Deep Learning for Detection of BGP Anomalies: Selected Contributions from ITISE 2017

Chapter · October 2018

DOI: 10.1007/978-3-319-96944-2_7

Obradovic Slobodan
University of East Sarajevo
40 PUBLICATIONS **48** CITATIONS
SEE PROFILE

Emina Junuz
University Dzemal Bijedic Mostar
15 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Deep Learning for Detection of BGP Anomalies

Marijana Cosovic[1], Slobodan Obradovic[1], Emina Junuz[2]

[1]Faculty of Electrical Engineering, University of East Sarajevo, Istocno Sarajevo, BiH
[2]Faculty of Information Technology, Dzemal Bijedic University, Mostar, BiH

marijana.cosovic@etf.unssa.rs.ba, slobo.obradovic@gmail.com, emina@edu.fit.ba

Abstract. The Internet uses Border Gateway Protocol (BGP) for exchange of routes and reachability information between Autonomous Systems (AS). Hence, BGP is subject to anomalous traffic that can cause problems with connectivity and traffic loss. Routing Table Leak (RTL), worm and power outage events are considered anomalous in the sense that they can disrupt Internet routing and cause slowdowns of varying severity, which leads to packet delivery reliability issues. Deep learning, a subfield of machine learning, could be applied in detection of BGP anomalies. Studying RTL, worm and power outage events is of interest to network operators and researchers alike. In this paper we consider datasets of several events, all of which caused large-scale Internet outages. We use artificial neural network (ANN) models based on a backpropagation algorithm for anomalous event classification.

Keywords: Machine learning·Deep learning·Anomaly detection·BGP·Sampling

1 Introduction

The AS-level Internet topology is a structure in which autonomous systems (AS), collections of routers with same routing policies, are represented by nodes, while the connection between the nodes are data paths used for exchanging reachability information between the ASs. Each AS is uniquely represented by an autonomous system number (ASN). As the number of ASs increased over time in the Internet, BGP was defined to improve existing EGP (External Gateway Protocol) and its drawbacks in terms of hierarchical structure that limited efficient expansion of the Internet. The latest version of the BGP protocol is BGP-4 defined in [1]. BGP is a routing protocol that connects different domains and it is used for routing in networks composed of a large number of ASs. It is used for routing among autonomous systems, and its latest version allows for Classless Inter-Domain Routing (CIDR), path aggregation, incremental additions, better filtering capabilities, and determining of the routing policy. The Routing Information Service (RIS) project was initiated in 2001 by the Réseaux IP Européens Network Coordination Centre (RIPE NCC). RIPE NCC belongs to one of the five Regional Internet Registries (RIRs) that manage allocation and registration of IP addresses and ASNs. The scope of the RIS project is collecting and storing routing data from

Remote route collectors (RRC) positioned predominantly at Internet exchange points. RRCs are software routers, realized on the Linux platform, that collect routing information. RRCs are positioned in all five RIRs but a majority of them are located within the RIPE NCC domain. Presently, eighteen RRCs are active and using Quagga routing software for collecting raw data that is stored in MRT routing information export format [2] to two different type of files: all BGP packets created every five minutes and a complete BGP routing table that is created every eight hours [3]. PyBGPDump, a library written in Python is used to convert MRT into ASCII format. The quality of the data is inspected after the conversion, as missing and corrupt data may occur.

By studying BGP packets files and, in particular, by extracting BGP update messages from them, as they contain important reachability information, we can study connectivity disruption in the Internet during anomalous events. We investigated several types of anomalous events, namely, routing table events, worms, and power outage events. RTL events are in general initiated by router misconfigurations and, although not malicious in nature, can cause connectivity problems and traffic loss. Worm and power outage events can also contribute to connectivity and traffic loss issues. All of the events considered in this study were globally visible events. We extracted fifteen features from BGP update messages on a minute-level: features related to volume of BGP messages and AS PATH features related to AS PATH attribute. For the duration of the anomalous event, we label the class feature with one (anomaly present), while the time before and after the anomalous event we label the class feature with zero (anomaly not present). In this way we obtained a labeled feature matrix for each of the events. Routing data, extracted from the BGP update messages, could be considered as time series data since data points are indexed in time order.

Machine learning techniques have been employed in anomaly classification tasks [4-7]. Deep learning, part of machine learning, has been used extensively in voice and image recognition, language modelling, and information retrieval, amongst others, and has impacted the wide range of information processing tasks [8]. Detection of anomalies in time series data has employed deep learning techniques in the past. ANNs are systems that can be trained to recognize patterns in data and classify anomalous from regular data instances [9], [10]. Routing data could be used to analyze past anomalous events and aid in classification of future anomalous events.

The paper is organized as follows. In Section 2, we describe ANNs. Introduction of anomalous events, such as particular RTL events, worm events and power outage event are discussed in Section 3. In addition, extraction of BGP features from the datasets concludes Section 3. Classification methodology and used performance measures are discussed in Section 4. We conclude with Section 5.

2 ANN - Deep Learning

Artificial Neural Networks (ANN) are originally developed to mimic basic biological systems and to learn based on examples in the way humans do. In essence, neural networks learn gradually from the interdependence of data input properties. This interdependence can be linear or non-linear in nature. Application of ANN has been present in the anomaly detection field [7], [9], [10]. When used in supervised learning neural network needs labeled input data; hence it is known in advance which class the data belongs to. Based on a comparison between the output of the neural network and the target function, during the training process, ANN adjusts the weights as shown in Fig. 1.

