# Inferring the Confidence Level of BGP-based Distributed Intrusion Detection Systems Alarms

Renato S. Silva*, Felipe M. F. de Assis*, Evandro L. C. Macedo*, Luís Felipe M. de Moraes*
*High-Speed Networks Laboratory, PESC/COPPE, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
E-mails: renato@ravel.ufrj.br; assis@ravel.ufrj.br; evandro@ravel.ufrj.br; moraes@ravel.ufrj.br

2023 7th Cyber Security in Networking Conference (CSNet) | 979-8-3503-4287-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/CSNET59123.2023.10339702

***Abstract*—As the protocol that enables the global routing system of the Internet, Border Gateway Protocol (BGP) is increasingly becoming a multipurpose protocol. However, it keeps suffering from security issues regarding bogus announcements for malicious goals. Some of these security breaches are especially critical for distributed intrusion detection systems that use BGP as the underlay network for interchanging alarms. In this sense, assessing the confidence level of detection alarms transported via BGP messages is critical to prevent internal attacks. Most of the proposals addressing the confidence level of detection alarms rely on complex and time-consuming mechanisms that can also be a potential target for further attacks. In this paper, we propose an out-of-band system based on machine learning to infer the confidence level of the intrusion alarms sent using BGP headers. Tests using a synthetic data set containing the indirect effects of a widespread worm attack over the BGP network show promising results considering well-known performance metrics, such as recall, accuracy, receiver operating characteristics (ROC), and f1-score.**

***Index Terms*—DIDS, Machine Learning, BGP, Distributed Intrusion Detection System**

## I. Introduction

The essential function of BGP is to control how IP packets are routed across the Internet through exchanging routing and reachability information between autonomous systems (AS). New prefixes announced by an AS to its AS-peers continuously propagate around the Internet. The role of BGP as "the glue of the Internet" also keeps pushing its evolution along the time to support other routing protocols in the case of MP-BGP and new features such as BGP-FlowSpec. However, despite the several improvements of BGP since its worldwide implementation, it is still extremely vulnerable to both malicious attacks and human error [1]. Thus, besides its importance to assure confident reachability, assessing the confidence level of these messages also helps to improve self-defense mechanisms of distributed intrusion detection systems that use BGP as their underlying network [2].

Although the myriad of approaches addressing security issues on the BGP protocol framework, resource public key infrastructure (RPKI) [3] has prevailed as the *de facto* approach. The distributed public database of RPKI is considered to be a highly secure and reliable mechanism, but one cannot guarantee its accuracy [4]. Actually, RPKI grounds full-scale architectures that provide origin and topology authentication, such as route origin validation (ROV), which uses route origin authorizations (ROAs) – digitally signed objects that fix an IP address to a specific network or autonomous system – to establish the list of prefixes a network is authorized to announce. However, RPKI still needs a third-party certification in spite of its remarkable reliability. In addition, implementing RPKI on the entire Internet is far from a simple task. According to the analysis proposed in [5], small ASs have not considered performing the origin validation at the time of this writing. At this point, it is worth reminding that the security of a chain system is only as strong as its weakest link.

Another important feature of BGP is the BGP-FlowSpec [6] that provides an extension to distribute granular flow specifications to network routers. Although BGP-FlowSpec already implements validation mechanisms, it is still possible for a malicious or compromised AS to announce fake FlowSpec updates [7]. Furthermore, due to the presumed numerous detection members distributed across the Internet, it would be difficult to infer the reliability of the BGP-FlowSpec messages originating from a given federation. To this end, using machine learning (ML) capabilities to support defense decisions is a good option.

In this paper, we propose an ML model to infer the confidence level ($C_L$) of BGP-based alarms that arrive at a given AS to be combined, according to the intrusion detection federation described in [2]. The ML model uses a 15-attributes data set built from mandatory fields of each BGP header to yield $0 \leq C_L \geq 1$, which indicates how reliable the BGP message is. This confidence level can be combined with the mean positive-prediction value (PPV) of the detection federation to support defense decisions. Results based on some well-known performance metrics such as *recall*, *accuracy*, *receiver operating characteristics* (*ROC*), and *f1-score*, show that the model is able to perform well for new input data.

