BGP Anomalies Classification using Features based on AS Relationship Graphs

Thales B. Paiva, Yaissa Siqueira, Daniel Macêdo Batista, R. Hirata Jr. and R. Terada
Department of Computer Science
University of Sao Paulo (USP), Brazil
{tpaiva,yaissa.siqueira,batista,hirata,rt}@ime.usp.br

2021 IEEE Latin-American Conference on Communications (LATINCOM) | 978-1-6654-4035-6/21/$31.00 ©2021 IEEE | DOI: 10.1109/LATINCOM53176.2021.9647824

***Abstract*—Ensuring the correct behavior of the Border Gateway Protocol (BGP) is essential for keeping a good quality of service on the internet. When an anomalous behavior is detected, operators of border gateways need to classify it correctly into a direct (intended or unintended) anomaly, an indirect anomaly, or a link failure. This classification helps to understand its cause and act upon it. Recently, some techniques for the classification of BGP anomalies using machine learning models were proposed. However, we notice some limitations of these classification models that make it unclear if they can be used in the real world to classify new anomalies. This paper presents a new model with good performance when classifying BGP events not seen in its training. Our model is based on Long Short-Term Memory (LSTM) networks and uses new features based on inferred relationships between Autonomous Systems (ASes) to classify sets of BGP update messages. The model classifies samples from new events achieving 91% of accuracy and F1 scores of 1.00, 0.93, and 0.80 for direct anomalies, indirect anomalies, and link failure, respectively.**

***Index Terms*—BGP, Anomaly Classification, LSTM.**

I. Introduction

The Border Gateway Protocol (BGP) is the most commonly used routing protocol. The BGP determines a communication standard between Autonomous Systems (AS) routers when propagating Network Reachability Information (NRI) to their neighbor ASes. This fact makes BGP-speaking routers an exciting target for attackers, and there are many documented attacks against these routers. Attackers targeting these routers have varying objectives, such as Denial-of-Service (DoS) and Person-in-the-Middle (PitM) attacks [1].

The core component of BGP is the *update messages* that an AS router sends to its neighbors. These messages usually contain an IP prefix, a path vector, and additional information to allow routers to tweak their routing tables according to their policies. The exchanged messages contain valid information, but BGP has no mechanism to ensure a valid message. The Internet Engineering Task Force led some proposals such as RPKI [2] and BGPSec [3], which use a Public Key Infrastructure (PKI) to ensure that ASes only announce valid paths and prefixes. However, they require more powerful hardware and yield a much heavier protocol, making for a more difficult adoption in the short term.

Update messages that do not contribute to BGP's purpose of propagating valid NRI are defined as anomalous [4]. Detection of BGP anomalies is a difficult task since abrupt changes in metrics of BGP messages may not always indicate malicious activity such as BGP hijacking attacks and may be the result of traffic engineering for network load balancing or consequence of some link failure by natural causes. When anomalous BGP messages are detected, it is vital to understand their cause to act accordingly. For example, if the anomaly is the result of prefix hijacking, operators of BGP routers should not propagate the hijacked prefix. Recently, supervised machine learning models have proven to be a promising approach for BGP anomaly detection [5]–[13].

Different from BGP anomaly detection, the classification of anomalies has received much less attention. Concerning direct anomalies only, Cho et al. [14] showed how to classify hijacking events into various types, which can separate intended and unintended misbehavior. Fonseca [15] uses Long Short-Term Memory (LSTM) to classify sets of BGP messages, and more recently, Cheng et al. [16] proposed a powerful Multi-Scale LSTM model for multi-class classification of sets of BGP messages. In both works, the authors consider specific events such as the spreading of the Slammer worm or the 2003 Moscow blackout.

The previous works have some limitations, however:

1) The classification is limited to events, not anomaly type (direct, indirect, and link failure).
2) The model is tested with data from events seen in the training phase.
3) The most widely used public datasets of BGP anomalies consist of a small number of events, that are mostly outdated (e.g. the effect of Nimda worm in 2001).

