Annals of Telecommunications
https://doi.org/10.1007/s12243-024-01045-1

Inferring the confidence level of BGP-based distributed intrusion detection systems alarms

Renato S. Silva[1] · **Felipe M. F. de Assis**[1] · **Evandro L. C. Macedo**[1] · **Luís Felipe M. de Moraes**[1]

Received: 8 January 2024 / Accepted: 25 May 2024

Abstract
Border Gateway Protocol (BGP) is increasingly becoming a multipurpose protocol. However, it keeps suffering from security issues such as 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 BGP messages, using just the mandatory fields of the header. Tests using two different data sets, (*i*) from the indirect effects of a widespread worm attack and (*ii*) using up-to-date data from the IPTraf Project, show promising results, considering well-known performance metrics, such as recall, accuracy, receiver operating characteristics (ROC), and f1-score.

Keywords DIDS · Machine learning · BGP · Distributed intrusion detection system

1 Introduction

The Border Gateway Protocol (BGP) turned 32 years old in 2023, making it one of the longest-lasting, widely used protocols ever deployed on the Internet. If we consider BGP was initially conceived by Yakov Rekhter (IBM) and Kirk Lougheed (Cisco) on two napkins during their lunchtime, as described in [1], it is no doubt one of the most impressive success stories of the Internet. The essential function of BGP is to control how IP packets are routed across the Internet through exchanging routing and reachability information between edge routers. As a whole, BGP plays the role of directing traffic between autonomous systems (AS), which are networks managed by a single entity on the backbone of the Internet. When an AS gets set up, its administrator configures a peer with another AS manually to share their IP prefixes, which are then propagated to other AS, and so on. Many researchers argue that the two main reasons for BGP's fast widespread establishment as the Internet routing protocol are its simplicity and its ability to combine technical and business criteria to set up AS neighborhoods.

The tremendous success 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 [2] and new features such as BGP-FlowSpec [3]. However, despite the several improvements it has had since its worldwide implementation, BGP still keeps being vulnerable to both malicious attacks and human errors [4]. For example, there are roughly 73,000 ASs that make up the global Internet and little information on how each AS peering filter must be configured. This means that whenever a new bogus route (also known as a bogus prefix) is announced (either through intentional hijacking or just a typo) by a malicious originator,

Felipe M. F. de Assis, Evandro L. C. Macedo, and Luís Felipe M. de Moraes contributed equally to this work.

✉ Renato S. Silva
renato@ravel.ufrj.br

Felipe M. F. de Assis
assis@ravel.ufrj.br

Evandro L. C. Macedo
evandro@ravel.ufrj.br

Luís Felipe M. de Moraes
moraes@ravel.ufrj.br

[1] High-Speed Networks Laboratory PESC/COPPE, Universidade Federal do Rio de Janeiro, Avenida Horacio Macedo, 2030 Cidade Universitaria, Rio de Janeiro 21941-914, RJ, Brazil

Published online: 18 June 2024

neighboring ASs send traffic to the wrong network, and this can spread across the Internet. Besides its importance for protecting prefixes' reachability, assessing the confidence level of these messages also helps to improve self-defense mechanisms of distributed intrusion detection systems (DIDS) that use BGP as their underlying network, as proposed in [5].

Improving the BGP security remains a hot topic in academia, inspiring several works on that for a long time. Although there is a myriad of approaches addressing security issues on the BGP protocol, it is possible to recognize the approach based on Resource Public Key Infrastructure (RPKI) [6] as the main proposal, instead of solely relying on Internet Routing Registries (IRRs) databases [7]. While filtering rules are both labor and time-intensive due to the need to constantly maintain and update records, the distributed public database of RPKI, which is composed of cryptographically signed records concerning routing information supplied by networks, is considered to be a highly secure and reliable mechanism, but one cannot guarantee its accuracy [8]. 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.

Nonetheless, although recognizing RPKI as a consolidated approach for validating BGP routing information, it still needs a third-party certification entity. In addition, implementing RPKI on the entire Internet is far from being a simple task. According to the analysis proposed in [9], while large ASs (*e.g.*, such as Google, AT&T, NTT, and Cogent) are already announcing to be performing the origin validation, small ASs are not considering it 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.

