Comparing Machine Learning Algorithms for BGP Anomaly Detection using Graph Features

Odnan Ref Sanchez, Simone Ferlin, Cristel Pelsser, Randy Bush

Comparing Machine Learning Algorithms for BGP Anomaly Detection using Graph Features

Odnan Ref Sanchez
University of Strasbourg, France
orsanchez@unistra.fr

Simone Ferlin
Ericsson Research
simone.ferlin@ericsson.com

Cristel Pelsser
University of Strasbourg, France
pelsser@unistra.fr

Randy Bush
Internet Initiative Japan
randy@psg.com

ABSTRACT

The Border Gateway Protocol (BGP) coordinates the connectivity and reachability among Autonomous Systems, providing efficient operation of the global Internet. Historically, BGP anomalies have disrupted network connections on a global scale, i.e., detecting them is of great importance. Today, Machine Learning (ML) methods have improved BGP anomaly detection using volume and path features of BGP's update messages, which are often noisy and bursty. In this work, we identified different graph features to detect BGP anomalies, which are arguably more robust than traditional features. We evaluate such features through an extensive comparison of different ML algorithms, i.e., Naive Bayes classifier (NB), Decision Trees (DT), Random Forests (RF), Support Vector Machines (SVM), and Multi-Layer Perceptron (MLP), to specifically detect BGP path leaks. We show that SVM offers a good trade-off between precision and recall. Finally, we provide insights into the graph features' characteristics during the anomalous and non-anomalous interval and provide an interpretation of the ML classifier results.

CCS CONCEPTS

• **Security and privacy → Network security**; • **Computing methodologies → Machine learning**;

KEYWORDS

BGP, machine learning, anomaly detection, graph features

ACM Reference Format:
Odnan Ref Sanchez, Simone Ferlin, Cristel Pelsser, and Randy Bush. 2019. Comparing Machine Learning Algorithms for BGP Anomaly Detection using Graph Features. In *3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks (Big-DAMA '19), December 9, 2019, Orlando, FL, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3359992.3366640

Big-DAMA '19, December 9, 2019, Orlando, FL, USA

ACM ISBN 978-1-4503-6999-2/19/12…$15.00
https://doi.org/10.1145/3359992.3366640

1 INTRODUCTION

The Internet is composed of thousands of administrative domains known as Autonomous Systems (ASes), where the reachability of IP address space is exchanged using the Border Gateway Protocol (BGP). Given today's global BGP use and uncertainty of ASes' propagated information, any misconfiguration or malfunction of the protocol can compromise the Internet's stability.

Regardless of the intent behind these anomalies, whether they are malicious, such as worms or targeted attacks, misconfiguration, or link failures, there has been growing interest in detecting and mitigating BGP anomalies by observing BGP traffic [33], without depending on large-scale deployment solutions such as RPKI [9].

BGP anomaly detection has evolved from techniques such as time-series analysis to Machine Learning (ML) approaches as the latter deemed to improve detection and identify a wider range of BGP anomalies, e.g., misconfiguration, blackout, and worms [4]. In previous works, the primary features used are message volume and AS-PATH attributes extracted from BGP's update messages. When analyzing BGP anomalies over time, certain characteristics of the data may have changed, e.g., in terms of volume, which need to be considered when analyzing anomalies using historical data. Therefore we explore anomaly detection using graph features, which are more robust and appropriate for capturing the dynamics in the network topology. Graph features are primarily based on node centrality [30], clique (graph theory) [3], clustering coefficient [32], and hop count measures such as eccentricity [19].

In this paper, we provide a rigorous evaluation of the aforementioned graph features through an extensive comparison of different ML algorithms used in BGP anomaly detection, i.e., Naive Bayes classifier (NB) [5], Decision Trees (DT) [24], Random Forests (RF), Support Vector Machines (SVM) [13], and Neural Networks (NN) [10], that use graph features to detect BGP path leaks. Our results indicate that these algorithms are able to detect anomalies, which demonstrate that graph features do not depend on any ML method to show their strength as data input predictors. In our observations, MLP achieved the highest accuracy. Given that SVM is only outperformed by 0.3% on average, and it is more robust in discriminating anomalous and non-anomalous instants, we conclude that our best classifier is achieved using SVM.

The paper is structured as follows: Section 2 briefly discusses the related work and the graph features used in this study. Section 3 and 4 present the methodology and the assestment results of different ML algorithms. Finally, the concluding remarks are presented in Section 5.

2 BACKGROUND AND RELATED WORK

We briefly discuss here the most relevant related work on BGP anomaly detection, the current ML-based detection strategies, and the features used in our analysis.

2.1 Related Work

BGP anomaly detection looks for inconsistencies in the origin of prefixes announced by ASes or unexpected path changes. These are classified by the type of data used for detection: (i) control-plane, (ii) data-plane, and (iii) hybrid approaches [4]. Control-plane methods are usually third-party services such as BGPmon [1] or BGPStream [2], which have been effective in detecting large-scale events. On the other hand, ARTEMIS [33], a self-operated detection system, exploits local configuration and real-time BGP data from public monitoring services such as the RIPE Routing Information Service (RIS)[1]. It also provides protection against different types of attacks, including timely response against monkey-in-the-middle traffic manipulation. All previously mentioned methods are reactive and notify routing anomalies after they occurred. Data-plane approaches use network tools such as `ping` and/or `traceroute` to detect anomalies in the forwarding of packets. These approaches rely on monitoring the reachability of prefixes of the victim to detect anomalies [4]. Hybrid approaches have been investigated to address the limitations of exclusively control or data-plane methods. The main idea is to use control-plane inconsistencies to inform data-plane measurements, i.e., by exploring the reachability of targets in a particular network [4].