Together, these limitations make it difficult to assess if their approaches can generalize knowledge to classify events not seen before into the three types, which are the main application of BGP anomaly classification.

a) Contribution: In this paper we propose an LSTM model for BGP anomaly classification that uses a small set of features based on the graph of ASes. We used data from six events corresponding to direct, indirect, and link-failure anomalies to train our model. We evaluate the model's performance considering one event of each type, different from the events used in training. To the best of our knowledge, this is the first classification model tested against events not seen in the training phase. Furthermore, the code for the model and feature extraction is available at https://github.com/thalespaiva/bgp-anomaly-classification.

b) Paper organization: We briefly discuss related work in Section II. In Section III we describe our feature extraction process. Section IV contains the description of the LSTM model and information on how we trained it. Next, the performance of the model is is evaluated in Section V. Finally, we conclude and discuss future work in Section VI.

II. Related Work

BGP anomaly detection is a widely studied problem, and there are several approaches to deal with the problem [4], [17]. Al-Musawi et al. [4] discuss the importance of classifying the type of anomaly since they often require a different set of responses. They consider the main limitation of BGP detection mechanisms up to 2016 to be their inability to classify the type of anomaly and propose a classification into *direct* (intended, or unintended), *indirect*, and *link failure*. Direct anomalies happen by the misbehavior of BGP routers and can be *intended*, such as BGP hijacking, or *unintended*, such as typos in prefixes introduced by operators of border routers. Anomalous behavior that results from BGP protocol manipulation is deemed indirect and caused by the spread of a worm over the Internet. Link failures are anomalous behaviors that happen after an AS or Internet Exchange Point (IXP) loses connection to the Internet. Moreover, it is often a result of a blackout or a natural disaster such as an earthquake.

In 2019, Cho et al. [14] focused on classifying BGP hijacking events, which correspond to direct BGP anomalies, into four types: *typos*, *prepending mistakes*, *origin changes*, and *forged AS paths*. To classify BGP hijacking events into these types, the authors propose a Random Forest model on features based on AS hegemony [18], a metric that represents the importance of an AS for paths to a given prefix on the Internet. Moreover, they build a ground-truth set of available BGP hijacking events and obtain a 95.71% classification accuracy. Even though the authors point to some inherent limitations of features based on AS hegemony, their work does not sufficiently address how their classifier performs when considering real-time BGP updates messages. Furthermore, since they test the model only with known hijacking events, it is unclear if it would deal with regular updates without raising many false alarms.

A recent work by Li et al. [9], [12] proposes machine learning models to detect and classify BGP anomalies. Their work is better suited for real-time detection and classification, as they use features that are easy to compute over many updates in a given time interval. However, a critical limitation of their work is that the datasets consider only three events, of which two occurred before 2005. Therefore, to increase confidence in the robustness of their model, it would be essential to test their approach under more recent anomalous events.

Sanchez et al. [13] introduced the use of graph-based features, such as graph number of cliques and graph centrality metrics, for anomaly detection. Using the same set of features, the authors compared the performance of some machine learning models: Support Vector Machines (SVM) Multi-Layer Perceptron (MLP), Naive Bayes, and Decision Trees. The authors trained and tested their model using four well-known BGP anomalies, obtaining around 89% balanced accuracy.[1] Apart from the limited dataset, one limitation of this work is that some of the graph features are very expensive and not practical for real-time detection. Furthermore, the authors did not consider the proposed graph-based features for classification, only for detection.

Addressing the absence of frameworks able to classify the anomalies into the categories considered by Al-Musawi et al. [4], Fonseca [15] tested different model architectures to detect and to classify BGP anomalies. Their most successful models consisted of a Wavelet deconvolution layer, a Convolutional Neural Network (CNN) and LSTM for detection, and stacked LSTM networks for classification. Furthermore, they show that domain classification tends to be easier than detection because the finer granularity of the labels makes it easier for the models to learn the different models for each label correctly.

