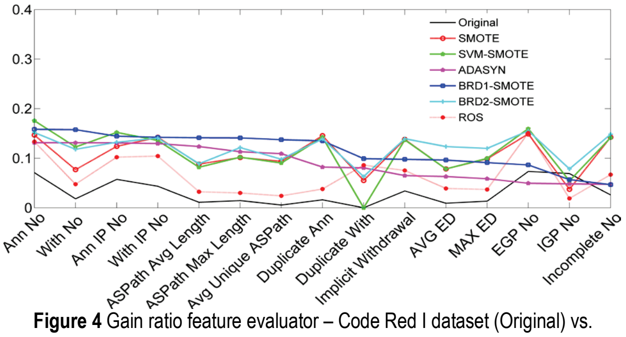

Figure 4 Gain ratio feature evaluator – Code Red I dataset (Original) vs. oversampled datasets

Models based on decision trees (J48) outperform other models (are best more often) in achieving the best classification performance measures on randomly oversampled (ROS) datasets (Tab. 5).

With one exception, models based on SVM and NB outperform decision tree models for undersampled datasets, regardless of whether they were created on original Slammer, Code Red I, Moscow Power Blackout or AS 9121 Routing Table Leakdatasets (Tab. 6).

4 CLASSIFICATION RESULTS

For each of the original datasets, namely, Slammer, Nimda, Code Red I, Moscow Power Blackout, and AS 9121 Routing Table Leak, we create 18 different datasets based on undersampling, oversampling and resampling combination techniques, for a total of 90 balanced datasets. Once we get the datasets, we use them as input to four Filter/Wrapper classifiers that each use SVM, J48 and NB as base classifiers, hence twelve different classifiers. We perform 360 simulations (= 90 datasets × 4 Filter/Wrapper classifiers) for each of the base classifiers and report the best results achieved for each of the undersampling, oversampling and resampling combination datasets. The original Nimda dataset is only slightly imbalanced (Tab. 2). Hence, performance measures attained from the Nimda resampled datasets only show marginal improvement in comparison to a previous analysis [1] and are not reported in this paper.

In contrast, performance measures of imbalanced Slammer, Code Red I, Moscow Power Blackout, and AS 9121 Routing Table Leak datasets that became balanced by resampling as reported here show improvements in all F-measures, MCC, ROC and PR compared to their values reported for imbalanced datasets in [1] and [2].

4.1 Slammer Dataset

Out of the six oversampling methods applied to the Slammer dataset, SMOTE achieves the best performance, and this is when SVM is used as the base classifier, regardless of the Filter and Wrapper classifier (Tab. 7). In all tables below, the best performing model (based on the F-measure) is highlighted in gray color for each of the three base classifiers, SVM, J48, and NB. For all tables, numbers 3 and 4 in "Model" columns refer to Filter models, whereas 5 and 6 refer to Wrapper models.

The time taken to build a model depends on the oversampling method, as exemplified by the SVM-5 classifier (Fig. 5). Time is normalized with respect to the longest time taken to build a model, which occurs using the ADASYN dataset in the case of the Filter and Wrapper classifiers that use SVM as a base classifier.

Table 7 Performance of the SVM base classifier combined with the four Filter/Wrapper classifiers (3, 4, 5, 6) on the SMOTE oversampled Slammer dataset; the row with the best F-measure is highlighted.

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9525	0.952	0.905	0.953	0.932	3.27
SVM-4	0.9621	0.962	0.924	0.962	0.946	3.74
SVM-5	0.9645	0.964	0.928	0.964	0.949	522.91
SVM-6	0.9652	0.965	0.929	0.965	0.950	2373.53

In the case of the SMOTE-generated balanced dataset, the time taken to build a model for all classifiers is longer than for the original dataset, which was expected because we use oversampling methods and increase the size of the feature matrix. The classifier testing option is a cross-validation with ten folds, so chances of overfitting are reduced.