Further, graph features are well studied in BGP literature. For instance, node centrality has identified key ASes in a countrywide study in [38]. It is also used in [18] to identify abnormal routing changes from BGP data. Similarly, monitoring geometric curvature of AS-level topology is proposed in [31]. Large variations of curvature could potentially be used to detect BGP events. Though graph features are already explored for different applications, to the best of our knowledge, this paper first explores them as inputs to ML to detect anomalies in BGP.

2.2 ML-based BGP Anomaly Detection

Early large-scale BGP anomalies are mostly due to worm attacks, hence being the focus of literature. A complete history of BGP anomaly detection schemes can be found in [4, 34].

Numerous studies [5, 6, 8, 10–12, 14–17, 21, 23, 24, 29] have adopted ML to increase the accuracy of BGP anomaly detection. Here, we discuss briefly the recent works for SVM and NN.

SVM have been proven to work well with worm detection in BGP (e.g., [8, 14, 16]). More recently, Dai et al. [13] proposed SVM-based BGP Anomaly Detection (SVM-BGPAD) using different SVM kernels and Fisher algorithm for feature selection. They achieved a maximum of 91.36% accuracy in detecting worms using RBF kernel.

Recent contributions use deep learning for anomaly detection. Cheng et al. [10] propose Multi-scale LSTM that utilizes Discrete Wavelet Transform to include the temporal information. They evaluated their ML algorithms on the worm attacks plus a single path leak (i.e., TTNet table leak). On the other hand, Cosovic et al. [11] uses a simple Multi-layer Perceptron (MLP) but generalizes by taking into account different types of anomalies such as worms, table leaks, and blackout. However, rather than accuracy, the effects of under- and oversampling are the focus of this study.

The current ML-based works focus on worm attacks (i.e., [5, 6, 10, 12, 16, 17]), while only a few (cf. [11, 14]) have studied the combination of worm attacks, blackouts, and table leaks. Thus, there is a literature gap for detection schemes with recent attacks such as route leaks, which our study aims to fill. Moreover, previous works use traditional BGP volume and AS-PATH features, which may not work given that these features are found to be noisy and bursty [10]. In this study, more robust features such as topological features are proposed and tested on current attacks.

2.3 AS-Level Graph Features

Here, we present the features that we consider in this work. They are derived from the AS topology, which include node centrality [30], clique theory [3], clustering coefficient [32], and eccentricity [19]. Most of these features are used in other areas such as network robustness [30], while here we explore them for anomaly detection.

Centrality metrics reveal information about the most important elements in a graph. They have been widely used to speed information propagation in the network, damp epidemic virus propagation, and study network stability [30]. In general, **betweenness**, **load**, **closeness**, and **harmonic** centrality are measures of the path length, i.e., path-length centrality, whereas **degree**, **eigenvector** and **PageRank** measure richness of the neighbor graph, i.e., neighborliness.

Clique [26] is a complete subgraph of an undirected graph, which are often used for modelling clusters user groups, i.e., users that tend to call each other more often. We use the **number of cliques** and the size (**nodes in a clique**) as features to detect BGP anomalies.

Clustering coefficient [32] is the tendency of nodes in a graph to group together. It is frequently used for analyzing graphs and was introduced for studying social networks.

Triangles [32] are composed of three nodes and three edges, formed in a network. They are formally known as 3-cycles, where a cycle is defined as a closed trail.

Square clustering coefficient [22] is similar to triangles, but uses squares, i.e. cycles composed of four connections.

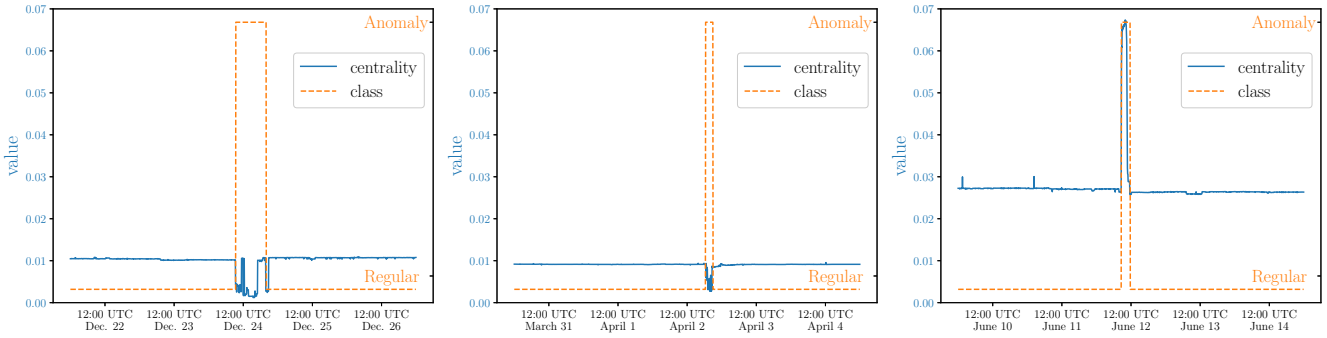
Average neighbor degree [7] measures the effective affinity to connect to high or low degree neighbors according to the actual interaction. Average neighbor degree, when combined with the clustering coefficient, better capture the effective level of cohesiveness between nodes [7].