In 2020, Shapira and Shavitt [11] proposed an elegant hijack detection mechanism that consists of an LSTM network whose embedding layer is an application of Word2Vec [19] to sets of AS paths in BGP tables and announcements. The embedding layer, called BGP2Vec, was already shown to be able to capture the relationships between ASes with some accuracy [20]. When testing their model using the ground-truth datasets by Cho et al. [14], the authors were able to detect around 67% of the hijacking events, with varying accuracy depending on the event type. Notice that the authors did not consider the classification of hijack events or even general detection of anomalies, focusing only on detecting hijacking events.

One important problem of previous works is that, in general, there is no standard dataset, or datasets, to compare different models. Therefore, it is tough to understand the impact of features and models on the overall performance when detecting and classifying anomalies. Furthermore, the source code for the vast majority of the tools is not usually publicly available. Fonseca et al. [21] and, more recently, Hoarau et al. [22] made an important step in this direction. Fonseca et al. [21] proposed an open-source framework for data collection and feature extraction of BGP announcements, together with a relatively large dataset of well-known anomalous events, in 2019. More recently, Hoarau et al. [22] proposed a similar framework that promises two advantages: it comes with a larger number of features, and it is easier for the user to add new features.

Closely related to our work are the models studied by Fonseca [15] and Cheng et al. [16]. These authors studied the detection and classification of BGP anomalies and achieved good results for what they propose. The main limitation of both works is that the authors do not consider the performance of their models to classify samples from events out of the training

[1]The mean between true positive and true negative rates.

set. Therefore, it is not easy to know if their models are good in real scenarios where the classifier must classify new events.

III. Feature Extraction from Anomalous Events

In this section, we present the features considered for classification and describe how to compute them.

A BGP *update message* can be downloaded using CAIDA's BGPReader tool[2] and contains several fields. Table I shows an example from RIS project collector `rrc04`, which corresponds to CERN's IXP in Geneva, Switzerland.

Unlike other BGP anomaly classification [15], [16] mechanisms based on deep learning, we consider, together with the raw *update messages*, information on the AS relationships as paths, prefix fields, and their approximate geographical location. This decision contrasts with most of the works on anomaly detection and classification, where features are mostly based on counting the number of messages of a given type, i.e., termed volume-based features.

TABLE I: An example of a BGP announcement update as downloaded by BGPReader.

Field	Value
Record type	Update
Element type	Announcement
Timestamp	23/01/2003 00:00:08 (UTC)
Project	RIS
Collector	rrc04
Router	None
Router IP	None
Peer ASN	6893
Peer IP	192.65.185.144
Prefix	64.30.64.0/19
Next hop	192.65.185.144
Origin AS	14900
AS path	6893, 12541, 3561, 209, 14900
Communities	-
Old state	-
New state	-

In the following subsections, we first give a high-level description of the feature extraction procedure and then describe the events we used to build the datasets.

A. Features based on the AS graph and relationship edges

There are two main difficulties when computing features based on AS relationships: (1) the information is not publicly available, and (2) the relationships change from time to time. Our solution to the first problem is to use Gao's [23] AS relationship inference. We propose building the AS relationship graph on-demand as described next to deal with AS relationships' dynamic nature.

Suppose we want to classify sets of updates occurring from 12/21/2010 to 12/22/2010. We first download the Routing Information Base (RIB) of well-known BGP collectors such as RIS or RouteViews from the first week of the month right before the target, which corresponds to November 2010. We extract only the AS paths from the RIB and use Gao's algorithm [23] to classify the relationships between the ASes involved in these paths into customer-to-provider, provider-to-customer, sibling-to-sibling, and peer-to-peer.

The proposed design avoids using fixed relationship datasets, which may be highly unrealistic. Suppose one such dataset was built in 2020, then every feature computation for updates from before 2020, would leak information from the future to the past.

We can now compute features such as the average number of provider-to-customer edges or the average degree of the ASes involved in the path field of BGP updates. Additionally, we can detect and count valleys in the AS paths, which are related to BGP hijackings events [11], [14].