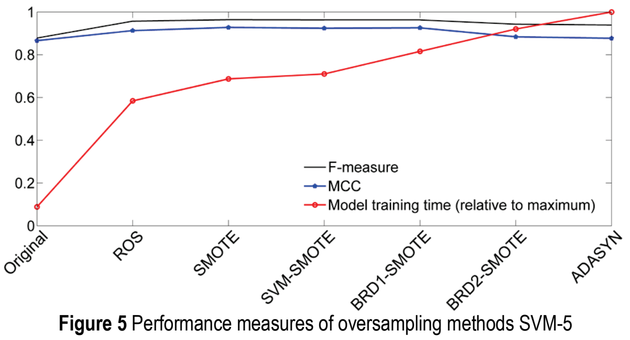

Figure 5 Performance measures of oversampling methods SVM-5

Table 8 Performance of the J48 classifiers on the ROS oversampled Slammer dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
J48-3	0.9776	0.978	0.956	0.983	0.959	0.28
J48-4	0.9847	0.985	0.969	0.986	0.966	0.42
J48-5	0.9829	0.983	0.966	0.984	0.959	10.44
J48-6	0.9827	0.983	0.966	0.987	0.967	158.25

The best performance measures for the balanced dataset that is produced by applying the ROS method to the Slammer dataset are achieved by the Filter and Wrapper classifiers that use a J48 as a base classifier

(Tab. 8). The time taken to build models using a decision tree as a base classifier varies considerably depending on the Filter and Wrapper methods (Tab. 8), as is also the case for the SVM base classifier (Tab. 7).

The best performance measures for the dataset generated by applying the BRD2-SMOTE method to the Slammer dataset tare achieved by the Filter and Wrapper classifiers that use a NB as a base classifier (Tab. 9).

Table 9 Performance of the NB classifiers on the BRD2-SMOTE oversampled Slammer dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-3	0.9764	0.976	0.953	0.997	0.998	0.28
NB-4	0.9785	0.979	0.957	0.998	0.998	0.32
NB-5	0.9796	0.979	0.959	0.998	0.998	3.71
NB-6	0.9798	0.980	0.960	0.998	0.998	33.97

Out of the ten undersampling methods applied to the Slammer dataset, RUS achieves the best performance with one of each of the SVM, J48, and NBbase classifiers performing almost equally well in terms of the F-measure. Hence, we present results in Tab. 10 for all three base classifiers. However, of those three, a J48 decision tree is the best performing base classifier in terms of time taken to build a model (Tab. 10). This time from using undersampled datasets is shorter than when using original datasets, as was expected given that the number of samples has decreased, which reduced model training time as well.

Results show that oversampling methods (Tabs. 7-9) achieve better performance measures than undersampling methods (Tab. 10), except for model training time; oversampling methods take longer. Also, although F-measures may be almost identical for two classifiers, training time may differ considerably. For example, the training time for wrapper model NB-5 is less than one ninth of the training time of NB-6 (Tab. 9), but their F-measures differ byonly a small amount.

This difference in training time among methods is especially notable in comparisons that include wrapper methods because they employ a subset evaluator to create all possible subsets of features. The learning algorithm is wrapped into the process of selection subsets; hence, it takes longer than filter methods to train the model.

Table 10 Performance of the SVM, J48, and NB classifiers on the RUS undersampled Slammer dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9378	0.938	0.876	0.938	0.913	0.08
SVM-4	0.9355	0.935	0.871	0.936	0.910	0.14
SVM-5	0.9413	0.940	0.833	0.941	0.922	12.92
SVM-6	0.9418	0.941	0.844	0.942	0.922	46.06
J48-3	0.9424	0.942	0.885	0.958	0.937	0.04
J48-4	0.9367	0.937	0.873	0.947	0.910	0.03
J48-5	0.9344	0.935	0.869	0.943	0.902	0.73
J48-6	0.9367	0.937	0.873	0.959	0.950	2.65
NB-3	0.9309	0.932	0.862	0.979	0.974	0.02
NB-4	0.9229	0.923	0.846	0.980	0.981	0.03
NB-5	0.9413	0.942	0.883	0.985	0.986	0.47
NB-6	0.9384	0.939	0.877	0.986	0.986	3.91

The procedure to create a balanced dataset using a resampling combination technique is to first oversample the original (Slammer or Code Red I) dataset using a SMOTE algorithm and then to undersample the resulting dataset using either the Tomek links or ENN method. This process generates the SMOTE+Tomek links and SMOTE+ENN Slammer datasets. Tab. 11 and Tab. 12 show the best performing classifiers out of all Filter and Wrapper classifiers evaluated on the SMOTE+Tomek links and SMOTE+ENN datasets, respectively. There are small differences in F-measures of the J48-4, NB-5 and NB-6 models, but the training time for the NB-5 classifier with the SMOTE+Tomek links dataset is less than one twelfth of the time of the NB-6 classifier, and J48-4 is one tenth of the time of NB-5 (Tab. 11).