Eccentricity [19] measures the maximum distance from a node to all other nodes in the graph.

3 METHODOLOGY

The main goal of this work is to assess the current status of an AS, whether it belongs to the "anomalous" or "normal" category, i.e., binary classification, for a given time interval using graph features. In this section, we describe the dataset used to train our classifiers. We also explain our methodology, the acquisition, and derivation of the graph features, the feature selection methods, the ML algorithms considered, and, finally, our evaluation metrics.

[1] RIPE RIS is a well-known repository for BGP datasets open for the research community: https://www.ripe.net.



(a) AS9121 (TTNet Table Leak) (b) AS4761 (Indosat Table Leak) (c) AS4788 (TM Table Leak) Set

Figure 1: An example showing eigenvector centrality graph feature during the anomaly events

Table 1: Datasets

Anomalies	Anomaly Start Date	Duration (min)	RRC
TTNet (AS 9121)	Dec. 24, 2004 (9:20 UTC)	627	RRC05
IndoSat (AS 4761)	April 2, 2014 (18:25 UTC)	150	RRC04
TM (AS 4788)	June 12, 2015 (8:43 UTC)	182	RRC04
AWS (AS 200759)	April 22, 2016 (17:10 UTC)	115	RRC04

3.1 The Datasets and the Experiment Setup

We select four well-known BGP events presented in Table 1: Turkish Telecom (TTnet) [17, 28], Indosat (Indonesia) [4, 11], Telecom Malaysia (TM) [11, 35], and the attack on Amazon Web Services (AWS) [11, 13, 36]. In contrast to TTNet, IndoSat, and TM, AWS was a narrowly targeted malicious incident. We chose them as they are recent BGP events that severely affected BGP and are also used in previous contributions [10, 11]. Furthermore, they had a large impact on the Internet and in the sheer number of prefixes.

Our BGP data was taken from the RIPE RIS Collectors RRC04 (Geneva) and RRC05 (Vienna). These collectors are widely used for research in BGP anomaly detection [10, 17]. We extracted the RIB table and control messages for the days before, during, and after the anomalies (5 days each in total). Collecting data over this time allowed us to distinguish between normal and anomalous behavior. We extracted 1440 samples for each event where each sample is composed of 14 features extracted from the Internet topology that is created every five minutes. The 5-minute interval is based on RIPE RIS' frequency of releasing the updates of the control messages. We obtained a total of 5760 samples from the four events, where only 218 out of 5760 are samples during the anomalous instant, i.e., our dataset is imbalanced. Thus, the classifier determines whether the current status of an AS shows anomalous behavior or not (i.e., binary classification), based on the graph features extracted from a snapshot of the Internet graph every 5-minute duration.

3.1.1 Feature Extraction. To extract the features, we recreated the AS-level topology from the AS-PATH field of the BGP announcements. Then, we extracted the graph features using `networkx`[2] for the whole 5-day duration, which served as input to our ML models. An example using the proposed eigenvector centrality feature is shown in Figure 1. It shows the feature only for the first three events from Section 3.1 due to space constraints, together with their labels during the regular and anomalous periods. Such periods are based on the anomaly start date and duration from Table 1.

The figures depict the feature's sudden change of behavior from normal to the anomalous event. This behavior is also found in most of the features. Interestingly, Figures 1a and 1b from TTNet and Indosat happened 10 years apart, but have similar values.

It is important to note that we are showing in the figure the graph features of the large-scale events which heavily affected the network. In these events, the patterns are clear as in the figures. However, not all attacks have this effect on the network. For instance, AWS does not have the same clear patterns which are not shown. Our goal is to show that graph features are good features for training our ML models and rely on ML solutions to detect more sophisticated patterns in the test sets/live deployment. ML discovers patterns that are difficult to find which are useful for complex data like BGP control messages.

3.1.2 Feature Selection. We based our feature selection algorithms from [17], which surveyed the most common feature selection algorithms used in BGP detection schemes. The methods included the minimum Redundancy Maximum Relevance (mRMR) family of algorithms (MR, MID, and MIQ) [27], and Fisher score [37]. Additionally, we also used univariate methods to select features. Univariate methods rank features by computing individual scores irrespective of the whole feature set. Such features include the Analysis of Variance (ANOVA), Mutual Information (MI), χ^2 test, and F-value scoring functions [39]. All these methods are used to select which subset of features are optimum.

3.1.3 GridSearchCV. The selected features were then fed into the different ML methods considered. For each ML method, there were different parameters that needed tuning. Finding the right set of parameters was needed in the training phase. Thus, we utilized sklearn's GridSearchCV[3] function to search for the optimum parameters. GridSearchCV performs the evaluation using cross-fold

[2]Networkx details can be found at https://networkx.github.io

[3]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

validation. It divides the training set into N subsets and uses N-1 subsets to train the model while using the remaining set for validation. In this study, we used 3-fold validation, i.e., GridSearchCV's default value. For each feature combination selected from Section 3.1.2, GridSearchCV explores different parameters, and those that yielded the optimal solution are chosen.