The remainder of this paper is organized as follows. In Section II, we position our approach in relation to the main works related to improving the BGP security framework. In Section III we explain the process adopted to build the data set used to train the machine learning model. Section IV describes the unsupervised and supervised tests using the data set built in III and present some relevant performance results obtained from that. In Section V, we close the article with an objective analysis correlating the results obtained from the models with the paper's contributions.

## II. Related Works

Network topology changes provoked by the effects of some kind of attack have been very well studied in academia [8].

The survey proposed in [9] presents a comprehensive approach regarding BGP anomalies, including a canonical taxonomy classifying them according to their intentionality and causality. The study in [9] relates some of the most important global worm attacks such as Nimda and Code Red II with large spikes of BGP messages observed during these attacks. Another global worm attack that provoked a dramatic increase in BGP update announcements – 100 times bigger in the case of some ASs – was Slammer. In all these cases, even though the attacks do not intend to directly compromise the BGP network, their effects certainly did. Taking advantage of this unnatural behavior, several works have been proposed BGP labeled data sets to train machine learning models aiming to detect – and sometimes classify – attacks. The labeled data set proposed in [10] has 35 features distributed as direct, indirect, volume, and statistical. To label their data set, P. Fonseca et al. [10] correlated information from some global events that affected Internet traffic, such as worldwide worm attacks, the 2005 Moscow blackouts, the 2011 earthquake in Japan, the 2015 AWS route leak, with BGP historical logs from Ripe Project. Performance tests using new data show promising results to detect and classify the anomalies between attacks and events. In the same track, the approach proposed in [11] relies on data mining models to detect abnormal behaviors on the global routing infrastructure, by learning from a labeled 15-features data set. According to the authors, abnormal events such as large-scale power outages, and worm attacks can affect the global routing infrastructure and consequently create regional or global Internet service interruptions. Graphical results show that the system is able to yield accurate classification in near real-time.

An autonomous system (AS) deals with an enormous number of BGP updates every day. These update messages aim to inform the AS route on how to reach a new prefix on the Internet or delete it from its routing table. In such a large amount of data, it is common to observe mistaken messages containing incorrect information as a result of misconfigured ASs or even fake messages originated by malicious attempts that can seriously damage Internet routing. The detector proposed in [12] relies on machine learning techniques to reproduce the "gut feeling" of a network expert to classify BGP updates as either attacks or misconfigured messages. The idea is to train auto-encoders to generate only clean data as opposed to attack data, which does not share the same essential features. However, due to the difficulties in obtaining a real data set containing collections of anomalous BGP announcements, the authors crafted their own attack data by editing random updates. The tests using the f-score as the main performance metric, which is a measure of the model's accuracy, show promising results.

The system proposed in [13] requires no protocol modifications and utilizes existing monitoring infrastructure to infer the consistency of the BGP announcements according to the network topology. Utilizing geographical location data from the "whois" database and the topological information, the system builds an AS connectivity graph, classifying all autonomous system nodes as either core or periphery nodes. Violations are detected by checking if the sequence of autonomous systems satisfies the constraints dictated by their observations regarding the AS_PATH attribute of update messages. Although the proposed system can be applied immediately and does not interfere with the existing infrastructure, it presents topological restrictions that permit some attacks to succeed.

The work presented in [14] reveals that malicious activity is not necessarily evenly distributed across the Internet. Rather, the model based on applying Jaccard similarity shows that there are ASs solely engaged in malicious activity. For example, while a majority of ASs have little to no malicious activity, a few ASs have as much as $0.5 \rightarrow 10\%$ of their IP addresses engaged in malicious activities. Another relevant result refers to the number of changes in BGP connections: ASs harboring malicious behavior have a greater number of connectivity changes than ASs not involved in malicious activities, and these changes involve more of their peers.

