Real Roots Isolation of Polynomials

XINLONG YI(862188160),
Computer Science and Engineering Department, University of California Riverside, USA

1 ABSTRACT

Hello s

2 PROBLEM

Given a model **M**,
Train M on a BGP anomaly dataset **D**
Detect BGP anomalies in other dataset(s)

As with any machine learning literature, the current literature for BGP anomaly detection does test for generalization, but only between training and testing sets *drawn from the same anomaly*. While these sorts of results are indicative of good performance for the tested anomalies, they cannot tell us much about the ability of models to generalize across multiple anomalies.

In this project, we present a novel analysis of BGP anomaly detection methods *between* different anomalies. We compare the generalization performance of several anomaly detection approaches, some of which have never been applied to this domain before. We measure the generalizability of these models using accuracy and F1 scores on test sets consisting of multiple anomalies. Furthermore, we explore the features responsible for generalization performance and attempt to interpret their importance.

2.1 Scope

An "anomaly" in networking can be difficult to define, as it is difficult to precisely characterize the appearance of "normal" traffic. A 2017 survey of BGP anomaly detection approaches by Al-Musawi et al. [1] constructs a taxonomy based on the cause of the anomalous behavior. E.g. "Direct" anomalies are caused by problems with the network itself, such as prefix hijacks ("direct intended") or origin misconfigurations("direct unintended"), while indirect anomalies occur as a result of events such as a worm spreading across the Web.

Finding data for many different anomalies was difficult itself, but compounding that problem was the fact that datasets generally share very few of the same features, which makes comparisons between them difficult. As a result, we limit our dataset to three indirect anomalies caused by computer worms in the early 2000s: Code Red I, Slammer, and Nimda. It would be ideal to include different types of anomalies from different time periods, but we cannot both gather the necessary data and perform the analysis with the time we have. The relative similarity of these anomalies can be beneficial though, as any deficiencies in generalization will represent a sort of upper bound on generalization performance.

Author's address: Xinlong Yi(862188160), xyi007@ucr.edu,
Computer Science and Engineering Department, University of California Riverside, 900 University Ave, Riverside, California, USA, 92507.

2.2 Previous and Related Work

Thus far, machine learning for BGP anomaly detection has not grown very complex, and still uses relatively simple methods such as SVMs [7], or simple RNNs such as LSTMs (Long Short-Term Memory) [4] or GRUs (Gated Recurrent Unit) [2].

Other anomaly detection tasks have seen the use of more complex models involving deep learning such as, GANs (Generative Adversarial Networks) [3] [5] and Deep SVDD (Support Vector Data Description) [6]. These have yet to be applied to the specific domain of BGP anomaly detection.

As stated above, there currently is no literature which compares the inter-anomaly generalizability of these models, so we will have to design our experimental framework from scratch.

3 SOLUTION

Since previous work in BGP anomaly detection has not thus far examined the generalization performance between different anomalies, we begin by surveying the generalizability of different anomaly detection methods on our three separate worm datasets. Using the information gleaned from this survey, we select the method with the highest generalizability (which we define as the highest mean F1 score), and perform a feature ablation experiment with it in order to ascertain which networking features are most important for generalizing between different the three different worms.

3.1 Generalization Performance

In order to evaluate the inter-anomaly generalization performance of each method, we use three combinations of training and testing data where each method is trained on one of the anomalies and tested on the other two.

	Nimda		**Code Red**		**Slammer**		**Mean**	
Model	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
One-class Methods								
OC-SVM	0.102	0.185	0.253	0.404	0.237	0.384	0.198	0.324
Entropy OC-SVM	0.102	0.185	0.253	0.404	0.237	0.384	0.198	0.324
Autoencoder	0.903	0.501	0.608	0.282	0.745	0.530	0.752	**0.438**
Deep SVDD	0.921	0.479	0.804	0.354	0.726	0.388	**0.817**	0.407
Unsupervised Methods								
KNN	0.895	0.488	0.786	0.453	0.778	0.449	**0.820**	**0.463**
Isolation Forest	0.856	0.395	0.769	0.355	0.769	0.324	0.798	0.358
PCA-based	0.827	0.295	0.711	0.245	0.747	0.177	0.762	0.239
LOF w/ Feature Bagging	0.779	0.249	0.679	0.178	0.695	0.176	0.717	0.201
Angle-based Outlier Detector	0.892	0.488	0.768	0.372	0.772	0.382	0.811	0.414

Table 1. All of the scores shown are the scores when the model is trained on the listed dataset and evaluated on the remaining two datasets. LOF stands for Local Outlier Factor.