3.2 Machine Learning Algorithms

Previous work already showed that SVM [13], NB [5], DT [24], and NN [10] provided satisfactory results in detecting anomalies in BGP. In addition, these techniques are the most common binary classification methods [25], which are appropriate in classifying between anomalous and non-anomalous events. Also, we use such simple methods, since our number of features and samples are not extensive, i.e., 14 x 1440 instances for each attack, avoiding complex methods such as Deep Learning. Further, we also include RF, which is a natural extension of DT that corrects its tendency of overfitting. Thus, we utilize these methods, comparing them to find which best represents our features. Finally, we used the traditional feed-forward MLP for the type of NN and used only 6–16 number of neurons in the hidden layer.

Table 2: Confusion Matrix

		Predicted Class	
		Anomaly	Regular
Actual Class	Anomaly	True Positives (TP)	False Negatives (FN)
	Regular	False Positives (FP)	True Negatives (TN)

3.3 Evaluation Metrics

We use standard metrics for binary classification, which include Overall Accuracy (OA), Precision (PR), Recall (RC), and F-measure (F1), such as in [10, 13]. Additionally, since the data is highly imbalanced between anomalous and non-anomalous periods, i.e., small number of samples belonging to the anomaly class, we also propose a skewed measure of accuracy known as "Balanced Accuracy" (BA). Among accuracy metrics, BA is more reliable since it penalizes accuracy if the anomaly is not detected, as more weight is assigned to the minority class. The accuracy measures are taken from the True Positives (TP), False Positives (FP), True Negatives, (TN), and False Negatives (FN) derived from the confusion matrix in Table 2. PR measures the ability to detect without necessarily introducing false alarms, while RC measures the ability of the classifier to detect all anomalies, i.e., also known as TP rate. Finally, F1 is the balanced scale between PR and RC. For this reason, F1 and BA are metrics that appropriately represent unbalanced classes. The metrics are calculated as follows:

$$Overall\ Accuracy\ (OA) = \frac{TP + TN}{TP + FP + TN + TN} \tag{1}$$

$$Balanced\ Accuracy\ (BA) = \frac{\frac{TP}{TP+FP} + \frac{TN}{TN+FN}}{2} \tag{2}$$

$$Precision\ (PR) = \frac{TP}{TP + FP};\ Recall\ (RC) = \frac{TP}{TP\ +\ FN} \tag{3}$$

$$F - Measure\ (F1) = 2\ \frac{PR * RC}{PR\ +\ RC} \tag{4}$$

Given the concern regarding data imbalance, F1 was chosen to be the best metric in the training phase.

In Figure 3, we also compute the Receiver Operating Characteristic (ROC) and the Area Under the Curve (AUC) to measure the model performance. ROC curves are based on the graph of TP against FP rate for different classification thresholds. The AUC provides the measure of separability, which tells how much the classifier can distinguish between different classes. Having larger AUC means that the system distinguishes better between anomalous and non-anomalous instants.

Table 3: Our Classifier Set-up

SVM Models	**Training**				**Test**
	TTNet	IndoSat	TM	AWS	
Model A	x	✓	✓	✓	TTNet
Model B	✓	x	✓	✓	IndoSat
Model C	✓	✓	x	✓	TM
Model D	✓	✓	✓	x	AWS

4 PERFORMANCE EVALUATION

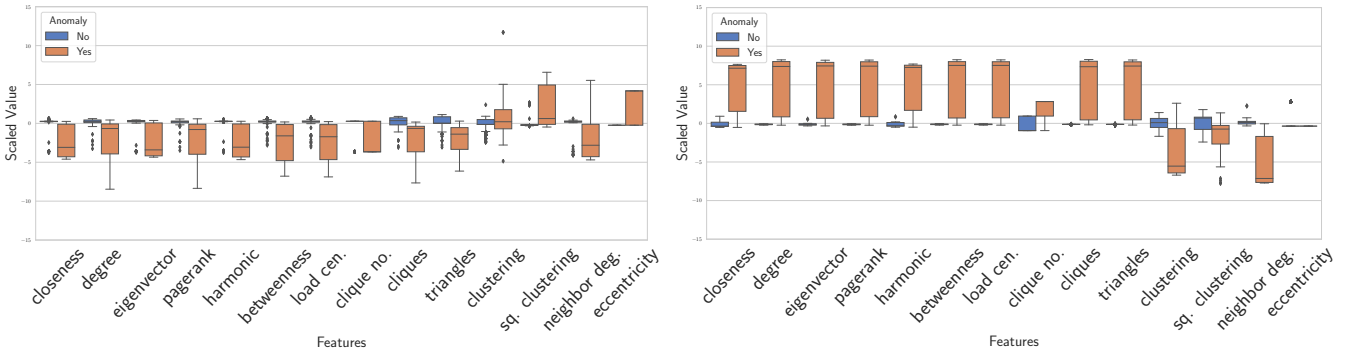
In this section, we discuss the results and comparison of the classifiers. Our data is split into training and test datasets for different incident combinations, as shown in Table 3. This allows the Models A—D to infer their performance on the test datasets that do not influence the training dataset itself. These combinations evaluate the strength of predicting unknown anomalies from training on existing anomalies. This implementation captures the real-world scenario where we currently have known dataset and see if it will detect unknown future events.