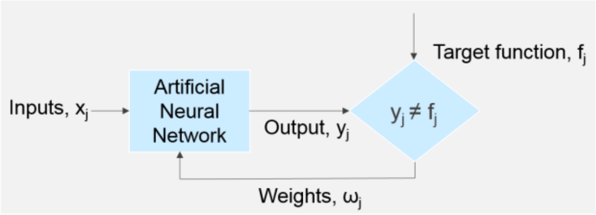

Fig. 1. Artificial Neural Network

Artificial neural networks can be classified as Feedforward or Feedbackward structures, depending on the direction of propagation of the information. The Feedbackward structure of neural networks refers to the spread of information backwards. When the input vector is applied to the input layer of the neural network, it propagates through the network throughout all its layers, and it generates output values by using the output layer of the network. The output values are compared with a desired target function, and for each of the neurons in the output layer the difference is calculated. Further information about these differences propagates backwards until all the neurons in the neural network are affected by the difference of the original and the target output value. The value of the weighting factors are determined by the optimization technique (typically a minimizing of the loss function with respect to the weights in the network), which determines the weighting factors such that the loss function is minimized.
ANN are simple mathematical methods made up of basic processing elements called neurons. The structures of neural networks differ in the number of layers used. Between the first and last layers of neural networks there are hidden layers: usually one hidden layer in simpler networks and more hidden layers in complex neural networks.

The architecture of the neural network is engaged in specific neuronal connectivity as a whole. Usually, the number of neurons in the input layer is equal to the number of features (number of columns in the feature matrix). Each neuron has one input, and all the outputs are connected to all neurons of the next layer, as shown in Fig. 2. When using a neural network for classification, the output layer can have one or more neurons, depending on whether it is binary or multi-class classification. The most commonly used functions for the output neuron modeling are sigmoid or normalized exponential [11] functions.

Perceptron is a neuron model type developed in the original neural networks, in which each neuron has a number of inputs (x_j) associated with corresponding weight factors (ω_j), which show the effect of a particular input on the output. Thus, the output neuron classifies information by comparing the value of the sum (1) and the threshold value, which is a parameter of the neuron.

$$\sum_{j} \omega_j x_j \tag{1}$$

Modeling of neurons with a perceptron has the following disadvantage: a small change in the weight factor of any perceptron can lead to a sudden change in its output. This in turn can lead to a complicated change in the rest of the network, which may be difficult to control. The most commonly used artificial neuron model, which solves the aforementioned problem, is the sigmoid neuron, shown by the following expression:

$$\frac{1}{1 + exp(-\sum_{j} \omega_j x_j - b)} \tag{2}$$

where ω_j are weighting factors, x_j are input neurons and b is bias. It turns out that a change in the output of sigmoid neurons linear function changes the weighting factors and bias. In this way it is easier to determine how changes in weighting factors and bias may influence the change of the output neuron; hence, the neural network could be considered more resilient to changes of data and the ability to learn.

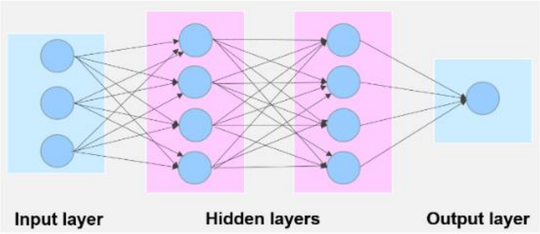

Fig. 2. Architecture of ANN with four layers: one input layer, two hidden layers and one output layer

3 Anomalous Events

The BGP routing system is subject to frequent incidents that result in significant interruptions of Internet connectivity. This can be observed in BGP update messages. In this paper we consider the following routing table leak events: Routing Leak AS9121 [12], AWS Route Leak [13], Telecom Malaysia AS4788 Route Leak [14], and Indosat Routing Table Leak [15], all of which showed an increased number of announced IP prefixes throughout the duration of the events. We also consider the Slammer [16] and Code Red I [17] worm events, as well as the Moscow power blackout event [18].

3.1 BGP Datasets

We obtain datasets from the RIPE NCC that collects Internet routing data by using Routing Information Service (RIS) Remote Route Collectors (RRC) positioned in various locations throughout the world. The effects of all the events considered in this paper and presented in Tab. 1 caused globally visible connectivity issues. We have used routing updates collected at two RRCs located in CIPX, Geneva and VIX, Vienna. We used BGP update messages during the occurrence of the routing leak, worm and power outage events stored in MRT format described in [2]. In order to create a feature matrix, we observed BGP update messages during a five day period, including two days before and two days after the actual event.

Table 1. BGP anomalous events

Dataset	Regular Class	Anomaly Class	Number of features
AS9121 RTL	7121	79	15
AWS RTL	7085	115	15
Malaysian Telecom RTL	7018	182	15
Indosat RTL	7050	150	15
Slammer	6331	869	15
Code Red I	6600	600	15
Moscow power outage	7031	169	15

Duration of the actual events lasted between 79 minutes, in the case of the AS9121 RTL event, and 869 minutes, in the case of the Slammer worm event. The rest of the anomalous events have durations that fall between those two values (Tab.1). Python code was written in order to extract features from the dataset that is a collection of BGP update messages after MRT to ASCII conversion for each of the events. For example, Tab. 2 shows that on June 12, 2015 at 16:05:02 UTC, AS12350 (192.65.185.157) announced that the address prefixes 177.155.50.0/23, 177.155.52.0/23, 201.46.160.0/19, and 201.46.232.0/21 were available. The path by which the above prefixes were available was ASPATH: 12350 174 6762 262589 262589 262589 262589 262589 28615. The original autonomous system was AS28615, while AS262589, AS6762 and AS174 were transit autonomous systems. BGP update messages traveled from the original, via transit, to the ultimate AS. On the other hand, the data transmitted were traversed by the sequence of ASs defined by the ASPATH attribute path (from left to right). Fifteen volume and AS-PATH features were extracted from BGP messages on a minute level during the five-day period, hence producing a feature matrix of 7200x15 in size.