Table 11 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+Tomek Links Slammer dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-5	0.9638	0.964	0.929	0.964	0.948	520.33
J48-4	0.9754	0.976	0.951	0.980	0.966	0.4
NB-5	0.9769	0.977	0.954	0.997	0.998	4.44
NB-6	0.9768	0.977	0.954	0.997	0.998	57.96

For the SMOTE+ENN dataset, the training time of the NB-5 classifier in less than one ninth of the training time of the NB-6 classifier, and J48-4 is less than one tenth of the time of NB-5 (Tab. 12).

Table 12 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+ENN Slammer dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-6	0.9662	0.966	0.933	0.966	0.951	2401.73
J48-4	0.9758	0.976	0.952	0.983	0.962	0.37
NB-5	0.9765	0.976	0.953	0.998	0.998	5.07
NB-6	0.9771	0.977	0.954	0.998	0.998	45.93

4.2 Code Red I Dataset

Out of the six oversampling methods applied to the Code Red I dataset, SVM-SMOTE achieves the best performance, and this occurs when SVM is used as the base classifier, regardless of the Filter and Wrapper classifier (Tab. 13).

Table 13 Performance of the SVM classifiers on the SVM-SMOTE oversampled Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9015	0.902	0.803	0.902	0.860	6.51
SVM-4	0.9187	0.918	0.838	0.919	0.886	6.66
SVM-5	0.9184	0.919	0.837	0.918	0.884	1231.39
SVM-6	0.9196	0.920	0.839	0.920	0.885	4973.44

Table 14 Performance of the J48 classifiers on the ROS oversampled Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
J48-3	0.9639	0.965	0.930	0.970	0.937	0.43
J48-4	0.9672	0.968	0.936	0.971	0.939	0.53
J48-5	0.9658	0.967	0.934	0.966	0.929	14.49
J48-6	0.9661	0.967	0.935	0.971	0.936	283.08

The Random oversampling (ROS) method applied to the Code Red I dataset achieves the best performance of the Filter and Wrapper classifiers that use a J48 as a base classifier. There is only a small difference in performance measures between J48-4 filter and wrapper models but a substantial difference in time needed to build the training model (Tab. 14).

The BRD2 SMOTE oversampling method applied to the Code Red I dataset achieves the best performance of the Filter and Wrapper classifiers that use a NB as a base classifier (Tab. 15).

Table 15 Performance of the NB classifiers on BRD2-SMOTE oversampled Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-3	0.9682	0.968	0.937	0.991	0.994	0.37
NB-4	0.9671	0.966	0.935	0.991	0.994	0.39
NB-5	0.9668	0.966	0.934	0.990	0.992	4.28
NB-6	0.9687	0.968	0.938	0.991	0.994	29.41

Out of the ten undersampling methods used on the Code Red I dataset, Near miss-1 achieves the best performance, and it occurs with SVM and NB base classifiers. SVM-3and NB-6 classifiers perform almost equally well in terms of the F-measure. However, of those two, a SVM is the best performing base classifier in terms of time taken to build a model (Tab. 16).