4.1 Data Analysis

Figure 2 shows the graph features of TTNet and TM events. We do not show Indosat and AWS due to space constraints. The graph emphasizes a sudden drop of centrality values during the TTNet (same case for IndoSat), in opposite to a sudden increase in TM (same case for AWS).

Graph features other than the centrality metrics also exhibit sudden change but in opposite behavior. For the eccentricity and average neighbor degree, as the centrality decreases, the number of links also decreases, potentially increasing the shortest path distances. For the clustering coefficients, the decrease in centrality means that the links directed to the AS decrease, and the links among its neighbors potentially increase. Therefore, we observe an increase in the clustering coefficients. Thus, for the TTNet and IndoSat incidents, the centrality metrics decrease, while eccentricity, average neighbor degree, and clustering coefficients increase. On the other hand, due to the characteristics of the incidents, TM and AWS show the opposite behavior.

The z-score normalization [20] was used to transform the data into a zero-mean distribution with unit variance for each feature. Then, to analyze the differences, we split the datasets into anomalous and non-anomalous instants for each feature, as shown in Figure 2. During the non-anomalous instant, the values are very close to the mean (≈ 0), which indicates the robustness of the measure during a non-anomalous instant. For the anomalous instant,



(a) TTNet (b) TM

Figure 2: Graph feature characteristics for TTNet and TM events split into anomalous and non-anomalous instants

the values tend to vary more widely, clearly showing the difference between both instants.

Table 4: Classification with SVM

Model	Training Metrics					Classification Results (Test)				
	FSA	NoF	C	γ	F1	OA	BA	F1	PR	RC
A	MID	3	2^{-9}	2^{-3}	69.18	97.57	90.41	85.48	89.56	81.74
B	MI	7	2^{-6}	2^{-6}	72.12	99.79	96.74	95.08	96.66	93.54
C	χ^2	4	2^{-7}	2^{-4}	68.53	99.38	87.84	86.15	100	75.68
D	χ^2	2	2^{-4}	2^{-4}	73.54	99.23	79.13	71.79	93.33	58.33

4.2 Experimental Results

Our results show that all the algorithms yield high detection rates. MLP and SVM have achieved the best results. Here, we focus on showing only SVM classifier results, as reported in Table 4. SVM performed best with RBF kernels, which are also found in [13].

For SVM, TTNet (Model A) can be detected with 90.41% BA using degree centrality, eccentricity, and ave. neighbor degree, which are selected by MID Feature Selection Algorithm (FSA). For IndoSat (Model B), SVM reaches 96.74% BA (99.8% OA) using seven features (i.e., five centrality metrics with ave. neighbor degree and square clustering coefficient). Detecting TM (Model C) also yields 87.8% BA (99.4% OA) with χ^2 as the FSA. The features include closeness, eigenvector, harmonic centralities, clique number, and square clustering. Most feature combinations detected the TM incident.

Similar to DT and RF, SVM yields the highest accuracy with IndoSat (Model B) and also achieves the lowest accuracy with AWS. All detectors achieved the lowest detection with AWS. Knowing that AWS is a particular case, we can conclude for the other three incidents that graph features are independent of ML algorithms.

Regarding features, AWS is detected using clique number with 75% BA (99.2 % OA) and clique size with 79.2% BA (99.3% OA), while Indosat, TM and TTNet are more likely to be detected by centrality metrics. These results show that centrality metrics are more likely to detect large-scale incidents while metrics that measure the grouping factor (e.g., clustering coefficient, cliques, triangles) are more likely to detect small-scale incident. Herewith, a plausible explanation is due to the robustness of the centrality features. Since it mainly measures the links to itself, small changes in these links do not provide significant changes. On the other hand, measures on grouping tendency take into account the connection between neighboring nodes, which arguably are more visible for small-scale incidents. During a small-scale incident, the small number of connections towards the node being measured will likely tend to connect to their neighboring nodes instead (i.e., forming cycles). Some links of the affected node disappear, while other new connection appears in the neighborhood of the affected AS. This, in turn, increases the measures of the features based on grouping (and remain undetected by centrality metrics). While it remains to be further studied, centrality metrics have the tendency to be more robust to noise and more reliable measures for large-scale incidents.