B. Features based on AS geographical location

We also consider two features that make use of the location of ASes. In order to calculate these features, we first generate a dataset containing the central latitude and central longitude of the country where the AS is registered.

The first feature tries to detect geographical changes in ASes that announce a given prefix. For each prefix seen in the BGP updates, we store the last AS that announced it. If the AS announcing the prefix changes at some point, we calculate the distance between the new AS announcing it and the last AS that was saw announcing this prefix. The feature then consists of the average distance over all announcements.

The second feature is similar to the previous one, but instead of verifying the prefix as a whole, we separate it into IP address and prefix length. In this case, we have to compute the distance between the ASes announcing the same IP address but with different prefix lengths.

C. Extracting Features from Anomalous Events

To train and test our classifier, we use the same events considered by Fonseca [21], who uses the largest set of anomalous events among all previous works. To build our dataset, we downloaded all BGP update messages from the duration of the events using CAIDA's library PyBGPStream. Then, for each event, we extracted the paths from RIBs corresponding to the first week of the month before each anomaly and built the AS relationship graph as described in Section III-A. Finally, we computed the features for each set of 1 minute of updates, which yields our time series' data points.

Table II presents the events considered in this work, together with their type, duration, total number of BGP updates during the event, and the number of data points. Notice that the number of BGP updates does not impact the number of data points, which depends only on the duration of the event in minutes.

A high-level description of the features is given in Table III. Their computation is efficient because the features can be computed faster than updates are observed. Furthermore, we designed the scripts for data collection and feature extraction in Python to make it easy to add new events and features to extract.

[2]https://bgpstream.caida.org/docs/tools/bgpreader

TABLE II: Anomalous events considered in this work.

Event	Type	Collector	Start time	Finish time	Total BGP Updates	Data Points
AS9121 RTL	Direct	rrc05	09:20 24/12/04	10:03 24/12/04	3,060,307	43
AWS Route Leak	Direct	rrc04	17:10 22/04/16	20:00 22/04/16	29,572,808	170
Malaysian Telecom	Direct	rrc04	08:42 12/06/15	10:24 12/06/15	16,554,716	102
Code Red v2	Indirect	rrc04	10:00 19/07/01	20:00 19/07/01	2,090,162	600
Nimda	Indirect	rrc04	13:00 18/07/01	12:00 21/07/01	3,095,343	4260
Slammer	Indirect	rrc04	05:31 25/01/03	19:59 25/01/03	2,320,859	868
Moscow Blackout	Link failure	rrc05	04:40 25/05/05	07:40 25/05/05	7,613,061	180
Japan Earthquake	Link failure	rrc06	09:13 11/03/11	15:39 11/03/11	545,397	386

The most complex features to compute are the ones that navigate all ASes in the AS path field (e.g., the average degree of ASes found in all paths). The time to compute the feature depends on the number of updates seen in the one-minute interval. This dependency makes it more costly to compute features on the most recent events, such as AWS Route Leak, when a considerably larger number of updates than in 2001, when Nimda and Code Red were released, are available.

IV. Model Training

In our work, we have focused on building a model to learn how to classify a given sequence of observed data points into the classes indirect, direct or link failure of BGP anomalies.

LSTM is a recurrent neural network architecture suitable to analyze time series with arbitrary gaps in their temporal sequence [24]. In other words, it is capable of learning about events that may have a considerable time distance between each other. For example, our dataset is configured by anomalies on BGP that happened in different years between 2001 and 2016. Therefore, LSTM models seem to be an interesting choice.

Our neural network starts with a one-dimension convolutional layer (Conv1D) with 32 filters and a kernel of size two, followed by a one-dimensional max-pooling layer (MaxPooling1D) with pool size and strides of value two, as well. In sequence, we have an LSTM layer and a Dropout layer with a dropout rate of 0.2. To finish, we have a dense layer with a softmax activation function. The model has a total of 54,623 parameters, of which all of them are trainable. Table IV shows a summary of the layers in our model.