The volume features that we observed are: the number of BGP messages an-nouncing new routes, the number of BGP messages withdrawing already existing routes, the number of announced IP prefixes (Fig. 3, 5, 6, 7, 8, 10 and 11), the number of withdrawn IP prefixes (Fig. 12), the number of duplicate announced messages, the number of duplicate withdrawn messages, the number of implicitly withdrawn messages, the number of BGP messages which NLRI originates from the Exterior Gateway Protocol (EGP), the number of BGP messages which NLRI originates from the Interior Gateway Protocol (IGP), and the number of BGP messages which NLRI originates from unknown sources.

Table 2. BGP update message during Telecom Malaysia AS4788 routing leak

FIELD:	VALUE:
TIME:	06/12/15 16:05:02
TYPE:	BGP4MP/MESSAGE/Update
FROM:	192.65.185.157 AS12350
TO:	192.65.185.40 AS12654
ORIGIN:	IGP
AS-PATH:	12350 174 6762 262589 262589 262589 262589 262589 28615
NEXT_HOP:	192.65.185.157
MULTI_EXIT_DISC:	0
ANNOUNCE	177.155.50.0/23 177.155.52.0/23 201.46.160.0/19 201.46.232.0/21

Duplicate announcements and withdrawal messages are defined as BGP update messages that announce the same combination of IP prefix and AS-PATH attribute that has previously been announced. Implicit withdrawal implies that the same IP prefix has been announced with a different AS-PATH attribute, hence it is an implicit withdrawal of a previous announcement (same IP prefix but different AS-PATH).

The features we computed based on the AS-PATH attribute are: the average length of the AS-PATH attribute, the maximum length of the AS-PATH attribute (Fig. 6), the average length of each unique AS-PATH attribute, the average edit distance, and the maximum edit distance (Fig. 4 and Fig. 9). While extracting information from the AS-PATH attribute, we considered regular and unique AS-PATHs. We also considered AS-PATHs as a string of ASNs (autonomous system number) and computed the similarity of two adjacent AS-PATHs by finding their edit distance [19].

Features belong to three types: continuous, categorical and binary. All of the volume features belong to the continuous type, since features may have an infinite number of values. On the other hand, features derived from the AS-PATH attribute may have a finite number of values and hence, are categorical. The class feature is of the binary type: given volume and AS-PATH features, we either have anomalous instances or not.

We have labeled all 7200 time instances (described by 15 features) as either belonging to anomalous or regular class in accordance with the information regarding the beginning, duration and end of each of the events. We have referred to several sources in order to label our data as correctly as possible.

Considering that global routing tables increased in size from the time of the first event, we needed to normalize feature values to account for Internet size growth. Normalization is done such that each feature vector has a zero mean and a standard deviation of one [20]. We also performed feature discretization for the features of the continuous type prior to training the neural network. We did not encounter any missing data during the seven events observed, although we did have an increased number of outliers in the case of the Indosat RTL and Code Red I datasets, which can be observed in Fig. 7 and Fig. 10 respectively.

3.2 Routing Table Leak Events

Many of the events that cause connectivity issues are classified as routing leaks. It is often unclear what is meant by that term. Based on research of actual events on the Internet, which can be of use to network operators and Internet users, the authors in [21] define routing leaks as a propagation of announced paths beyond the intended scope. This means that the BGP path announcement from one AS to another in some way violates the routing agreements between a sending AS, a receiving AS or any transit AS. The consequence of routing leaks is traffic redirection through a path not originally planned, and thus, various malicious attacks from analyzing data to eavesdropping could be performed. The most common reasons why routing leaks occur are errors in the router's configuration [22].

AS9121 Routing Table Leak.

The AS9121 Routing Table Leak took place on December 24, 2004. AS9121 announced to other ASs through BGP sessions that were used to reach almost 70% of all prefixes, which at that time amounted to more than 106k prefixes. As a result, the data of tens of thousands of networks were either lost or diverted. AS9121 started to announce prefixes to its neighbors around 9:20 GMT, and the event lasted until just after 10:00 GMT. AS9121 continued announcing prefixes for the rest of the day. The prefix announcement rate reached a second peak at 19:47 GMT. The number of announced IP prefixes during the routing leak event is shown in Fig. 3. An increase was observed in the number of withdrawn IP prefixes, as well. Besides the increase in the number of announced/withdrawn prefixes, the maximum edit distance (the measure of similarity between two ASPATH attributes) increased during the duration of the event, as can be observed in Fig. 4. This could indicate that the choice of the paths differed from the common ones, and it was sign of disruption between commonly connected ASs.

AWS Route Leak.

The AWS Route Leak started at 17:10 UTC on April 22, 2016 and affected a large number of ASs and prefixes. Loss of traffic and connectivity were present since networks with high traffic prefixes, such as Google, Amazon, and Twitter, were affected, amongst others.