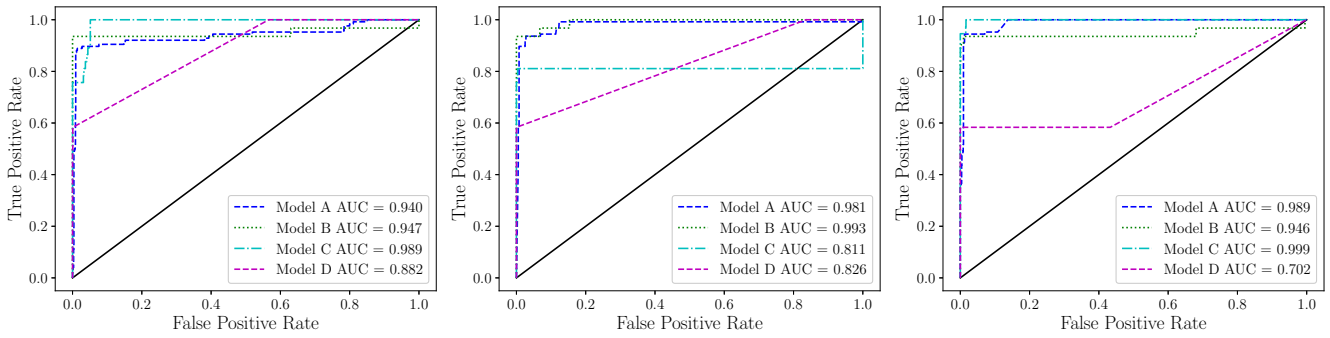
Regarding accuracy, AWS is detected the least among the four events. This result can be traced to its training set. In this model (model D), it is trained on three large-scale events, which means that the anomaly pattern of small-scale events is not taken into account, resulting in lower accuracy of prediction. This also supports the result that model B is the best model configuration as it is trained from both large and small-scale attacks. To improve our detection, we plan to extend our dataset and include more types of attacks.

We also evaluated the features by using them as individual inputs to the ML methods. The results provide information regarding which feature dominates on accuracy. The overall top five BAs for single feature prediction include, on average: node clique number (80.7%), number of cliques (78.5%), triangles (77.4%), eigenvector centrality (72.7%), and closeness centrality (71.9%).

4.3 Performance Comparison

We compare the performance of the ML algorithms and determine which one provides the best results using the accuracy and AUC metrics. The accuracy provides the "correctness" of the classifier in prediction while AUC provides the measure of class separability. AUC measures the classifier's "confidence" in its decision.

Accuracy. The MLP detector outperforms on average all other ML method's accuracy, 88.83% BA (99.01% OA), followed by SVM with a very small margin, 0.3% BA. Note that MLP already provides the best accuracy, even though we considered only a single hidden layer in our evaluations. Moreover, the MLP detector with a single hidden layer is enough to distinguish between anomalous and non-anomalous periods, indicating that graph features have distinct properties that are simple enough to be detected by conventional ML algorithms.



(a) SVM **(b) MLP** **(c) Naive Bayes**

Figure 3: ROC Curves and the AUCs of the ML models (showing only the top 3 ML algorithms)

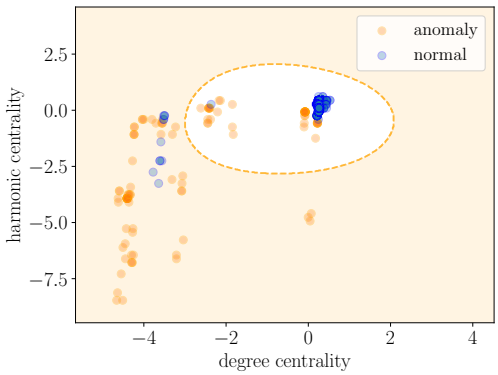


Figure 4: A 2-feature SVM decision boundary

Following SVM, DT and NB are the next best detectors, with RF performing worst on average with 82.56% BA (98.6% OA). However, RF underperforms DT, where it is interesting to note the changes as the dataset expands significantly. In theory, this may unveil the advantage of RF over DT.

AUC. Figure 3 shows the ROC curves of each model for the top three algorithms with their respective AUC values. On average, the SVM detector reached the highest AUC among all models, namely, 0.94 as shown in Figure 3a. This means that SVM models have higher confidence in its prediction, with 94% probability to distinguish correctly between anomalous and non-anomalous instants. SVM is followed by NB, MLP, DT and RF detectors, respectively. NB also yields high AUC for all anomalous instants as shown in Figure 3c, reaching on average 0.91. For MLP, although performing best on Models A and B, it only reached 0.81 with Model C, which is inferior to SVM, NB and DT altogether.

4.4 Result Interpretation

Since SVM is concluded as our best classifier, we show how our SVM models can be interpreted. Our 2-feature suboptimal SVM solution of model C is shown in Figure 4. It shows a non-linear decision boundary built from the training phase. The points in the figure are the samples from the test set (i.e., TTnet event). The classifier predicts "non-anomalous" when both degree and harmonic centrality values are near to the mean, which validates our intuition. These values near the mean are values near to zero in the figure as produced by the z-score normalization. Although, harmonic centrality is stricter since samples with smaller variations from the mean fall in the anomalous region. Decision boundaries help us interpret how the model predicts; however, it becomes more difficult when the dimensions in the feature space increase.

5 CONCLUSION

We compared the ability of different ML algorithms in detecting BGP path leaks from graph features. We have shown that graph features can be used to detect anomalies. MLP achieved the highest accuracy ("correctness"), which reached an average of 88.9% BA (99.01% OA), while SVM achieved the highest AUC Curve ("robustness"), which reached 94% on average. Given that SVM was only outperformed by 0.3% on average accuracy and it has been far more robust discriminating anomalous and non-anomalous periods, we conclude that it is our best classifier.

Interestingly, our results also provide preliminary views regarding large and small-scale attacks: centrality metrics are more likely to detect large-scale events, while metrics that measure the grouping factor (e.g., clustering coefficient, triangles, cliques) are more likely to detect small-scale events.