Table 16 Performance of the SVM, J48, and NB classifiers on the Near Miss-1 undersampled Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9191	0.914	0.845	0.919	0.910	0.11
SVM-4	0.9175	0.912	0.842	0.918	0.909	0.09
SVM-5	0.9091	0.903	0.825	0.909	0.896	8.8
SVM-6	0.9166	0.912	0.839	0.917	0.905	54.84
J48-3	0.8891	0.885	0.781	0.891	0.864	0.03
J48-4	0.8945	0.890	0.793	0.897	0.874	0.04
J48-5	0.8950	0.890	0.793	0.888	0.880	0.37
J48-6	0.8966	0.891	0.798	0.900	0.887	2.23
NB-3	0.9125	0.907	0.831	0.951	0.965	0.02
NB-4	0.9075	0.902	0.820	0.950	0.965	0.02
NB-5	0.9041	0.910	0.815	0.955	0.932	0.22
NB-6	0.9100	0.915	0.827	0.958	0.942	2.95

Tab. 17 and Tab. 18 show the best performing classifiers out of all Filter and Wrapper classifiers evaluated on SMOTE+Tomek links and SMOTE+ENN datasets that were created from the original Code Red I dataset. For both datasets, the best performing models are Wrapper models using NB as a base classifier. The NB-5 classifier is the first choice for both datasets because it has a shorter training time, although it has an equal or slightly smaller ROC and PR values compared to the NB-6 model. Also, performance measures for the combination of over- and undersampling techniques are in between those of separate oversampling and undersampling techniques.

Table 17 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+Tomek Links Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-5	0.8851	0.880	0.773	0.885	0.854	1472.55
J48-6	0.9537	0.953	0.908	0.975	0.975	130.96
NB-5	0.9667	0.966	0.934	0.989	0.992	3.78
NB-6	0.9664	0.966	0.934	0.991	0.993	34.29

Table 18 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+ENN Code Red I dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-5	0.8849	0.880	0.775	0.886	0.855	1401.25
J48-6	0.9492	0.949	0.901	0.971	0.971	156.87
NB-5	0.9668	0.967	0.937	0.990	0.993	3.93
NB-6	0.9662	0.966	0.935	0.991	0.993	39.54

4.3 Moscow Power Blackout Dataset

Out of the six oversampling methods applied to the MPB dataset SVM-SMOTE achieves the best performance, and this occurs when SVM is used as the base classifier, regardless of the Filter and Wrapper classifier (Tab. 19).

Table 19 Performance of the SVM classifiers on the SVM-SMOTE oversampled MPB dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9835	0.984	0.967	0.984	0.973	2.7
SVM-4	0.9905	0.990	0.981	0.990	0.985	2.2
SVM-5	0.9909	0.991	0.982	0.991	0.986	369.82
SVM-6	0.9909	0.991	0.982	0.991	0.986	2059.67

The Random oversampling (ROS) method applied to the MPB datasets achieves the best performance of the Filter and Wrapper classifiers that use a J48 as a base classifier. There is only a small difference in performance measures (MCC and PR) between J48-4 filter and wrapper models for MPB dataset but a substantial difference in time needed to build the training model (Tab. 20).

Table 20 Performance of the J48 classifiers on the ROS oversampled MPB dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
J48-3	0.9976	0.998	0.995	0.998	0.993	0.22
J48-4	0.9982	0.998	0.996	0.998	0.995	0.28
J48-5	0.9981	0.998	0.996	0.998	0.993	3.25
J48-6	0.9979	0.998	0.996	0.998	0.994	80.27

The BRD2 SMOTE oversampling method applied to the MPB dataset achieves the best performance of the Filter and Wrapper classifiers that use a NB as a base classifier (Tab. 21). NB-4 model has a slightly smaller F-measure than NB-6, yet its training time is just a small fraction of that of NB-6 model.

Table 21 Performance of the NB classifiers on BRD2-SMOTE oversampled MPB dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-3	0.9938	0.994	0.988	0.999	0.999	0.24
NB-4	0.9949	0.995	0.990	0.999	0.999	0.33
NB-5	0.9942	0.994	0.988	0.999	0.999	3.88
NB-6	0.9956	0.996	0.991	0.999	0.999	71.89

Table 22 Performance of the SVM, J48, and NB classifiers on the Near Miss-1 undersampled MPBdataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9728	0.973	0.948	0.973	0.973	0.01
SVM-4	0.9757	0.976	0.954	0.976	0.976	0.01
SVM-5	0.9661	0.966	0.937	0.967	0.967	0.5
SVM-6	0.9658	0.966	0.937	0.967	0.967	1.94
J48-3	0.9459	0.946	0.894	0.956	0.958	0.01
J48-4	0.9447	0.948	0.908	0.941	0.951	0.01
J48-5	0.9491	0.949	0.900	0.949	0.953	0.08
J48-6	0.9523	0.952	0.906	0.957	0.966	0.15
NB-3	0.9628	0.963	0.931	0.980	0.986	0.01
NB-4	0.9659	0.966	0.937	0.988	0.992	0.01
NB-5	0.9639	0.964	0.929	0.980	0.985	0.1
NB-6	0.9552	0.955	0.912	0.972	0.979	0.44

