Detecting forged AS paths from BGP graph features using Recurrent Neural Networks

Kevin Hoarau, Pierre Ugo Tournoux, Tahiry Razafindralambo

Detecting forged AS paths from BGP graph features using Recurrent Neural Networks

Kevin Hoarau
Université de La Réunion, LIM, France
kevin.hoarau@univ-reunion.fr

Pierre Ugo Tournoux
Université de La Réunion, LIM, France
pierre.tournoux@univ-reunion.fr

Tahiry Razafindralambo
Université de La Réunion, LIM, France
tahiry.razafindralambo@univ-reunion.fr

***Abstract*—The Border Gateway Protocol (BGP) is in charge of the route exchange at the Internet scale. Anomalies in BGP can have several causes (misconfigurations, outages and attacks). Forged AS paths are small scale and subtle attacks on BGP and therefore are hard to detect. In this paper we use a Machine Learning (ML) model applying a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) architecture to take the temporal aspect of BGP into account. We show that our ML model is able to detect forged AS path anomalies with an accuracy of 67% and a precision of 72%. These preliminary results outperform the existing proposals and allow us to think that ML on temporal graphs is worth investigating.**

I. Introduction

The Border Gateway Protocol (BGP) is the routing protocol standard on the Internet. A failure of the protocol could impact any service relying on the Internet. Such failures, namely BGP anomalies, happen for several reasons ranging from hardware failures to malicious attacks [1]. Depending on the attacker behavior, forged AS path anomaly may or may not result in a traffic black-hole for BGP. Forged AS paths are small scale and subtle attacks on BGP [2] since they may not stop the routing process. The ML model used to detect forged AS path anomalies can be fed with the BGP graph extracted from BGP data traces [4]. Indeed, there are two main types of features that can be used to feed a ML model for BGP: Statistical features or thirteen graph features. The authors of [4] show that graph features are well suited for BGP small scale anomaly detection. However, the anomaly detection method described in [4] does not take the temporal aspect of the BGP graph into account. To capture the temporal pattern of a forged AS path anomaly on the BGP graph, we use a ML model applying a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) architecture. Our LSTM machine learning model is able to detect forged AS path anomalies with an accuracy of 67% and a precision of 72%. These preliminary results outperform the existing proposals and allow us to think that ML on temporal graphs is worth investigating and might eventually address the forged AS path anomaly detection with decent performance in the future.

This project has received funding from the Région Réunion and the European Union - European Regional Development Fund (ERDF) as part of the INTERREG V - 2014-2020 program.

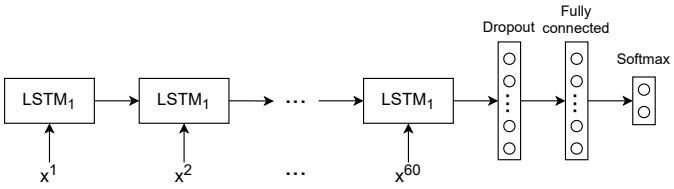

Fig. 1. Architecture of the RNN model

II. ML based anomaly detection

A. The RNN model architecture

The problem that we target in this work is a sequence classification problem. For an input sequence $X = x^1, ..., x^{60}$ our model should produce an output $Y = 1$ if an anomaly is detected within the sequence or $Y = 0$ otherwise. To address this problem, we use a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) architecture. In an RNN, for each element x^t an hidden state h^t is produced using a shared set of parameters θ and the hidden state of the previous element h^{t-1}:

$$h^t = f(h^{t-1}, x^t; \theta) \tag{1}$$

Moreover, RNN are able to leverage the temporal dimension of the data. LSTM are an improvement of RNN that allow to capture long-term dependencies by solving the issue of vanishing gradient.

The architecture of our RNN model is depicted in figure 1 and the hyperparameters used to train the model are listed in table I. First, the input sequence x^t is fed to a 1-layer LSTM network which produces a 32-dimensional output. Then, a dropout layer is used to help the generalization of the model. For each training sample and for each epoch, the dropout layer randomly zero out 25% of the 32-dimensional vector which prevents the overfitting of the model. Finally, a single layer fully-connected neural network with a softmax activation function is used to produce the output class prediction.