Most ML methods, such as in [13], still use the worm trio (Nimda-Code Red-Slammer) from 18 years ago. Thus, the set of attacks we considered was a step forward. Although the patterns in this dataset are fairly easy to detect, thanks to the proposed robust graph features, the use of ML prepares our detector for anomalies exhibiting more sophisticated patterns. The investigated ML techniques also prepare our detectors for future work in anomaly source detection.

We will continue to extend the number of events and evaluate our system by running it in a live deployment. Although, in the deployment, we expect that some of the features that are not computable in real-time will not be included. For instance, betweenness and load centrality are computationally expensive since they both need to compute the shortest path of all node pairs.

ACKNOWLEDGEMENTS

This project has been made possible in part by a grant from the Cisco University Research Program Fund, an advised fund of Silicon Valley Community Foundation.

REFERENCES

[1] 2019. BGPmon. https://bgpmon.net. (2019).

[2] 2019. CAIDA BGP Stream. https://bgpstream.caida.org. (2019).

[3] Eralp Abdurrahim Akkoyunlu. 1973. The enumeration of maximal cliques of large graphs. *SIAM J. Comput.* 2, 1 (1973), 1–6. https://doi.org/10.1137/0202001 arXiv:https://doi.org/10.1137/0202001

[4] Bahaa Al-Musawi, Philip Branch, and Grenville Armitage. 2017. BGP anomaly detection techniques: A survey. *IEEE Communications Surveys Tutorials* 19, 1 (2017), 377–396. https://doi.org/10.1109/COMST.2016.2622240

[5] Nabil Al-Rousan, Soroush Haeri, and Ljiljana Trajković. 2012. Feature selection for classification of BGP anomalies using bayesian models. In *2012 International Conference on Machine Learning and Cybernetics*, Vol. 1. IEEE, 140–147. https://doi.org/10.1109/ICMLC.2012.6358901

[6] Nabil M Al-Rousan and Ljiljana Trajković. 2012. Machine learning models for classification of BGP anomalies. In *2012 IEEE 13th International Conference on High Performance Switching and Routing*. IEEE, 103–108. https://doi.org/10.1109/HPSR.2012.6260835

[7] Alain Barrat, Marc Barthélemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. 2004. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences* 101, 11 (2004), 3747–3752. https://doi.org/10.1073/pnas.0400087101 arXiv:https://www.pnas.org/content/101/11/3747.full.pdf

[8] Prerna Batta, Maninder Singh, Zhida Li, Qingye Ding, and Ljiljana Trajkovic. 2018. Evaluation of Support Vector Machine Kernels for detecting network anomalies. In *IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–4. https://doi.org/10.1109/ISCAS.2018.8351647

[9] Randy Bush and Rob Austein. 2017. *The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1.* RFC 8210. RFC Editor.

[10] Min Cheng, Qing Li, Jianming Lv, Wenyin Liu, and Jianping Wang. 2018. Multi-Scale LSTM Model for BGP anomaly classification. *IEEE Transactions on Services Computing* (Apr 2018), 1–14. https://doi.org/10.1109/TSC.2018.2824809

[11] Marijana Cosovic, Slobodan Obradovic, and Emina Junuz. 2017. Deep learning for detection of BGP anomalies. In *Time Series Analysis and Forecasting*. Springer International Publishing, 95–113.

[12] Marijana Ćosović, Slobodan Obradović, and Ljiljana Trajković. 2015. Performance evaluation of BGP anomaly classifiers. In *2015 Third International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*. IEEE, 115–120. https://doi.org/10.1109/DINWC.2015.7054228

[13] Xianbo Dai, Na Wang, and Wenjuan Wang. 2019. Application of machine learning in BGP anomaly detection. *Journal of Physics: Conference Series* 1176, 3 (mar 2019), 1–12. https://doi.org/10.1088/1742-6596/1176/3/032015

[14] Iñigo Ortiz de Urbina Cazenave, Erkan Köşlük, and Murat Can Ganiz. 2011. An anomaly detection framework for BGP. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*. IEEE, 107–111. https://doi.org/10.1109/INISTA.2011.5946083

[15] Shivani Deshpande, Marina Thottan, and Biplab Sikdar. 2004. Early detection of BGP instabilities resulting from Internet worm attacks. In *IEEE Global Telecommunications Conference, GLOBECOM '04*, Vol. 4. IEEE, 2266–2270 Vol.4. https://doi.org/10.1109/GLOCOM.2004.1378412

[16] Qingye Ding, Zhida Li, Prerna Batta, and Ljiljana Trajković. 2016. Detecting BGP anomalies using machine learning techniques. In *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 3352–3355. https://doi.org/10.1109/SMC.2016.7844751

[17] Qingye Ding, Zhida Li, Soroush Haeri, and Ljiljana Trajković. 2018. Application of Machine Learning Techniques to Detecting Anomalies in Communication Networks: Datasets and Feature Selection Algorithms. (2018), 47–70. https://doi.org/10.1007/978-3-319-73951-9_3

[18] Romain Fontugne, Anant Shah, and Emile Aben. 2017. AS Hegemony: A Robust Metric for AS Centrality. In *Proceedings of the SIGCOMM Posters and Demos (SIGCOMM Posters and Demos '17)*. ACM, New York, NY, USA, 48–50. https://doi.org/10.1145/3123878.3131982

[19] Javier Martın Hernández and Piet Van Mieghem. 2011. *Classification of graph metrics.* Technical Report. Delft, Netherlands. 1–20 pages.