We note that autoencoders appear to be the best one-class method and KNN is the best unsupervised method for generalizing between these datasets. In general, the unsupervised methods outperform the one-class methods, which is an expected result, given that the unsupervised methods learn from both anomalous and unanomalous data, whereas the one-class methods are limited to learning their decision boundaries from the unanomalous data.

3.2 Feature Ablation Analysis

In order to explore *what* makes these models generalizable, we ablatively remove features from our data and note which features are most important for the best methods. We then note the most and least important features based on how much they decrease the F1 score of the methods.

Method	Most important	Least important
KNN	Number of withdrawn NRLI prefixes Number of announcements	Packet size Number of duplicate withdrawals
ABOD	Number of withdrawn NRLI prefixes Avg unique AS-path	Packet size Number of duplicate withdrawals
AE	Avg AS-path length Max AS-path length	Max AS-path length = 8 Number of duplicate withdrawals

Table 2. Top two most important and least important features for the three best methods: K-nearest neighbor, Angle-based Outlier Detector, Autoencoder

In general, it seems that the number of withdrawn NLRI prefixes and one of the AS-path length features are important to the generalization performance of the each of the models. This suggests that generalizable models must key in on features that are informative about the reachability of other ASs. Worms such as the ones investigated in these datasets typically affect networks most when attempt to rapidly replicate themselves, overloading the capacity of the network and leading to Denial of Service events. As such, it would follow that some nodes become unreachable and different AS-paths must be found.

Least important features included packet size, and the number of duplicate withdrawals. Packet size is negligible as packet size can vary normally for any number of reasons. That the number of duplicate withdrawals is unimportant to all three suggests that normal traffic sees a similar number of duplicate withdrawals as anomalous traffic.

4 CHALLENGES

Since nobody else has performed this type of analysis on BGP data before, we also had difficulty deciding how exactly to evaluate our methods. We had a hard time deciding on how to evaluate the generalizabilty of our methods.

Ideally, the anomalies in our dataset would have represented a diverse array of anomalies from different time periods, but almost every dataset we encountered used a different set of features, which would have made comparison and analysis impossible with our current experimental framework.

We had to write code to load the datasets properly because of how we chose to evaluate the data. We also to re-implement several of methods because there were no implementations online that worked with our data. Examples of these were the Fence-GAN [5], which had old code online, but it only worked with images and no longer ran on newer hardware. We reimplemented this from scratch only to find that it would take too long to train to a reasonable level of accuracy (which is why it is not included in the table). The autoencoder method was another example of this, where the papers describing it for anomaly detection lacked detail and a sample implementation, so we had to guess about how to fill in some of the blanks. Another challenge is the low accuracy and

F1-score performance by the OC-SVM and entropy OCSVM, it lead our team to consider which entropy or one class method would fit better on such type of BGP dataset.

REFERENCES

[1] Bahaa Al-Musawi, Philip Branch, and Grenville Armitage. 2016. BGP anomaly detection techniques: A survey. *IEEE Communications Surveys & Tutorials* 19, 1 (2016), 377–396.

[2] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. (12 2014). http://arxiv.org/abs/1412.3555

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[4] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (11 1997), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[5] Cuong Phuc Ngo, Amadeus Aristo Winarto, Connie Kou Khor Li, Sojeong Park, Farhan Akram, and Hwee Kuan Lee. 2019. Fence GAN: Towards Better Anomaly Detection. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* 2019-November (4 2019), 141–148. http://arxiv.org/abs/1904.01209

[6] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep One-Class Classification *(Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, Stockholmsmässan, Stockholm Sweden, 4393–4402. http://proceedings.mlr.press/v80/ruff18a.html

[7] Bernhard Schölkopf. 1998. SVMs - A practical consequence of learning theory. *IEEE Intelligent Systems and Their Applications* 13, 4 (7 1998), 18–21. https://doi.org/10.1109/5254.708428