Considering specifically the distributed intrusion detection system (DIDS) environment, trusting warning messages according to their source's reputation or skill is a critical security point to prevent internal attacks. The intrusion detection network proposed in [15] infers the trustworthiness of each distributed peer based on its performance to solve internal puzzles. The more successful a node is in solving security puzzles, the more reliable it is considered to the rest of the intrusion network. In the same sense, the more reliable a node is according to its network's point of view, the higher priority it has to challenge others. In our previous work [2], each federated IDS traversed by a suspicious flow that detects it as an intrusion uses the BGP-FlowSpec framework to cooperate with the distributed detection platform by announcing a possible ongoing attack. For a destination target that receives these BGP-based alarms from a distant AS, knowing how much it can trust this information before making security decisions is imperative. In this case, the consensus-based approach of the distributed system imposes a message-by-message analysis, instead of extracting volumetric attributes from the row data of BGP update messages. The main contribution of this paper is to show that it is possible to infer the confidence level of each BGP update message individually, based solely on its mandatory header information.

## III. Data Set Description

Building a labeled dataset demands either customizing an open Internet-available data set related to the objectives at hand or using a specific data set reproducing the same scenario [16]. Thinking on that, we propose building a data set containing strategic features, extracted individually from the path attributes of each BGP update message. This data set is used to teach a regression-based machine learning model to infer the confidence level of each BGP update message ($C_{L_i}$) based on its mandatory header information. Combining the positive-prediction value $\text{PPV}_i$ that measures the precision of the $\text{IDS}_i$, with $C_{L_i}$ evaluated from the $\text{BGP}_i$ header, we consolidate the confidence mass ($M_{C_i}$) of the intrusion evidence

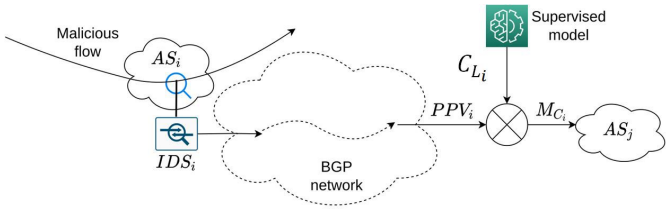



Figure 1: Process of consolidating the confidence mass $M_{C_i}$ of the intrusion evidence, by combining $\text{PPV}_i$ and the confidence level $C_{L_i}$ evaluated by the machine learning model.

$i$ [2]. In other words, besides solely using the fickle positive-prediction value of each federated IDS member ($\text{PPV}_i$), the idea is to combine the two data inputs, as depicted in Figure 1.

To the best of our knowledge, we still do not have any DIDS data set based on FlowSpec messages. Therefore, we consider a global worm attack named Code Red II occurred in July $19^{th}$ 2001 between 10am and 8pm GMT, as our reference. In that time, Code Red II imposed a worldwide impact on the BGP network, triggering message spikes on the RIPE NCC routing collector RRC04 coming from ASs 513, 559, and 6893 peers during the active attack interval [10]. In order to have a comprehensive view regarding the attack occurrence, we collected raw BGP data in three different time intervals: before the attack (2001-07-12), during the attack, and after the attack (2001-07-26).

### A. Direct Features

The direct feature is the data set attribute extracted from the input data and used in the machine learning model as it is.

*1) Origin:* It is directly extracted from the $ORIGIN$ code, which is a mandatory attribute whose values define the origin of the Network Layer Reachability Information field (NLRI) according to its learning process. It can take three different values: $i$ (IGP), $e$ (EGP), or ? (incomplete). Normally, the $ORIGIN$ code plays a secondary role in the BGP route selection algorithm as the fourth decision criterion. However, according to the analysis presented in [17], there are some vulnerabilities related to bogus $ORIGIN$ code.

*2) ASN Repetitions:* Reflects the number of ASN repetitions in the AS_PATH. It is generally related to the AS prepending (ASPP) mechanism to manipulate the route choice for the AS destination. The ASPP is largely used as a traffic engineering tool to control the usage of input links by adding the local Autonomous System Number (ASN) multiple times in the AS-PATH, making it longer and thus less likely to be chosen by other ASs. Although ASPP is beneficial for traffic engineering, it can compromise the security of Internet routing. More precisely, ASPP use can increase the risk of prefix interception attacks or trigger DoS attacks.