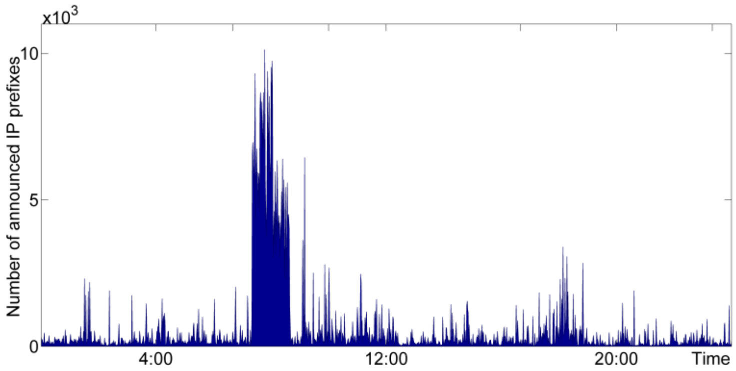

Fig. 3. Number of announced Network Layer Reachability Information (NLRI) prefixes during AS9121 Routing Leak Event as observed on RIPE Route Collector rrc04, CIPX

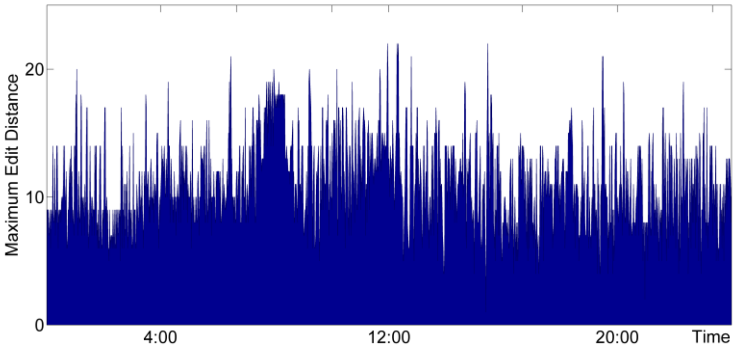

Fig. 4. Maximum edit distance during AS9121 Routing Leak Event as observed on RIPE Route Collector rrc04, CIPX

The event occurred due to maintenance issues on Innofield AG (AS 200759) that is connected to Swiss Internet eXchange (SwissIX). Innofield AG normally announces one IPv4 and IPv6 prefix to SwissIX. During maintenance reactivation of BGP sessions, AS 200759 distributed prefixes belonging to Amazon as belonging to private AS 65021. Prefix announcements were propagated through AS 6939 Hurricane Electric (HE) that peers at SwissIX. This resulted in a redirection of traffic passing through HE to a private AS, and hence, it compromised the reachability of Amazon AS. Since the event was widespread and likely caused by a misconfigured route optimizer, we observed an increase in announced IP prefixes at CIPX, as shown in Fig. 5.

Telecom Malaysia Route Leak.

The Malaysian Telecom (AS 4788) leaked one third of all IP prefixes in the global routing table to the backbone provider Level3 (AS 3549).

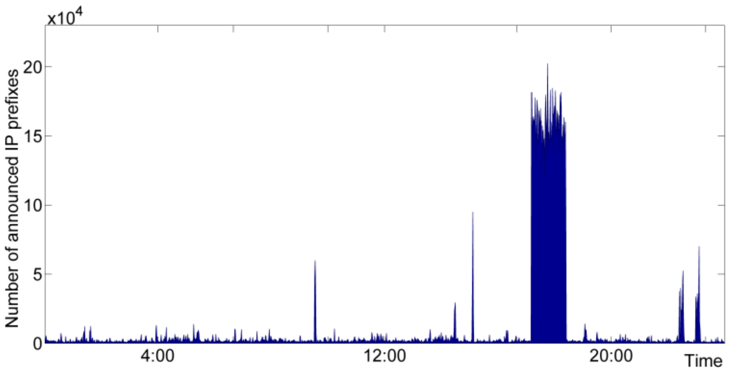

Fig. 5. Number of announced NLRI prefixes during AWS Routing Leak Event as observed on RIPE Route Collector rrc04, CIPX

The event, triggered by routers misconfiguration at Telecom Malaysia, started on June 12, 2015 at 8:43 UTC and lasted until 11:45 UTC. Level3 (AS 3549) propagated traffic from its peers and customers via Telecom Malaysia, which was not capable of handling the traffic volume, resulting in major packet loss and performance degradation. The performance degradation was especially pronounced between the Asia Pacific region and the rest of the Level3 network. Fig. 6 shows an increased number of announced IP prefixes (left) and also an increase of maximum AS-PATH length (right) for the duration of the route leak event.

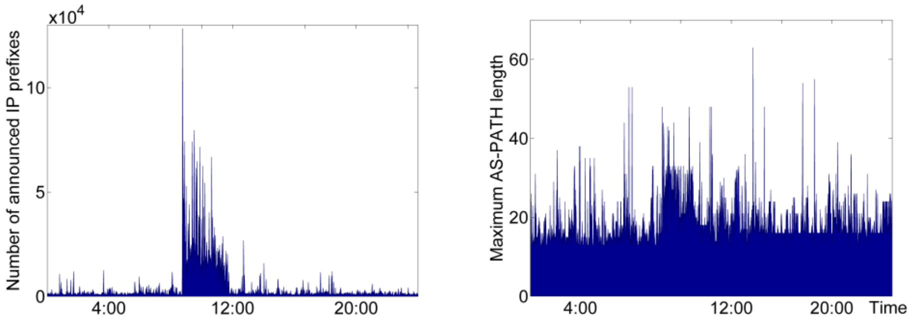

Fig. 6. Number of announced NLRI prefixes (left) and maximum AS-path length (right) during Telecom Malaysia Routing Leak Event as observed on RIPE Route Collector rrc04, CIPX

Indosat Routing Table Leak.

The Indosat routing table leak occurred on April 2, 2014. At the time of the event the global routing table consisted of nearly half a million routes. AS 4761(Indosat) leaked around 320,000 routes, which happened during scheduled maintenance, starting at 18:25 UTC. The reason behind Indosat originating prefixes that were not assigned to it is assumed to be that BGP was redistributed with bad upstream filtering. This inadvertent error had an impact that was observed on various route collectors through an increase of announced IP prefixes, as shown in Fig. 7. Several hundreds of those prefixes were widely accepted, and services of some networks such as Akamai, a leading content delivery network (CND) and cloud service provider, were disrupted.