To train our model, we have prepared a dataset containing the calculated features of the selected events: AS9121 Routing Table Leak, AWS Leak, Nimda, Code Red II, and Moscow Blackout. The features described in Table III were calculated for batches of one minute of BGP updates. Therefore, each dataset entry corresponds to a batch evaluation, which entails our time series. The next step was to cut each event's sequence into smaller lengths of 10 minutes, that is, a sequence of 10 data points of features computed for 1 minute. All the slices were non-overlapping, as overlapping sequences are highly correlated and may quickly lead to overfitting. We have labeled each slice with its corresponding class, and there were no slices with mixed classes.

The dataset is split into training, validation, and test sets following the ratios of 70%, 20%, and 10%, respectively. When splitting the dataset, we made sure that each set contained at least one sequence of each class. Finally, we have trained our model with batches of size one and considered ten epochs, using categorical cross-entropy [24] and the Adam optimizer [24] with a learning rate of 0.0067.

Before the feature selection, we fixed which events were used for training and testing and final validation. This separation beforehand makes our feature selection procedure less prone to insert data leakage and better assess our model's robustness. We evaluated our model performance using the test set and the events missing from the training list: Japan Earthquake, Slammer, and Malaysian Telecom. Our intention with this second validation was to confront our model with events that it has never seen before.

We use standard metrics for classification, such as precision, recall, accuracy, and F1-score. Precision identifies the proportion of how many items classified by a model are correctly classified. Recall is similar to precision but differs as it can be interpreted as how frequent, for each class, the model is correct when it classifies a sample into the given class. The F1-score is a harmonic average between precision and recall and summarizes both metrics. For this reason, we have mainly used F1-score to evaluate our model performance. Additionally, we have also used accuracy, which gives us a ratio of correctly predicted instances over the total instances but being careful not to misinterpret it when dealing with highly unbalanced datasets.

Let TP, TN, FP, FN be the total of True Positive, True Negative, False Positive and False Negative classified instances, respectively. The formal definitions of the above metrics can be written as:

- Precision $= \frac{\text{TP}}{\text{TP+FP}}$.
- Recall $= \frac{\text{TP}}{\text{TP+FN}}$.
- F1 Score $= 2\frac{(\text{Recall}\times\text{Precision})}{\text{Recall+Precision}}$.
- Accuracy $= \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}$.

V. Results and Discussion

In this section, we evaluate the performance of the model for the classification of BGP anomalies. We divide this section into two parts. First, we show the results regarding the classification of samples from the events used to train the model. Then we consider the classification of events not seen in the training phase.

TABLE III: Features considered in this work.

Type	Feature	Has been used before for classification? (Considering previous works [15], [16])
Volume	Number of announcements	Yes
AS path	Maximum length of AS paths	Yes
AS path	Average length of AS paths	Yes
AS path	Maximum length among unique AS paths	Yes
AS path	Average length among unique AS paths	Yes
AS graph	Variance of the degree of ASes found in all paths	No
AS graph	Average degree of ASes found in all paths	No
AS relationship graph	Average number of edges not in AS graph	No
AS relationship graph	Average number of peer-to-peer edges in paths	No
AS relationship graph	Average number of customer-to-provider edges in paths	No
AS relationship graph	Average number of provider-to-customer edges in paths	No
AS relationship graph	Average number of sibling-to-sibling edges in paths	No
AS relationship graph	Average number of non valley-free paths	No
Prefix	Average number of bits in prefix (IPV4 only)	No
Prefix	Maximum number of bits in prefix (IPV4 only)	No
AS geographical location	Average geographical distance between last two ASes that announced the same prefix	No
AS geographical location	Average geographical distance between last ASes that announced a similar prefix	No

TABLE IV: Model's layers summary.