*3) AS_PATH Length:* Refers to the number of non-repeating ASNs in the $AS\_PATH\_LENGTH$ field. Recent works as in [18] shown that the most traffic on the Internet crosses up to 5 ASs before arriving at its destination AS. Thus, announcements from distant ASs are less common and, therefore, less reliable.

### B. Indirect Features

Indirect features are the ones obtained from the information present in the input data, requiring some further processing to become data set attributes.

*1) Betweenness of the Originator-AS:* The betweenness or customer cone size of a certain $\text{AS}_i$, named $\text{Bet}_i$, measures the number of prefixes and other ASs that can be directly or indirectly reached through this $\text{AS}_i$. The Center for Applied Internet Data Analysis (CAIDA) offers for free the AS RANK page to the Internet community, classifying all the ASNs according to their betweenness. Regarding the DIDS scenario described before, an alarm message from a highly classified AS tends to be more reliable.

*2) Security Reputation:* The reputation of an $\text{AS}_i$, $\text{REP}_i$, refers to how reliable $\text{AS}_i$ looks for the other ASs. BGP Ranking assesses the security level of each Internet AS based on the number of blacklisted prefixes it has in any of the 15 blacklist platforms it considers as input data. The higher the BGP Ranking level, the more reliable the AS is.

*3) Mean Betweenness:* The mean betweenness is evaluated by averaging the betweenness value of each AS in the $AS\_PATH$, as shown in Equation 1, in which $n$ is the number of non-repeated ASs in the $AS\_PATH$.

$$\overline{Bet}_{AS\_PATH} = \frac{1}{n}\sum_{i=1}^{n} Bet_i \tag{1}$$

*4) Mean Security Reputation:* Likewise, mean security reputation is evaluated by averaging the security reputation of each AS in the $AS\_PATH$, as shown in Equation 2, in which $n$ is the number of non-repeated ASs in the $AS\_PATH$.

$$\overline{REP}_{AS\_PATH} = \frac{1}{n}\sum_{i=1}^{n} REP_i \tag{2}$$

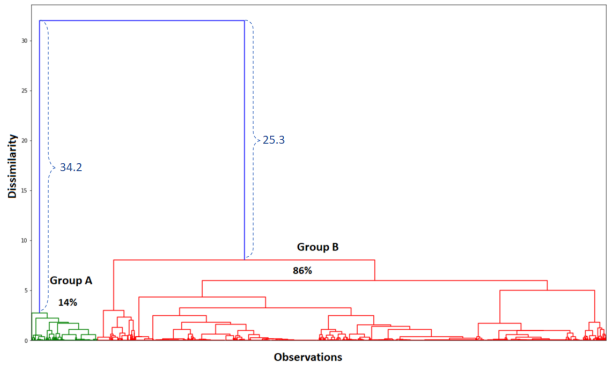
*5) AS Peer Betweenness:* AS peer is the last ASN of the $AS\_PATH$, from which the BGP update message arrives at the target AS. In general, peer agreements are celebrated involving both reciprocal trust and business criteria, so it is not expected to receive malicious messages from AS peers.

*6) AS Peer Security Reputation:* The security reputation of an AS peer candidate is usually mutually assessed before the peering agreement. However, as it can change over time, it should be continually monitored by the AS administrator.

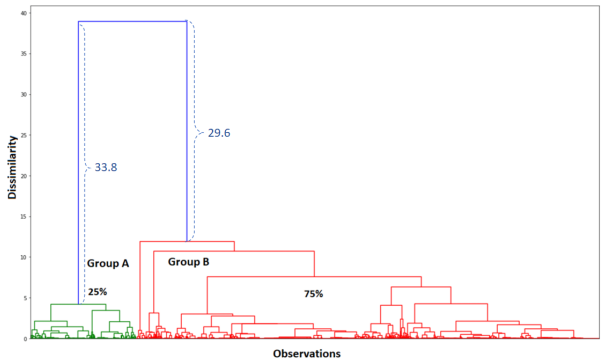
*7) Maximum Betweenness in the AS_PATH:* This feature is obtained by evaluating the betweenness of each AS in the $AS\_PATH$ list and selecting the highest one.