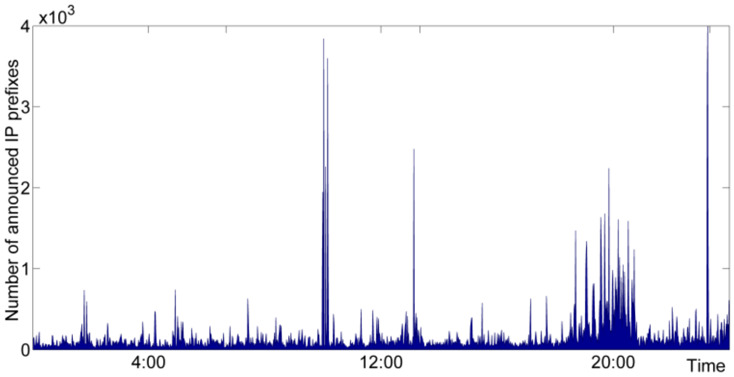

Fig. 7. Number of announced NLRI prefixes during Indosat Routing Leak Event as observed on RIPE Route Collector rrc04, CIPX

3.3 Worm Events

The Slammer worm is a single packet UDP scanning worm (404 bytes) that attacked MS SQL server and MS SQL server desktop addition on January 25, 2003. It spread worldwide in less than 10 minutes by sending copies of itself to random IP addresses. The main reason behind this rapid spread was the result of a bandwidth-limited scanner: each copy of the worm could scan at the maximum rate that the processor and network bandwidth could support. Depending on the upload bandwidth, every Slammer copy could be sending infectious packets at the maximum rate, hence the rapid spreading in which the number of infected machines doubled every 8.5 seconds. The number of announced IP prefixes during the Slammer worm is shown in Fig. 8, while the maximum edit distance increase can be observed in Fig. 9.

The Code Red I worm was released through IIS servers in June 2001, but the peak of infected computers was observed on July 19, 2001. The worm spread itself by creating a sequence of random IP addresses using static seed for generation of new IP addresses. Since every infected computer went through the same list of IP addresses, fewer systems were infected in comparison to the Slammer worm spread, and the number of infected machines doubled only every 40 minutes. The number of announced NLRI prefixes during the Core Red I worm event as observed on RIPE Route Collector rrc04, CIPX is shown in Fig. 10.

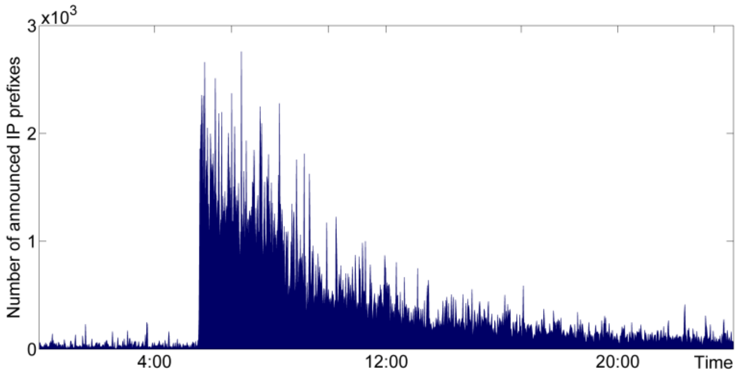

Fig. 8. Number of announced NLRI prefixes during Slammer worm event as observed on RIPE Route Collector rrc04, CIPX

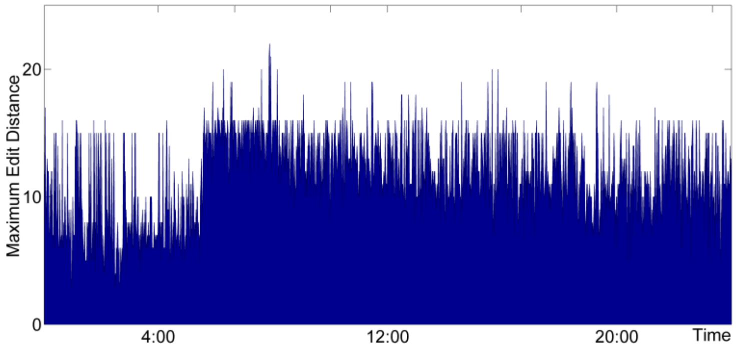

Fig. 9. Maximum edit distance during Slammer worm event as observed on RIPE Route Collector rrc04, CIPX

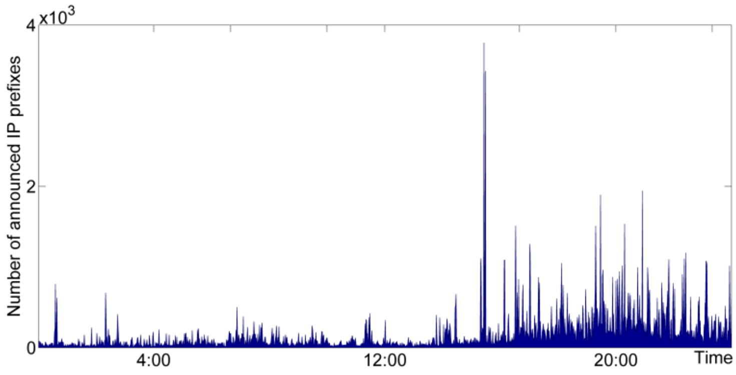

Fig. 10. Number of announced NLRI prefixes during Core Red I worm event as observed on RIPE Route Collector rrc04, CIPX

3.4 Power Outage Events

The Power outage event considered in this study is a Moscow power blackout that occurred on May 25, 2005. Moscow Internet eXchange (MSK-IX) was shut down at that time. Considering that 80% of Russian traffic at the time was passing through MSK-IX and that telecommunication infrastructure is greatly centralized around Moscow, rerouting of traffic created congestion. The outage had an impact on wide scale connectivity issues, hence, there was a disruption of Internet service: even though the data centers of main Russian websites had power, their traffic was still going through MSK-IX. Fig. 11 shows an increased number of announced IP prefixes in BGP update messages during the power outage, as observed at RIS remote route collector in Vienna Internet eXchange (VIX). The number of withdrawn IP prefixes during the Moscow power outage is shown in Fig. 12, and we can observe the time delay between the onset of power outage and the onset of the increase in number of withdrawn IP prefixes in BGP messages.