Layer type	Output dimension	Number of parameters
Conv1D	(10, 32)	1,120
MaxPooling1D	(5, 32)	0
LSTM	(100)	53,200
Dropout	(100)	0
Dense	(3)	303

A. Classification of samples from events used for training

In previous works on BGP anomaly classification [16], [21], the authors test the performance of their models using part of the data used for training. Furthermore, they test the classification of samples into an event, not into an anomaly class. Even though it is important to understand the model and effect of different features to learn their essential characteristics, we also believe it is important to verify if the model can generalize its knowledge of the different types of anomalies.

Table V shows the performance of our model when classifying samples into direct, indirect, or link failure anomalies. We can see that it perfectly classified the training set, even though the samples were highly unbalanced. Unfortunately, since we used 10% for testing, only 5 instances were not from the indirect type.

TABLE V: Performance of the model when classifying data from events that were used for training.

Class	Precision	Recall	F1 score	Total instances
Direct	1.00	1.00	1.00	3
Indirect	1.00	1.00	1.00	49
Link failure	1.00	1.00	1.00	2

The main question now is whether the model could generalize knowledge on the anomaly types or not. However, Table II shows us that both indirect events used for training occurred in 2001, and within the same month. Even though it is tempting to believe that the model performed well, it may be the case that the model learned the year of the anomaly, not the anomaly type.

Therefore, we believe that it is essential to analyze the model's performance when classifying events that are not considered in the training data. We do this analysis in the next subsection.

B. Classification of samples from new events

We now present the main contribution of this work: the classification of samples from events not seen by the model in the training phase. Table VI presents the results of the model when classifying data from new events. We can see that it perfectly classified direct events. However, even though its performance was good for indirect and link failure events, these two types appear to be slightly confused by the model.

TABLE VI: Performance of the model when classifying data from new events, which were not used for training.

Class	Precision	Recall	F1 score	Total instances
Direct	1.00	1.00	1.00	11
Indirect	0.87	1.00	0.93	88
Link failure	1.00	0.67	0.80	39

Table VII presents a confusion matrix to understand better how indirect and link failure events are classified. Notice that, even though it has 100% precision for link failure classification, it incorrectly classified about 33.3% of the samples as indirect (13 out of 39 samples). A possible explanation for the misclassification of link failures is the limited data used for training, which consists of only one event, together with the large gap between the dates when the events used for training and testing occurred.

These results suggest that the model can generalize what it learned from the training data to new events. This generalization ability is an important step for the adoption of these classifiers for real-world applications.

TABLE VII: Confusion matrix of the classification of samples from new events. Accuracy of 91%.

	Direct	Indirect	Link Failure
Direct	11	0	0
Indirect	0	88	0
Link failure	0	13	26

VI. Conclusion and Future Work

The main contribution of this work is the first BGP classification model that appears to be robust enough to classify events that were not part of the training dataset. The proposed model uses a set of features based on the inferred AS relationship graph, together with a simple LSTM network. Our code and datasets are publicly available, and we encourage researchers to experiment with them.

It is important to validate our results using larger sets of BGP anomalies and even a more diverse set of collectors for future work. Concerning the feature extraction, it would be interesting to assess the effect of better AS relationship inference algorithms such as Problink [25] or BGP2Vec [20] on the quality of the features.

Acknowledgments

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq proc. 465446/2014-0, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, FAPESP proc. 14/50937-1, and FAPESP proc. 15/24485-9. It is also part of the FAPESP proc. 18/22979-2 and FAPESP proc. 18/23098-0.

References

[1] Z. Nabi, "The Anatomy of Web Censorship in Pakistan," in *Proceedings of the 3rd USENIX Workshop on Free and Open Communications on the Internet (FOCI 13)*, 2013.

[2] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol – RFC6810," 2013, https://datatracker.ietf.org/doc/rfc6810/. Accessed at October 28, 2021.

[3] M. Lepinski and K. Sriram, "BGPSEC Protocol Specification – RFC8205," 2017, https://datatracker.ietf.org/doc/rfc8205/. Accessed at October 28, 2021.

[4] 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, 2016.

[5] Q. Ding, Z. Li, P. Batta, and L. Trajković, "Detecting BGP Anomalies using Machine Learning Techniques," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 003 352–003 355.