*8) Minimum Betweenness in the AS_PATH:* This feature is obtained by evaluating the betweenness of each AS in the $AS\_PATH$ list and selecting the lowest one.

*9) Median Betweenness in the AS_PATH:* Refers to the middle betweenness value, considering all the ASs in the $AS\_PATH$. This kind of feature is usually adopted in machine learning models to workaround distortions related to heavy tails in probability distributions.




(a) Dendrogram graph considering BGP messages *outside* the attack interval, $Diss_{AB} = 25.3 + 34.2 = 59.5$




(b) Dendrogram graph considering BGP messages *inside* the attack interval, $Diss_{AB} = 29.6 + 33.8 = 63.4$

Figure 2: Dendrogram graphs obtained from the data set proposed in Section III.

*10) Maximum Security Reputation in the AS_PATH:* This feature is obtained by evaluating the security reputation of each AS in the $AS\_PATH$ list and selecting the highest one.

*11) Minimum Security Reputation in the AS_PATH:* This feature is obtained by evaluating the security reputation of each AS in the $AS\_PATH$ list and selecting the lowest one.

*12) Median Security Reputation in the AS_PATH:* Refers to the middle-security reputation value, considering all the ASs in the $AS\_PATH$. This kind of feature is usually adopted in machine learning models to workaround distortions related to heavy tails in probability distributions.

## IV. Machine Learning Model

Firstly, it is worth stating that different from choosing the best machine learning model, our goal consists in proving it is possible to infer the confidence level of each BGP update message individually, based solely on its mandatory header information.

Machine learning is an application of artificial intelligence (AI) that provides systems with the ability to automatically learn and improve from previous data without being explicitly programmed for the task at hand. Besides preparing data, some relevant factors come into play when choosing a machine learning algorithm, such as the level of accuracy needed, the time required to train the model, the number of features in your data set, the linearity of your data, and, finally, whether you need to combine more than one Algorithm (Ensemble methods). As stated in Section II, the main objective of this paper is to prove that it is possible to train a machine learning model to infer the confidence level of each single BGP update announcement by using only its mandatory header information. In order to assess consistently the usability of our data set, we performed non-supervised and supervised tests.

### A. Non-supervised Tests

Although they are far from limited to this, non-supervised models (NS) are commonly used before running a supervised model as a support to label its input data. It works by separating an unlabeled data set into a finite and discrete set of data clusters. There are many methods to implement data clustering in NS models. In this case, we choose to combine $K$-means and hierarchical clustering (HC), which are by far the most common algorithms used in non-supervised models [19].

*1) Hierarchical model:* The hierarchical clustering algorithms (HC) organize data according to the proximity matrix, eliminating previous definitions of parameters. The results of HC are usually depicted by a binary tree or dendrogram. The root node of the dendrogram represents the whole data set of observations, and each leaf node is regarded as a data object. The intermediate nodes describe the extent that the objects are close to each other, in which the height of the dendrogram expresses the dissimilarity ($Diss$) between each pair of clusters. The ultimate clustering results can be obtained by cutting the dendrogram at different levels. Figure 2 shows both dendrogram graphs calculated for the data set messages outside (Figure 2a) and within (Figure 2b) the attack interval.

Analyzing the dendrograms shown in Figures 2a and 2b it is possible to evaluate the dissimilarity ($Diss_{AB}$) between groups A (green) and B (red) in the two scenarios depicted in Figures 2a and 2b.