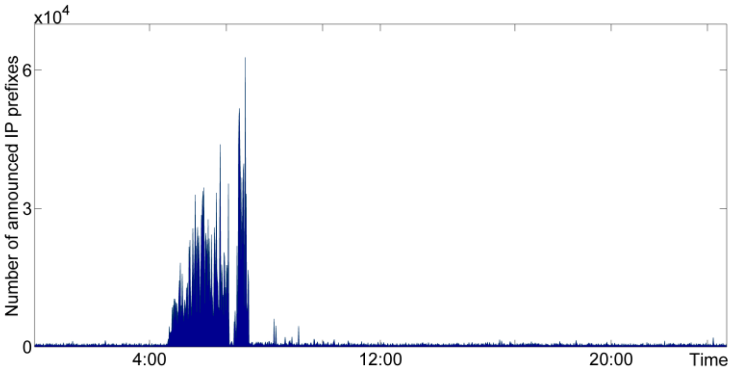

Fig. 11. Number of announced NLRI prefixes during Moscow power blackout as observed on RIPE Route Collector rrc05, VIX

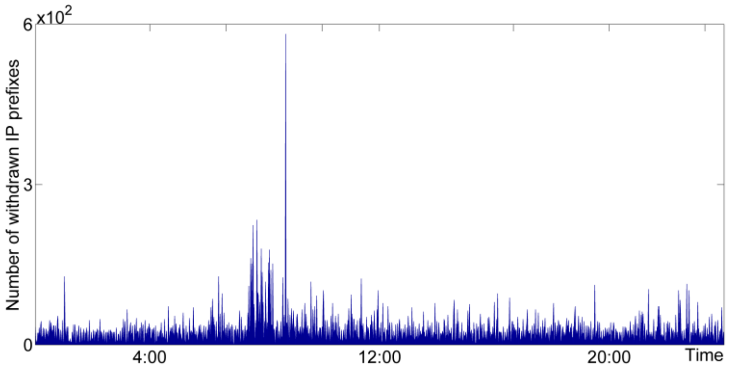

Fig. 12. Number of withdrawn NLRI prefixes during Moscow power blackout as observed on RIPE Route Collector rrc05, VIX

4 Classification of Anomalous Events

4.1 Methodology

We used the Keras Python library [23] with the Theano backend for development and evaluation of deep learning models. Models, based on a backpropagation algorithm for training of fully connected multilayer perceptron (MLP) neural networks, are defined as sequences of layers: an input layer, hidden layers and an output layer. The shape of the input data needs to be specified only for the first layer in the sequence. In Keras, using Dense class is one of the ways to define fully connected layers. Network weights can be initialized to random numbers using either uniform or Gaussian distribution. Use of appropriate activation function allows for better training of the network [24]. Traditionally, sigmoid and tanh activation functions are used, but the authors in [24] have shown that better performance can be achieved using a rectifier activation function. In the output layer we use a sigmoid function as we are dealing with binary classification.

We use 10-fold cross validation for determining accuracy on the test dataset, and as we increase the number of hidden layers beyond two, classification accuracy decreases. We found that a neural network with two hidden layers is the optimal model for the anomalous datasets considered: routing table leak, worm and power outage. Using either too few or too many neurons in the hidden layers may result in problems of underfitting and overfitting, respectively. General guidelines are used for determining the number of neurons within each hidden layer. We selected neural network architecture based on trial and error, but in accordance with the following general guidelines: the number of neurons in hidden layers should be between the sizes of input and output layers, and they should be the sum of 2/3 of the input layer neurons and output layer neurons. Hence, we trained the neural network with two dense hidden layers with 15 and 10 neurons, respectively.

4.2 Performance Measures

The goal of binary classification is to categorize data into two different classes: regular or anomalous. In most cases, the number of anomalous instances is a fraction of regular instances, and as such, the cost of classifying regular or anomalous instances is not the same. The performance measures employed in this paper, needed for comprehensive comparison of different deep learning models, are accuracy, F-measure, the Matthews Correlation Coefficient (MCC), the area under Precision-Recall (PR), the area under Receiver Operating Characteristics (ROC), and time taken to build a model. Accuracy, considering our datasets are highly imbalanced (Tab. 1), might not be the most accurate performance measure. This is due to the fact that misclassification would have different costs associated with points belonging to either the regular or anomalous class. Accuracy is defined as the ratio of points belonging to the regular/anomalous class that are classified as regular/anomalous and the total number of points in the dataset. In order to define F-measure, we first define recall (R) as the ratio of detected anomalous points and all points labeled as anomalous. On the other hand, precision (P) is a ratio of detected anomalous points and all anomalous points. Specificity (S) is a ratio of detected regular points and all regular points; hence it is a measure of how many regular instances are identified as regular. F-measure is given as a double ratio of the product of P and R and the sum of P and R. MCC is given by (3) where N is the number of all points and TP is the number of data points classified as anomalous.

$$MCC = \frac{TP / N - PR}{\sqrt{PR(1 - P)(1 - R)}} \tag{3}$$

The PR curve is more often used when there is a class imbalance problem [25], because both precision and recall measures are defined by focusing on the number of detected anomalous points. ROC curve represents the relationship between recall and specificity measure, and as such, remains the same regardless of the baseline prior probability of the anomalous class. This is reflected in the reported results (Tab. 3–Tab. 5) in which the area under the PR curve is often smaller than the area under the ROC curve.