[6] P. Batta, M. Singh, Z. Li, Q. Ding, and L. Trajković, "Evaluation of Support Vector Machine Kernels for Detecting Network Anomalies," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018, pp. 1–4.

[7] Q. Ding, Z. Li, S. Haeri, and L. Trajković, "Application of Machine Learning Techniques to Detecting Anomalies in Communication Networks: Datasets and Feature Selection Algorithms," in *Cyber Threat Intelligence*. Springer, 2018, pp. 47–70.

[8] A. L. G. Rios, Z. Li, G. Xu, A. D. Alonso, and L. Trajković, "Detecting Network Anomalies and Intrusions in Communication Networks," in *Proceedings of the IEEE 23rd International Conference on Intelligent Engineering Systems (INES)*, 2019, pp. 000 029–000 034.

[9] Z. Li, A. L. G. Rios, G. Xu, and L. Trajković, "Machine Learning Techniques for Classifying Network Anomalies and Intrusions," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.

[10] Z. Li, Q. Ding, S. Haeri, and L. Trajković, "Application of Machine Learning Techniques to Detecting Anomalies in Communication Networks: Classification Algorithms," in *Cyber Threat Intelligence*. Springer, 2018, pp. 71–92.

[11] T. Shapira and Y. Shavitt, "A Deep Learning Approach for IP Hijack Detection Based on ASN Embedding," in *Proceedings of the Workshop on Network Meets AI & ML*, 2020, pp. 35–41.

[12] Z. Li, A. L. G. Rios, and L. Trajković, "Detecting Internet Worms, Ransomware, and Blackouts Using Recurrent Neural Networks," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 2165–2172.

[13] O. R. Sanchez, S. Ferlin, C. Pelsser, and R. Bush, "Comparing Machine Learning Algorithms for BGP Anomaly Detection using Graph Features," in *Proceedings of the 3rd ACM CoNEXT Workshop on Big DAta, Machine Learning and Artificial Intelligence for Data Communication Networks*, 2019, pp. 35–41.

[14] S. Cho, R. Fontugne, K. Cho, A. Dainotti, and P. Gill, "BGP Hijacking Classification," in *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)*, 2019, pp. 25–32.

[15] P. C. d. R. Fonseca, "A Deep Learning Framework for BGP Anomaly Detection and Classification," Ph.D. dissertation, Universidade Federal do Amazonas, 2020.

[16] M. Cheng, Q. Li, J. Lv, W. Liu, and J. Wang, "Multi-Scale LSTM Model for BGP Anomaly Classification," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 765–778, 2021.

[17] A. Mitseva, A. Panchenko, and T. Engel, "The State of Affairs in BGP Security: A Survey of Attacks and Defenses," *Computer Communications*, vol. 124, pp. 45–60, 2018.

[18] R. Fontugne, A. Shah, and E. Aben, "The (Thin) Bridges of AS Connectivity: Measuring Dependency using AS Hegemony," in *Proceedings of the International Conference on Passive and Active Network Measurement*, 2018, pp. 216–227.

[19] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and Their Compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 2*, 2013, pp. 3111–3119.

[20] T. Shapira and Y. Shavitt, "Unveiling the Type of Relationship Between Autonomous Systems Using Deep Learning," in *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2020, pp. 1–6.

[21] P. Fonseca, E. S. Mota, R. Bennesby, and A. Passito, "BGP Dataset Generation and Feature Extraction for Anomaly Detection," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*, 2019, pp. 1–6.

[22] K. Hoarau, P. Tournoux, and T. Razafindralambo, "BML: An Efficient and Versatile Tool for BGP Dataset Collection," in *Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops)*, 2021.

[23] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 6, pp. 733–745, 2001.

[24] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org. Accessed at October 28, 2021.

[25] Y. Jin, C. Scott, A. Dhamdhere, V. Giotsas, A. Krishnamurthy, and S. Shenker, "Stable and Practical AS Relationship Inference with ProbLink," in *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, 2019, pp. 581–598.