As a subdivision of Artificial intelligence (AI), machine learning (ML) is an important tool to support decision-making [10]. Using massive data sets as input, a ML model is able to discover patterns and deviations from expected behavior. Indeed, machine learning models have been intensively used to detect BGP anomalies, including those related to widespread worm attacks, *e.g.*, such as Code Red II [11], which occurred in July 19^{th}, 2001 and Slammer [12], that took place in January 25^{th}, 2003. However, most models proposed with this goal utilize volume or statistic-based attributes that require a large amount of data for each distinctive feature, which compromises the one-by-one processing nature of some distributed intrusion detection systems.

This paper is an extension of our former paper [13], which proposes an out-of-band approach composed of a 15-attribute data set and a machine learning model, able to infer the confidence level of each BGP update message that arrives at a given AS. The data sets used in this paper combine direct and indirect attributes obtained from individual instances of these messages. That is, in the same way as each BGP update announcement is processed by a router to update its reachability table, the confidence level is evaluated message by message, using only the mandatory fields of the BGP header, thus dispensing volume or statistic-based attributes. This is especially important due to the one-by-one nature of the DIDS alarms from the federated IDSs to be combined at the destination AS. In this case, knowing the confidence level of each BGP message helps to prevent fake detection alarms, aiming to compromise the overall detection performance of the DIDS. Results obtained considering 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 aforementioned extension includes a summarized analysis of a new up-to-date dataset, by matching the BGP row data collected from the Rede-Rio Project [14] network with the anomaly reports of the IPTraf Project [15].

The remainder of this paper is organized as follows. In Sect. 2, we shortlist the main works related to improving the BGP security. We also emphasize the contributions of this paper by comparing it with the existing approaches. Section 3 explains the process adopted to build the data set used to train the machine learning model. Section 4 describes the unsupervised and supervised tests using the data set built in Sect. 3 and presents some relevant performance results. Section 5 presents the data set analysis performed over the real data obtained from IPTraf Project. Finally, in Sect. 6, we close the article with an objective analysis correlating the results obtained from the models with the paper's contributions.

2 Related work

Network topology changes provoked by the effects of some kind of attack have been very well studied in academia [16]. The survey proposed in [17] presents a comprehensive approach regarding BGP anomalies, including a canonical taxonomy classifying them according to their intentionality and causality. The study in [17] 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 did not intend to directly compromise the BGP network, their effects certainly did. Taking advantage of this unnatural behavior, several works have proposed BGP-labeled data sets to train machine learning models aiming to detect—and sometimes classify—attacks.

The labeled data set proposed in [18] has 35 features distributed as direct, indirect, volume, and statistical. To label their data set, P. Fonseca et al. [18] 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 the Ripe Project. Performance tests using new data show promising results in detecting and classifying the anomalies between attacks and events. In the same track, the approach proposed in [19] 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 [20] 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 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 [21] 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 [22] 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 [23] infers the trustworthiness of each distributed peer based on its performance in solving internal puzzles. The more successful a node performs in solving security puzzles, the more reliable it is 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 the priority it has to challenge others. In our previous work [5], each federated IDS traversed by a suspicious flow that detects it as an intrusion uses the BGP-FlowSpec protocol 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.

Created on May 22, 1992, Rede-Rio [14] carries out activities related to science, technology, and education in the State of Rio de Janeiro - Brazil. Rede-Rio interconnects several government institutions in Brazil with the Internet, permitting them to interchange knowledge for the common good of the Brazilian people. Connected to Rede-Rio, IPTraf [15] is part of its security solutions. IPTraf collects and processes flow data from Rede-Rio routers, aiming to detect traffic anomalies. The flow data processed by IPTraf gives rise to several traffic dashboards that permit the network administrators to check traffic online. IPTraf also submits the flow information into a detection systems chain that emits security alarms in case of detecting traffic anomalies. Both Rede-Rio and IPTraf are part of this paper by enabling us to work with up-to-date data to build the extent dataset to train our machine learning model.

3 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 [24]. 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 PPV_i that measures the precision of the IDS_i, with C_{L_i} evaluated from the BGP_i header, we consolidate the confidence mass (M_{C_i}) of the intrusion evidence i [5]. In other words, besides solely using the fickle positive-prediction value of each federated IDS member (PPV_i), the idea is to combine the two data inputs, as depicted in Fig. 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 [18]. 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).

3.1 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.

- **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 [25], there are some vulnerabilities related to bogus $ORIGIN$ code.
- **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.
- **AS_PATH Length:**
  Refers to the number of non-repeating ASNs in the AS_PATH_LENGTH field. Recent works as in [26] show 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.

3.2 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.