Out of the ten undersampling methods used on the MPB dataset, Near miss-1 achieves the best performance with SVM base classifiers. The SVM-4 classifier performs the best in terms of the F-measure and model training time (Tab. 22).

Tab. 23 and Tab. 24 show the best performing classifiers out of all Filter and Wrapper classifiers evaluated on SMOTE+Tomek links and SMOTE+ENN datasets that were created from the original MPB dataset. For both datasets, the best performing models are Wrapper models using NB as a base classifier. The NB-5 classifier is the first choice for SMOTE+Tomek links dataset because it has a shorter training time, although it has all other performance measures equal to the NB-6 model. In case of SMOTE+ENN dataset, the NB-5 model has the best performance of all models in Tab. 24.

Table 23 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+Tomek Links MPB dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-6	0.9968	0.997	0.993	0.999	0.999	36.27
NB-5	0.9969	0.997	0.993	0.999	0.999	4.2
NB-4	0.9961	0.996	0.991	0.999	0.998	0.27
J48-6	0.9936	0.994	0.988	0.996	0.994	115.19

Table 24 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+ENN MPB dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-5	0.9967	0.997	0.994	0.999	0.999	4.25
NB-6	0.9962	0.996	0.993	0.999	0.999	45.49
NB-4	0.9948	0.995	0.991	0.999	0.998	0.38
J48-5	0.9941	0.994	0.988	0.994	0.990	7.29

4.4 AS 9121 Routing Table Leak Dataset

Out of the six oversampling methods applied to the RTL dataset, SVM-SMOTE achieves the best performance when SVM is used as the base classifier, regardless of the Filter and Wrapper classifier (Tab. 25).

Table 25 Performance of the SVM base classifiers on the SVM-SMOTE oversampled RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9317	0.932	0.905	0.878	0.901	4.12
SVM-4	0.9409	0.941	0.904	0.921	0.903	5.38
SVM-5	0.9543	0.954	0.918	0.934	0.934	478.18
SVM-6	0.9546	0.955	0.919	0.939	0.943	2651.22

Table 26 Performance of the J48 classifiers on the BRD1-SMOTE oversampled RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
J48-3	0.9558	0.956	0.961	0.966	0.973	0.22
J48-4	0.9639	0.964	0.963	0.970	0.971	0.33
J48-5	0.9727	0.973	0.939	0.964	0.948	3.34
J48-6	0.9758	0.976	0.941	0.966	0.955	32.07

The best performance measures for the balanced dataset that is produced by applying the BRD1-SMOTE method to the RTL datasets is achieved by the Filter and Wrapper classifiers that use a J48 as a base classifier (Tab. 26).

The best performance measures for the dataset generated by applying the BRD2-SMOTE method to the RTL dataset is achieved by the Filter and Wrapper classifiers that use a NB as a base classifier (Tab. 27).

Out of the ten undersampling methods applied to the RTL dataset, RUS achieves the best performance with one of each of the SVM, J48, and NBbase classifiers performing in terms of the F-measure for all three base classifiers (Tab. 28). J48 classifiers have similar performance measures as well as identical model training time for filter classifiers.