4.3 Classification Results

We used a neural network with two hidden layers and obtained the performance measure values shown in Tab. 3. Accuracy is not the best approach to compare classification of different events, as the datasets are highly imbalanced. Hence, we used performance measures as introduced in section 4.2. In addition, time taken to build a model is added in the results. Tab. 1 shows that amongst all RTL events, Malaysian Telecom RTL has the largest set of data labeled as anomalous – 182 compared to the AS 9121 RTL event in which only 79 instances are labeled as anomalous. The Indosat RTL event shows the worst performance of all RTL datasets, and we can contribute that to noise in the dataset (Fig. 7). Slammer, followed by the Code Red I dataset, has the largest number of instances belonging to the anomaly class amongst all datasets. Noise in the Code Red I dataset, as shown in Fig. 10, might be the reason behind poor performance measures of Code Red I presented in Tab. 3. We used undersampling and oversampling techniques as in [26] to balance regular and anomalous instances in all datasets. In the case of oversampled and undersampled datasets, their imbalance ratio is around 1, meaning the classes are balanced; hence, accuracy and F-measure are approximately the same values.

Oversampling techniques are algorithms that create additional instances of the class that is represented by a smaller number of instances in the dataset. We used six oversampling techniques, namely, Synthetic Minority Oversampling Technique (SMOTE), Support Vector Machine (SVM)-SMOTE, Borderline1-SMOTE, Borderline2-SMOTE, Adaptive Synthetic Sampling (ADASYN), and Random Oversampling (ROS) algorithms. By using balancing techniques of the datasets, we achieved better performance measures, as shown in Tab. 4. The best results were achieved using the SVM-SMOTE oversampling technique for AS9121 RTL, AWS RTL, Indosat RTL, Slammer, Code Red I and Moscow dataset, while the Malaysian Telecom RTL dataset, when oversampled by ROS algorithm, had the best performance measure that was better by a small margin than when oversampled by the SVM-SMOTE algorithm.

Undersampling techniques are algorithms that remove instances from the dataset that belong to the more represented class. We used ten undersampling algorithms, namely, Near Miss-1, Near Miss-2, Near Miss-3, Tomek Links, Cluster Centroids, One-sided selection, Random undersampling (RUS), Edited Nearest Neighbours, Neighbourhood Cleaning Rule, and Condensed Nearest Neighbours. By using undersampling balancing techniques of the datasets, we achieved better performance measures, as shown in Tab. 5. When comparing Tab. 4 and Tab. 5, the values of performance measures (F-measure, MCC and ROC) are greater in the case of oversampling techniques for most datasets, and this is due to possible overfitting. Also, when datasets are oversampled, additional points from the anomalous class are added into the original dataset, hence, the area under the PR curve increases, as can be observed in Tab. 4.

Table 3. Performance measures of the original anomalous events

Dataset	Acc	F- measure	MCC	ROC	PR	Time(s)
AS9121 RTL	0.99375	0.945	0.942	0.998	0.946	10.55
AWS RTL	0.99431	0.808	0.807	0.961	0.848	10.95
Malaysian Telecom RTL	0.9925	0.852	0.848	0.979	0.883	10.58
Indosat RTL	0.93056	0.753	0.707	0.897	0.802	10.65
Slammer	0.95986	0.834	0.811	0.976	0.916	10.57
Code Red I	0.94542	0.586	0.582	0.887	0.628	10.65
Moscow power outage	0.99639	0.920	0.919	0.974	0.923	8.76

Table 4. Performance measures of anomalous events using oversampling techniques

Dataset	Acc	F-measure	MCC	ROC	PR	Time(s)
AS9121 RTL	0.99816	0.998	0.996	0.999	0.999	22.02
AWS RTL	0.99167	0.992	0.983	0.994	0.984	20.95
Malaysian Telecom RTL	0.98953	0.990	0.979	0.995	0.994	21.12
Indosat RTL	0.92087	0.923	0.844	0.958	0.940	21.19
Slammer	0.93854	0.940	0.878	0.977	0.967	19.63
Code Red I	0.89932	0.899	0.799	0.954	0.947	20.43
Moscow power outage	0.98962	0.990	0.979	0.997	0.997	18.1

Table 5. Performance measures of anomalous events using undersampling techniques

Dataset	Acc	F- measure	MCC	ROC	PR	Time(s)
92AS9121 RTL	0.98734	0.987	0.975	0.999	0.999	0.28
AWS RTL	0.96087	0.960	0.923	0.979	0.986	0.36
Malaysian Telecom RTL	0.95055	0.950	0.901	0.975	0.981	0.29
Indosat RTL	0.88333	0.878	0.770	0.927	0.948	0.31
Slammer	0.94131	0.941	0.883	0.983	0.979	2.73
Code Red I	0.9075	0.901	0.821	0.947	0.964	1.93
Moscow power outage	0.9645	0.964	0.929	0.986	0.989	0.48

The best results were achieved using the RUS undersampling technique for AS9121 RTL, AWS RTL, Slammer and Code Red I datasets, while Code Red I, Indosat RTL and Malaysian Telecom RTL datasets, when undersampled by the Near-Miss 1 algorithm, had the best performance measure, which was only better by a small margin than when undersampled by the RUS algorithm.