B. Experimental results

For both data collection and feature extraction we use BML [3]. We collect the positive samples one hour before and one hour after the event and for the negative samples we collect two hours data one day before the events. We evaluate the performance of our model using the following metrics:

Hyperparameter	Value
LSTM units	32
LSTM layers	1
Fully connected layers	1
Dropout rate	0.25
Nb. epochs	50
Learning rate	0.1

TABLE I
MODEL HYPERPARAMETERS

Test fold	Accuracy		F1 score		Precision		Recall	
	Mean	Std dev	Mean	Std dev	Mean	Std dev	Mean	Std dev
1	0.68	0.10	0.59	0.16	0.83	0.18	0.49	0.19
2	0.67	0.13	0.64	0.20	0.67	0.17	0.65	0.26
3	0.67	0.11	0.71	0.10	0.65	0.12	0.82	0.17
All	**0.67**	**0.11**	**0.65**	**0.16**	**0.72**	**0.18**	**0.65**	**0.25**

TABLE II
CROSS-VALIDATION RESULTS (13 FEATURES)

Accuracy, Precision, Recall and F1 score [7]. Our experiments are implemented using the *PyTorch* and *scikit-learn* library. The implementations are available online[1]. We use a 3-folds cross-validation scheme.

The results of the cross-validation are reported in table II when all the 13 graph features are used. The model achieves a correct classification performance with a mean accuracy of 67% over all the test folds. Moreover, we see that the mean recall (65%) and the mean precision (72%) are good indicator in the context of anomaly detection.

Features are not equally impacted by the anomaly. Feature selection is commonly used in the machine learning community to reduce the complexity of the model. In our study, four features stand out: the number of edges, the number of nodes, the pagerank and the percolation limit. We trained our model using only these four features as input. The results in table III show that we have the same accuracy (67%) and the same precision (72%) and that the recall stays stable (65%).

These performances can be explained by the fact that we removed irrelevant features that introduce noise in the training process. This reduction in the number of features also leads to a more efficient pipeline in a production environment. First, it allows to train a less complex model that is computationally more efficient. Second, less features need to be extracted to classify a sample.

Test fold	Accuracy		F1 score		Precision		Recall	
	Mean	Std dev	Mean	Std dev	Mean	Std dev	Mean	Std dev
1	0.66	0.11	0.56	0.18	0.80	0.19	0.46	0.20
2	0.66	0.15	0.67	0.15	0.66	0.15	0.69	0.17
3	0.69	0.14	0.71	0.13	0.68	0.14	0.77	0.18
All	**0.67**	**0.13**	**0.65**	**0.17**	**0.72**	**0.17**	**0.64**	**0.23**

TABLE III
CROSS-VALIDATION RESULTS (TOP 4 FEATURES)

[1]https://github.com/KevinHoarau/BGP_LSTM

C. Discussions, limitations and future works

First, collecting more data will improve the generalization of the model. Moreover, it could learn more specific patterns that could avoid false positives on events. Second, instead of a global approach where the entire BGP graph is used to extract a set of features, we could use more local approaches. This could be done, for example, by extracting regional graph or by extracting the k-hop neighborhood of the victim AS. Finally, graph neural networks (GNN) [5] and temporal graph neural networks (TGN) [6] may allow to leverage BGP graph structures without the need of intensive graph features computations. The resulting model could be adapted for the online detection of forged AS paths.

III. CONCLUSION

This paper describes a machine learning model that takes into account the evolving graph of BGP to detect forged AS path (path hijacking) anomalies. Especially, we use a machine learning model applying a Long Short-Term Memory (LSTM) based Recurrent Neural Network (RNN) architecture to capture the temporal aspect of BGP. We evaluate the performance of our model by computing the accuracy, the precision, the recall and the F1 score of the detection.

We show that the temporal aspect could help detect forged AS paths. We also show that with our model we are able to detect forged AS path anomalies with an accuracy of 67% and a precision of 72%. Moreover, we show that with a limited number (4) of graph features instead of 13, we are able to detect forged AS paths with the same accuracy (67%) and precision (72%). This model and its simplification to only four graph features allows us to think that in the future our model could be adapted to perform online detection of BGP forged AS path anomalies.

REFERENCES

[1] B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 377–396, FebJan 2017. [Online]. Available: http://dx.doi.org/10.1109/comst.2016.2622240

[2] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "Bgp hijacking classification," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, 2019, pp. 25–32.

[3] K. Hoarau, P.-U. Tournoux, and T. Razafindralambo, "BML: an efficient and versatile tool for BGP dataset collection," in *WS22 IEEE ICC 2021 the 3rd International Workshop on Data Driven Intelligence for Networks and Systems (WS22 ICC'21 Workshop - DDINS)*, Montreal, Canada, Jun. 2021.

[4] ——, "Suitability of graph representation for bgp anomaly detection," in *2021 IEEE 46th Conference on Local Computer Networks (LCN) (LCN 2021)*, Edmonton, Canada, Oct. 2021.

[5] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[6] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, F. Monti, and M. Bronstein, "Temporal graph networks for deep learning on dynamic graphs," *arXiv preprint arXiv:2006.10637*, 2020.

[7] C. Sammut and G. I. Webb, *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.