Table 27 Performance of the NB classifiers on the BRD2-SMOTE oversampled RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-3	0.9356	0.936	0.937	0.937	0.917	0.27
NB-4	0.9309	0.931	0.986	0.993	0.952	0.28
NB-5	0.9799	0.980	0.978	0.981	0.982	2.33
NB-6	0.9810	0.981	0.979	0.983	0.983	26.59

Table 28 Performance of the SVM, J48, and NB classifiers on the RUS undersampled RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
SVM-3	0.9778	0.978	0.916	0.978	0.944	0.01
SVM-4	0.9718	0.972	0.911	0.976	0.957	0.05
SVM-5	0.9803	0.980	0.877	0.984	0.964	6.27
SVM-6	0.9809	0.981	0.891	0.969	0.967	26.16
J48-3	0.9779	0.980	0.959	0.979	0.965	0.01
J48-4	0.9806	0.981	0.962	0.981	0.969	0.01
J48-5	0.9878	0.988	0.977	0.989	0.981	0.04
J48-6	0.9872	0.987	0.975	0.987	0.981	0.08
NB-3	0.9831	0.983	0.971	0.980	0.981	0.01
NB-4	0.9888	0.989	0.979	0.991	0.990	0.02
NB-5	0.9918	0.992	0.986	0.996	0.996	0.08
NB-6	0.9937	0.994	0.987	0.999	0.999	0.23

Results show that undersampling methods on RTL dataset (Tab. 28) achieve better performance measures than oversampling methods (Tab. 25, 26, 27), including the model training time; undersampling methods take less time since dealing with less data points in the feature matrix. Also, although F-measures may be almost identical for two classifiers, training time may differ. For example, the training time for wrapper model J48-5 is one half of the training time of J48-6 (Tab. 28), but their F-measures differ by only a small amount.

Tab. 29 and Tab. 30 show the best performing classifiers out of all Filter and Wrapper classifiers evaluated on the SMOTE+Tomek links and SMOTE+ENN datasets, respectively. There are small differences in F-measures of the NB-5 and NB-6 models, but the training time for the NB-5 classifier with the SMOTE+Tomek links dataset is one tenth of the time of the NB-6 classifier (Tab. 29).

Table 29 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+Tomek Links RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
NB-4	0.9506	0.950	0.949	0.984	0.988	0.74
J48-6	0.9778	0.978	0.976	0.987	0.968	140.67
NB-5	0.9801	0.980	0.979	0.979	0.984	4.17
NB-6	0.9819	0.981	0.980	0.980	0.986	41.4

Table 30 Performance of the best models of SVM, J48, and NB classifiers on the SMOTE+ENN RTL dataset

Model	Acc	F-measure	MCC	ROC	PR	Time (s)
J48-6	0.9651	0.965	0.978	0.984	0.980	5.99
NB-4	0.9829	0.983	0.981	0.989	0.988	0.58
NB-6	0.9836	0.984	0.983	0.989	0.989	44.71
NB-5	0.9863	0.986	0.984	0.989	0.989	3.99

For the SMOTE+ENN dataset, the training time of the NB-5 classifier is less than one eleventh of the training time of the NB-6 classifier, and NB-4 is one seventh of the time of NB-5 (Tab. 30).

5 CONCLUSION

To improve performance measures of BGP detection models, we investigated the effects of oversampling, undersampling and the combination of over- and undersampling methods when applied to imbalanced datasets of known network anomalies. We used Slammer, Nimda, Code Red I, Moscow Power Blackout, and Routing Table Leak datasets that contain known anomalies to develop anomaly detection algorithms. For each of these five datasets, we applied 18 resampling methods to generate 18 different balanced datasets and evaluated their performance using 12 Filter and Wrapper classifiers with NB, SVM, and J48 as base classifiers.

Because the Nimda dataset is not imbalanced, we observed only minor improvement in performance measures when comparing performance of classifiers on the resampled Nimda and original Nimda datasets, as expected.

All other datasets benefited from application of resampling techniques; performance measures improved compared to the original imbalanced datasets. The datasets that were oversampled by using the ROS algorithm showed the best classifier performance measures compared to other oversampling methods when evaluated by the J48-4decision tree classifier.