[20] Anil Jain, Karthik Nandakumar, and Arun Ross. 2005. Score normalization in multimodal biometric systems. *Pattern Recognition* 38, 12 (2005), 2270–2285. https://doi.org/10.1016/j.patcog.2005.01.012

[21] Jun Li, Dejing Dou, Zhen Wu, Shiwoong Kim, and Vikash Agarwal. 2005. An Internet routing forensics framework for discovering rules of abnormal BGP events. *SIGCOMM Comput. Commun. Rev.* 35, 5 (Oct. 2005), 55–66. https://doi.org/10.1145/1096536.1096542

[22] Pedro G Lind, Marta C Gonzalez, and Hans J Herrmann. 2005. Cycles and clustering in bipartite networks. *Physical Review E* 72, 5 (Nov 2005). https://doi.org/10.1103/physreve.72.056127

[23] Andra Lutu, Marcelo Bagnulo, Jesus Cid-Sueiro, and Olaf Maennel. 2014. Separating wheat from chaff: Winnowing unintended prefixes using machine learning. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 943–951. https://doi.org/10.1109/INFOCOM.2014.6848023

[24] Andra Lutu, Marcelo Bagnulo, Cristel Pelsser, Olaf Maennel, and Jesus Cid-Sueiro. 2016. The BGP visibility toolkit: Detecting anomalous internet routing behavior. *IEEE/ACM Transactions on Networking* 24, 2 (April 2016), 1237–1250. https://doi.org/10.1109/TNET.2015.2413838

[25] Neelam Naik and Seema Purohit. 2017. Comparative study of binary classification methods to analyze a massive dataset on virtual machine. *Procedia computer science* 112 (Sep 2017), 1863–1870. https://doi.org/10.1016/j.procs.2017.08.232

[26] James Orlin. 1977. Contentment in graph theory: covering graphs with cliques. *Indagationes Mathematicae (Proceedings)* 80, 5, 406–424. https://doi.org/10.1016/1385-7258(77)90055-5

[27] Hanchuan Peng, Fuhui Long, and Chris Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, 8 (Aug 2005), 1226–1238. https://doi.org/10.1109/TPAMI.2005.159

[28] Alin C Popescu, Brian J Premore, and Todd Underwood. 2005. *Anatomy of a leak: AS9121.* Technical Report.

[29] Andrian Putina, Steven Barth, Albert Bifet, Drew Pletcher, Cristina Precup, Patrice Nivaggioli, and Dario Rossi. 2018. Unsupervised real-time detection of BGP anomalies leveraging high-rate and fine-grained telemetry data. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 1–2. https://doi.org/10.1109/INFCOMW.2018.8406838

[30] Diego F Rueda, Eusebi Calle, and Jose L Marzo. 2017. Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements. *Journal of Network and Systems Management* 25, 2 (1 Apr 2017), 269–289. https://doi.org/10.1007/s10922-016-9391-y

[31] Loqman Salamatian, Dali Kaafar, and Kavé Salamatian. 2018. A Geometric Approach for Real-time Monitoring of Dynamic Large Scale Graphs: AS-level graphs illustrated. *CoRR* abs/1806.00676 (2018). arXiv:1806.00676 http://arxiv.org/abs/1806.00676

[32] Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. 2007. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E* 75, 2 (Feb 2007). https://doi.org/10.1103/physreve.75.027105

[33] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas A. Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti. 2018. ARTEMIS: neutralizing BGP hijacking within a minute. *CoRR* abs/1801.01085 (2018). arXiv:1801.01085 http://arxiv.org/abs/1801.01085

[34] Kotikapaludi Sriram, Oliver Borchert, Okhee Kim, Patrick Gleichmann, and Doug Montgomery. 2009. A comparative analysis of BGP anomaly detection and robustness algorithms. In *2009 Cybersecurity Applications & Technology Conference for Homeland Security*. IEEE, 25–38. https://doi.org/10.1109/CATCH.2009.20

[35] Andree Toonk. 2015. *Massive route leak causes internet slowdown.* Technical Report.

[36] Andree Toonk. 2016. *Large hijack affects reachability of high traffic destinations.* Technical Report.

[37] Koji Tsuda, Motoaki Kawanabe, and Klaus-Robert Müller. 2003. Clustering with the Fisher Score. In *Advances in Neural Information Processing Systems 15*. MIT Press, 745–752.

[38] Matthias Wählisch, Thomas C. Schmidt, Markus de Brün, and Thomas Häberlen. 2012. Exposing a nation-centric view on the German internet–A change in perspective on AS-level. In *Lecture Notes in Computer Science (International Conference on Passive and Active Measurement)*, Vol. 7192. Springer, Berlin, Heidelberg, Berlin, Heidelberg, 200–210. https://doi.org/10.1007/978-3-642-28537-0_20

[39] Zheng Zhao, Fred Morstatter, Shashvata Sharma, Aneeth Anand, and Huan Liu. 2010. Advancing Feature Selection Research. (2010), 28 pages.