Comparing the dissimilarities evaluated in Figures 2b and 2a, demonstrates that the two groups, A (green) and B (red), become better characterized as different clusters during the attack. In addition, comparing the number of observations in each group, the relative size of group B regarding group A within the attack interval in Figure 2b is larger ($86\%$) than in Figure 2a ($75\%$), outside the attack interval, reinforcing the hypothesis that it tends to contain the most malicious or attack-related BGP messages.

### B. Supervised Tests

The non-supervised tests described in Section IV-A confirmed the hypothesis that our unlabeled data set can be divided into two different and well-defined clusters, one of them related to the attack event. However, our main goal to precisely predict the confidence level of each BGP update message requires us to go further toward using a supervised

learning model. Supervised models rely on learning algorithms to approximate a mapping function from the input to the output by training it with a previously labeled data set, as a teacher supervises a student's learning process. Therefore, besides preparing a data set representing the target scenario, whose process is described in Section III, we also need to label the data set observations, according to their potential relation to a malicious attempt.

As mentioned in Section III, our data set was built from the public data set containing BGP data from ASs 513, 559, and 6893 collected by RIPE RRC04, detailed in [20] Taking the Code Red day in July $19^{th}$ 2001 as our reference, we extracted direct and indirect attributes from the header of each BGP update announce, according to Sections III-A and III-B.

The data set resulting from the combination described previously was divided into three different partitions, keeping the attack causality in the timeline:

- Training – The training set is a portion of a data set used to fit (train) a model for predicting values that are known in the training set, but unknown in other (future) data.
- Validation – The validation partition is used to assess the performance of the learning model that has been fit on a separate portion of the same data set (the training set). Typically, a validation set provides a useful guide to selecting the best-performing model.
- Test – The test partition is a portion of a data set used to assess the likely future performance of the learning model that has been selected from among competing models, based on its performance with the validation set.

To label our training partition, we added a new feature named *ATTACK*, linking each observation to the Code Red II attack mentioned before. The *ATTACK* feature was populated by matching each sample observation in the before-mentioned training partition with the data set proposed in [10], which is 98% accurate, according to the results presented in [21]. $ATTACK = 0$, it indicates the observation is not related to the Code Red II attack. Otherwise, $ATTACK = 1$ indicates the observation at hand matches.

In the validation partition, we also added a new column named *Confidence_Level* ($0 \leq C_L \geq 1$) to indicate the confidence level of each observation, considering the reputation of its origin AS, among others. The ML model then populates this new column based on the learning obtained in the training phase. After that, the *ATTACK* label is settled according to Equation 3.

$$ATTACK = \begin{cases} 0 \text{ , for } C_L > 0.5 \\ 1 \text{ , for } C_L \leq 0.5 \end{cases} \quad (3)$$

The validation phase also refers to choosing the best machine learning model in terms of performance, comparing the *ATTACK* field is settled by using Equation 3 with the *ATTACK* label, obtained from the work in [10]. The validation algorithm uses TensorFlow to automatically test different model setups, changing the number of neurons, dropout, learning rate, activation, and loss functions. After that, the validation algorithm chooses the model presenting the best performance regarding metrics obtained based on each confusion matrix derived from the tests. The best model, selected after the validation tests, has 112 neurons for the input layer, 88 hidden layers, and 128 neurons for the output layer.

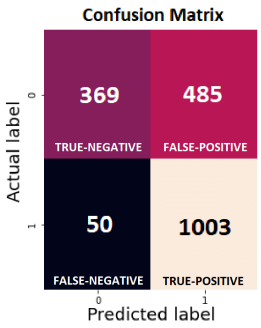



Figure 3: Confusion matrix obtained from new data.

Table I: Performance results obtained from the confusion matrix in Figure 3.

| Metric | Value |
|---|---|
| Loss | 0.56 |
| Accuracy | 0.72 |
| Precision | 0.67 |
| Recall | 0.95 |
| AUC | 0.75 |
| PRC | 0.77 |
| f1-score | 0.79 |

Table II: Comparing f1-score metric with similar works.

| Work | f1-score | Year | #features |
|---|---|---|---|
| SVM-based [22] | 0.96 | 2019 | 37 |
| SVM-LSTM [23] | 0.72 | 2016 | 37 |
| Graph [24] | 0.93 | 2021 | 17 |
| Multi-view [25] | 0.96 | 2021 | 46 |
| This work | 0.79 | 2022 | 15 |

The main goal of the test phase is to evaluate the overall performance of the learning model chosen in the validation phase, using new data. It was accomplished by (*i*) generating the confusion matrix from the test partition shown in Figure 3, and by (*ii*) calculating the performance metrics from its numbers, presented in Table I.