Training time for the original Slammer dataset is one third of the training time of the Slammer dataset oversampled by the ROS algorithm (0.14 s vs. 0.42 s), whereas the training time of the original Code Red I dataset is slightly less (0.52 s vs. 0.53 s) than that of Code Red I dataset oversampled by the ROS algorithm. Also, training time for the original MPB dataset is less than a half of the training time of the MPB oversampled by the ROS algorithm (0.15 s vs. 0.28 s) whereas training time for the original RTL dataset is less than a fifth of the training time of the RTL oversampled by the ROS algorithm (0.05 s vs. 0.33 s). Decision tree algorithms have performed better in combination with ROS oversampling algorithm than with the original datasets on all datasets except RTL. For RTL the best performance measures result when oversampled by BRD2-SMOTE algorithm, although only by a small margin over the ROS algorithm. The Slammer and RTL datasets that were undersampled with the RUS algorithm showed the best classifier performance measures compared to other undersampling methods when evaluated by the J48-3and J48-5 decision tree classifiers, respectively. Training time for the Slammer dataset undersampled by using RUS algorithm is one third of the original Slammer dataset (0.04 s vs. 0.13 s), whereas the training time for the RTL dataset undersampled by the same algorithm is one seventeenth of the original RTL dataset (0.04 s vs. 0.69 s), as expected because of fewer data points being processed. The Code Red I and MPB datasets that were undersampled by using the Near Miss-1 algorithm, showed the best classifier performance measure compared to using other undersampling methods when evaluated by the Naïve Bayes classifier NB-6 and SVM-4 respectively. Training time for the original Code Red I dataset is slightly less (2.87 s vs. 2.95 s) than the training time for the Code Red I dataset undersampled by using the Near Miss-1 algorithm, whereas training time for the original MPB dataset is thirty-five times larger (0.35 s vs. 0.01 s) than the training time for the MPB dataset undersampled by using the Near Miss-1 algorithm.

Our analyses of oversampling, undersampling, and resampling combination techniques show that the combination methods perform comparatively well. Given the potential problems of possible overfitting caused by oversampling techniques, as well as loss of information that can occur when using undersampling techniques, we suggest using the resampling combination techniques SMOTE+Tomek Links and SMOTE+ENN. For those techniques with the Slammer dataset, the performance measures for the classifier method NB-5 were equal to or better than those of other classifiers, except for model training time, which was best for J48-4. For the Code Red I dataset, NB-5 performed best across all performance measures for both resampling combination techniques. When SMOTE+TomekLinks techniques were applied to the Moscow Power Blackout dataset, we found equal performance measures for NB-5 and NB-6 models apart from model training time, which was superior for the NB-5 model. For the SMOTE+ENN technique with the MPB dataset, the performance measures for the classifier method NB-5 were better than those of other classifiers. In the case of the RTL dataset when used with SMOTE+Tomek Links, the resampling combination technique NB-6 model had better performance measures (apart from model training time) by a small margin over NB-5 model. We recommend using a NB-5 classifier along with combination resampling techniques to produce the best trade-off between (1) the F-measure value and the length of training time, and (2) avoidance of overfitting (oversampling) and loss of information (undersampling).

6 REFERENCES

[1] Ćosović, M., Obradović, S., & Trajković, Lj. (2015). Performance evaluation of BGP anomaly classifiers. *Proceedings of the Int. Conference on Digital Information, Networking and Wireless Communication* / Moscow, 115-120.

[2] Ćosović, M., Obradović, S., & Trajković, Lj. (2016). Classifying anomalous events in BGP datasets. *Proceedings of the 29th Annual IEEE Canadian Conference on Electrical and Computer Engineering* / Vancouver, 697-700.

[3] Ding, Q., Li, Z., Batta, P., & Trajković, Lj. (2016). Detecting BGP anomalies using machine learning techniques. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* / Budapest, 3352-3355.

[4] Data Mining for Imbalanced Datasets: An Overview. // Data Mining and Knowledge Discovery Handbook / Nitesh V. Chawla. Springer US, 2005. 853-867.

[5] Lemaitre, G., Nogueira, F., & Aridas, C. K. (2016). Cornell University Library : Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning // arXiv. https://arxiv.org/abs/1609.06570. (31.01.2017)

[6] Frank, E., Hall, M. A., & Witten, I. H. (2016). *The WEKA Workbench*. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition.