5 Conclusion

We have developed a model for anomaly detection based on artificial neural networks with two hidden layers, which are optimal because performance indices deteriorated with additional hidden layers. We used a cross-validation technique to determine the number of neurons in each of the layers. Balancing techniques (dataset oversampling and undersampling) were employed, as the original datasets are highly imbalanced. Classification of the Indosat RTL and Code Red I datasets achieved the worst performance measures, possibly due to noise in the datasets. Similar performance measures on those datasets propagated when undersampling and oversampling techniques were used. We concluded that employing volume and AS-PATH features extracted from BGP update messages could lead to reliable classification of anomalous events.

References

1. Rekhter, Y., Li T., Hares S.: A Border Gateway Protocol 4 (BGP-4). http://ietf.org/rfc/rfc4271 (2006) Accessed 20 June 2017
2. Manderson, T.: Multi-threaded routing toolkit (MRT) Border Gateway Protocol (BGP) routing information export format with geo-location extensions. rfc6397.txt (2011) Accessed 20 November 2017
3. RIPE RIS raw data. http://www.ripe.net/data-tools/stats/ris/ris-raw-data.
4. Ćosović, M., Obradović, S., Trajković, Lj.: Performance evaluation of BGP anomaly classifiers. In: Proceedings of the International Conference on Digital Information, Networking and Wireless Communication, pp. 115–120 (2015)
5. Cosovic, M., Obradovic, S., Trajkovic, Lj.: Classifying anomalous events in BGP datasets. In: Proceedings of the 29th Annual IEEE Canadian Conference on Electrical and Computer Engineering (CCECE 2016), pp. 697-700 (2016)
6. Cosovic, M., Obradovic, S.: Ensemble methods for classifying BGP anomalies. Industrial Technologies. 4(1), 12-20 (2017)
7. Ćosović, M., Obradović, S., Junuz, E.: Deep learning for detection of BGP anomalies. In: Proceedings of International work-conference on Time Series (ITISE 2017), pp. 487–498 (2017)
8. Deng, L., Yu, D.: Deep Learning: Methods and Applications. Foundations and Trends in Signal Processing. 7(3–4), 197-387 (2014)
9. Dau, H.A., Ciesielski, V., Song, A.: Anomaly Detection Using Replicator Neural Networks Trained on Examples of One Class. In: Proceedings of the 10th International Conference on Simulated Evolution and Learning, pp. 311-322 (2014)
10. Jadidi, Z., Muthukkumarasamy, V., Sithirasenan, E., Sheikhan, M.: Flow-Based Anomaly Detection Using Neural Network Optimized with GSA Algorithm. In: Proceedings of the 33rd

IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW '13), pp. 76-81 (2013)

11. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York Inc., Secaucus, NJ, USA (2006)
12. Popescu, A. C., Premore, B. J., Underwood, T.: Anatomy of a Leak: AS9121. https://www.nanog.org/meeting-archives/nanog34 /presentations/underwood.pdf (2005) Accessed 20 November 2017
13. AWS Route Leak-North American Network Operators Group Mailing List. https://mailman.nanog.org/pipermail/nanog/2016-April/085410.html (2016) Accessed 20 June 2016
14. Telecom Malaysia AS4788 Route Leak-North American Network Operators Group Mailing List. https://mailman.nanog.org/ pipermail/nanog/2015-June/076187.html (2015) Accessed 20 June 2016
15. Indosat Routing Table Leak-North American Network Operators Group Mailing List. https://mailman.nanog.org/pipermail/ nanog/2014-April/065920.html (2014) Accessed 20 June 2016
16. Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., Weaver, N.: Inside the Slammer Worm. IEEE Security and Privacy 1(4), 33-39 (2003)
17. Schauer, R.C.: The mechanisms and effects of the Code Red worm. https://www.sans.org/reading-room/whitepapers/dlp /mechanisms-effects-code-red-worm-87 (2001) Accessed 20 November 2017
18. Moscow Power Blackout-North American Network Operators Group Mailing List. https://www.nanog.org/mailinglist/ mailarchives/old_archive/2005-05/msg00650.html (2005) Accessed 20 June 2016
19. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions and reversals. Doklady Akademii Nauk SSSR 163(4), 845-848 (1965)
20. LeCun, Y., Bottou, L., Orr, G. B., Müller, K.-R.: Effiicient BackProp. In: Montavon, G., Orr, G. B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade. LNCS, vol. 7700, pp. 9-48. Springer-Verlag, London, UK (1998)
21. Sriram, K., Montgomery, D., McPherson, D., Osterweil, E., Dickson, B.: Problem Definition and Classification of BGP Route Leaks. https://www.rfc-editor.org/rfc/rfc7908.txt (2016)
22. Mahajan, R., Wetherall, D., Anderson, T.: Understanding BGP misconfiguration. In: Proceedings of the Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '02), pp. 3-16 (2002)
23. Chollet, F.: Keras. Available: https://github.com/ fchollet/keras (2016)
24. Nair, V., Hinton, G. E.: Rectified linear units improve restricted Boltzmann machines. In: Proceedings of 27th International conference on Machine Learning, pp. 807-814 (2010) Accessed 20 November 2017
25. Davis, J., Goadrich, M.: The relationship between Precision-Recall and ROC curves. In: Proceedings of 23rd International conference on Machine Learning, pp. 233–240 (2006) Accessed 20 November 2017
26. Ćosović, M., Obradović, S.: BGP anomaly detection with balanced datasets. Tehnički vjesnik/Technical Gazette, 25(3), (2018)