The performance metrics in Table I reveal a high recall, which indicates the model performs well in classifying potentially malicious messages. However, the model lacks the precision to classify the set of malicious messages that are truly related to an attack, which affects the f1-score metric, shown in Table II.

The *f1-score* metric is one of the most well-known statistical measures to compare ML models' performance. It can be defined as the harmonic mean between precision and recall. As can be observed in Table II, although our model presents a high recall value, the far-from-sensational precision of our model takes its *f1-score* down.

Figure 4 plots in the same picture the confidence level ($M_C$) of all the BGP update messages in the test partition with their respective *ATTACK* labels from the matching process with the data set proposed in [10]. Although most of the orange points – meaning the BGP message is potentially related to an attack – are concentrated on the bottom part of the Figure 4, we also have blue points – meaning the BGP update message is not related to an attack – in the same area, indicating a poor false-positive performance. The best-expected condition is having all the blue points on the top, holding the orange points on the bottom. However, due to the lackluster precision

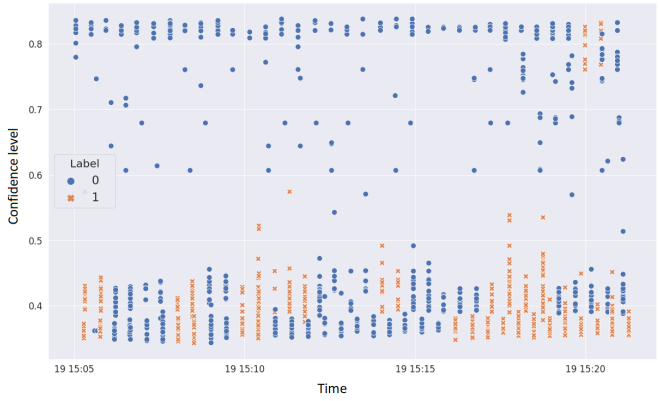



Figure 4: The blue points – indicating no-attack – are expected to be concentrated on the top, while the orange points – indicating attack – should be concentrated on the bottom.

of the model, one can see many blue points on the bottom and a few orange ones on the top.

## V. Conclusion and Future Works

In the widely distributed and cooperative Internet ecosystem, it is not possible to trust the information before spending some effort to check it according to its reputation on the network. In this paper we considered a scenario where distributed IDSs cooperatively share detection information by using the BGP network to propose a machine-learning-based approach that can be seen as an insight for developing systems aiming to prevent the BGP network itself from malicious updates.

To the best of our knowledge, there is not a public dataset based on intrusion detection alarms transported via BGP messages [2]. Thus, we built our own data set from the indirect effects over the BGP network due to a widespread worm attack, already addressed in [10]. Even using a not-directly related data set, performance results obtained from the confusion matrix show that the proposed model performs accurately to evaluate the confidence level of each BGP message. In addition, although the precision still needs to be improved, further performance metrics, namely ROC and PRC, show that the model is able to generalize for new BGP data. Differently from choosing the best machine learning model, we prove it is possible to infer the confidence level of each BGP update message individually, based solely on its mandatory header information. Another conclusion speculates performance tends to be even better, considering using a data set from input data directly related to intrusion detection events.

For future works, we plan to improve the supervised learning model by including weighting for feature selection, aiming to solve the precision problem mentioned in Section IV-B. We are also implementing the DIDS proposal described in [2] based on which we will build a new data set using BGP update messages directly derived from intrusion detection events.