[7] (Feb. 2, 2017) North American Network Operators Group Mailing List, https://www.nanog.org/mailinglist/mailarchives/old_archive/2003-01/msg00583.html

[8] (Feb. 2, 2017) North American Network Operators Group Mailing List, https://www.nanog.org/mailinglist/mailarchives/old_archive/2001-09/msg01355.html

[9] (Feb. 2, 2017) North American Network Operators Group Mailing List, https://www.nanog.org/mailinglist/mailarchives/old_archive/2001-07/msg00375.html

[10] (Feb. 2, 2017) North American Network Operators Group Mailing List, https://www.nanog.org/mailinglist/mailarchives/old_archive/2005-05/msg00650.html

[11] (2017, Apr.) Popescu, A. C., Premore, B. J., & T. Underwood. (May 19, 2005). Anatomy of a Leak: AS9121. Renesys Corporation. Manchester, NH, USA. http://research.dyn.com/content/uploads/2013/05/renesys-nanog34.pdf.

[12] (2017, Jan.) RIPE RIS raw data, http://www.ripe.net/data-tools/stats/ris/ris-raw-data.

[13] Manderson, T. Multi-threaded routing toolkit (MRT) Border Gateway Protocol (BGP) routing information export format with geo-location extensions. RFC 6397. IETF. http://www.ietf.org/rfc/rfc6397.txt. (31.01.2017)

[14] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research, 16*(1), 321-357.

[15] Akbani, R., Kwek, S., & Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets. *Proceedings of the 15th European Conference on Machines Learning* / Pisa, 39-50. https://doi.org/10.1007/978-3-540-30115-8_7

[16] Haibo, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. *IEEE International Joint Conference on Neural Networks, IEEE World Congress on Computational Intelligence* / Hong Kong, 1322-1328. https://doi.org/10.1109/IJCNN.2008.4633969

[17] (2005). Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. Advances in intelligent computing / Hui Han, Wen-Yuan Wang, Bing-Huan Mao. Springer Berlin Heidelberg, 878-887.

[18] Mani, I. & Zhang, I. (2003). kNN approach to unbalanced data distributions: a case study involving information extraction. *Proceedings of workshop on learning from imbalanced datasets*.

[19] Tomek, I. (1976). Two modifications of CNN. *IEEE Transactions on Systems, Man, and Cybernetics, 6*(11), 769-772.

[20] Rahman, M. M. & Davis, D. (2013). Cluster based under-sampling for unbalanced cardiovascular data. *Proceedings of the World Congress on Engineering* / London, 1480-1485.

[21] Kubat, M. & Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. *Proceedings of the 14th Int. Conference on Machine Learning* / Nashville, 179-186.

[22] Wilson, D. (1972). Asymptotic Properties of Nearest Neighbor Rules Using Edited Data. *IEEE Transactions on Systems, Man, and Cybernetrics, 2*(3), 408-421. https://doi.org/10.1109/TSMC.1972.4309137

[23] (2001). Improving identification of difficult small classes by balancing class distribution. Artificial Intelligence in Medicine / Jorma Laurikkala. Springer Berlin Heidelberg, 63-66.

[24] Hart, P. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory, 14*(3), 515-516. https://doi.org/10.1109/TIT.1968.1054155

[25] Batista, G. E. A. P. A., Bazzan, A. L. C., & Monard, M. C. (2003). Balancing Training Data for Automated Annotation of Keywords: a Case Study. *Proceedings of the Second Brazilian Workshop on Bioinformatics* / Rio de Janeiro, 35-43.

[26] Batista, G. E. A. P. A., Prati, R. C., & Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Exploration Newsletter, 6*(1), 20-29. https://doi.org/10.1145/1007730.1007735

[27] Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA) - Protein Structure, 405*(2), 442-451.

[28] Davis, J. & Goadrich, M. (2005). The relationship between Precision-Recall and ROC curves. *Proceedings of the 23rd Int. Conference on Machine Learning* / Pittsburgh, 233-240.

[29] Witten, I. H. & Frank, E. (2011). *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 314-322. https://doi.org/10.1016/0005-2795(75)90109-9

Contact information:

Marijana ĆOSOVIĆ, corresponding author
Faculty of Electrical Engineering, University of East Sarajevo
Vuka Karadzića 30, 71123 East Sarajevo, B&H
E-mail: marijana@etf.unssa.rs.ba

Slobodan OBRADOVIĆ
Faculty of Electrical Engineering, University of East Sarajevo
Vuka Karadzića 30, 71123 East Sarajevo, B&H
E-mail: slobo.obradovic@gmail.com