## Acknowledgments

The authors thank FAPERJ — the official funding agency for supporting science and technology research in the state of Rio de Janeiro, Brazil and Rede-Rio — the state academic backbone network — for the support given in the course of this work.

## References

[1] G. Huston, M. Rossi, and G. Armitage, "Securing BGP—A literature survey," *IEEE Comm. Surveys*, vol. 13, no. 2, pp. 199–222, 2010.
[2] R. S. Silva and L. F. de Moraes, "A cooperative approach with improved performance for a global intrusion detection systems for internet service providers," *Annals of Telecom.*, vol. 74, no. 3, pp. 167–173, 2019.
[3] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol, Version 1," RFC 8210, Sep 2017.
[4] T. e. a. Chung, "Rpki is coming of age: A longitudinal study of rpki deployment and invalid route origins," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. ACM, 2019, pp. 406–419.
[5] K. Kirkpatrick, "Fixing the internet," *Commun. ACM*, vol. 64, no. 8, p. 16–17, Jul 2021.
[6] C. L. et al., "Dissemination of flow specification rules," RFC 8955, Dec 2020.
[7] O. Nordström and C. Dovrolis, "Beware of bgp attacks," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 1–8, 2004.
[8] O. Nordström and C. Dovrolis, "Beware of bgp attacks," *Computer Communication Review*, vol. 34, pp. 1–8, Apr 2004.
[9] 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, 2017.
[10] P. Fonseca, E. S. Mota, R. Bennesby, and A. Passito, "Bgp dataset generation and feature extraction for anomaly detection," in *2019 IEEE ISCC*, 2019, pp. 1–6.
[11] I. O. de Urbina Cazenave, E. Köşlük, and M. C. Ganiz, "An anomaly detection framework for bgp," in *2011 International Symposium on Innovations in Intelligent Systems and Applications*, 2011, pp. 107–111.
[12] K. McGlynn, H. B. Acharya, and M. Kwon, "Detecting bgp route anomalies with deep learning," in *IEEE INFOCOM Workshops*, 2019, pp. 1039–1040.
[13] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur, "Topology-based detection of anomalous bgp messages," vol. 2820, 08 2003.
[14] C. A. Shue, A. J. Kalafut, and M. Gupta, "Abnormally malicious autonomous systems and their internet concectivity," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 220–230, 2012.
[15] C. J. Fung and R. Boutaba, *Intrusion Detection Networks - A Key to Collaborative Security*. CRC Press, 2013.
[16] Y. Roh, G. Heo, and S. E. Whang, "A survey on data collection for machine learning: a big data-ai integration perspective," *IEEE Trans. on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1328–1347, 2019.
[17] S. L. Murphy, "BGP Security Vulnerabilities Analysis," RFC 4272, Jan 2006.
[18] C. Wang, Z. Li, X. Huang, and P. Zhang, "Inferring the average as path length of the internet," in *2016 IEEE IC-NIDC*, Sep 2016, pp. 391–395.
[19] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.
[20] ©RIPE NCC, "Ris raw data," Acessed on http://data.ris.ripe.net/rrc04/, Jul 2021, (Accessed at: Sep 13th 2021, 16:17:11.).
[21] P. C. da Rocha Fonseca, "A deep learning framework for bgp anomaly detection and classification," PhD thesis, Federal University of Amazonas, Manaus, Amazonas, Brazil, March 2020.
[22] X. Dai, N. Wang, and W. Wang, "Application of machine learning in BGP anomaly detection," *Journal of Physics: Conference Series*, vol. 1176, p. 032015, Mar 2019.
[23] Q. D. et al., "Detecting BGP anomalies using machine learning techniques," in *2016 IEEE SMC*, 2016, pp. 003 352–003 355.
[24] T. B. e. a. Paiva, "BGP Anomalies Classification using Features based on AS Relationship Graphs," in *2021 IEEE LATINCOM*. IEEE, 2021, pp. 1–6.
[25] S. P. et al., "A multi-view framework for BGP anomaly detection via graph attention network," *CoRR*, vol. abs/2112.12793, 2021.