- **Betweenness of the originator-AS:** The betweenness or customer cone size of a certain AS_i, named Bet_i, measures the number of prefixes and other ASs that can be

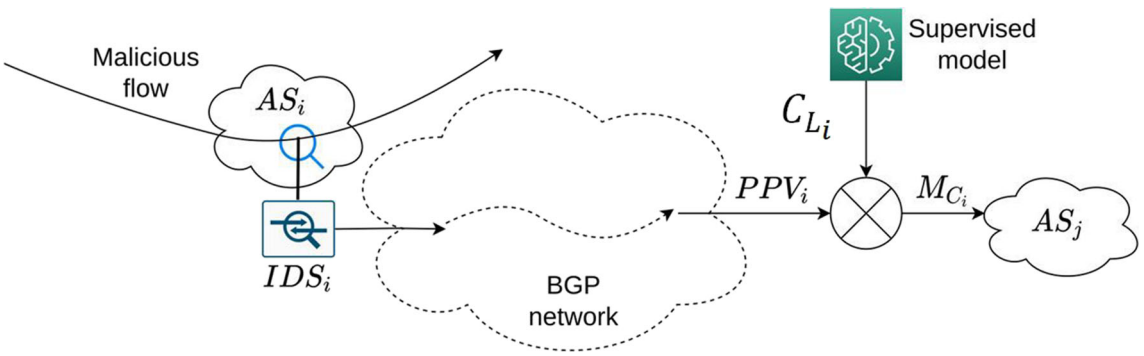


Fig. 1 Process of consolidating the confidence mass M_{C_i} of the intrusion evidence, by combining PPV_i and the confidence level C_{L_i} evaluated by the machine learning model

directly or indirectly reached through this 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.

- **Security reputation**:
  The reputation of an AS_i, REP_i, refers to how reliable 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.
- **Mean betweenness:**
  The mean betweenness is evaluated by averaging the betweenness value of each AS in the AS_PATH, as shown in Eq. 1, in which n is the number of non-repeated ASs in the AS_PATH.

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

- **Mean security reputation:**
  Likewise, mean security reputation is evaluated by averaging the security reputation of each AS in the AS_PATH, as shown in Eq. 2, in which n is the number of non-repeated ASs in the AS_PATH.

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

- **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.
- **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.
- **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.
- **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.
- **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.
- **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.
- **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.
- **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.

4 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 Sect. 2, 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.

4.1 Non-supervised tests

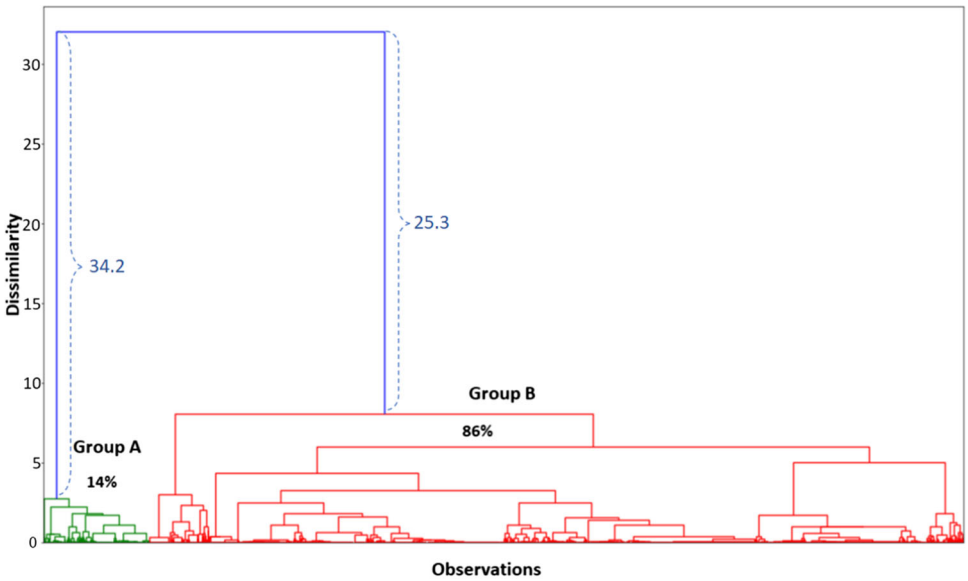
Although they are far from limited to this, non-supervised models (NS) are commonly used before running a supervised model as a support to label its input data. It works by separating an unlabeled data set into a finite and discrete set of data clusters. There are many methods to implement data clustering 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 [27].

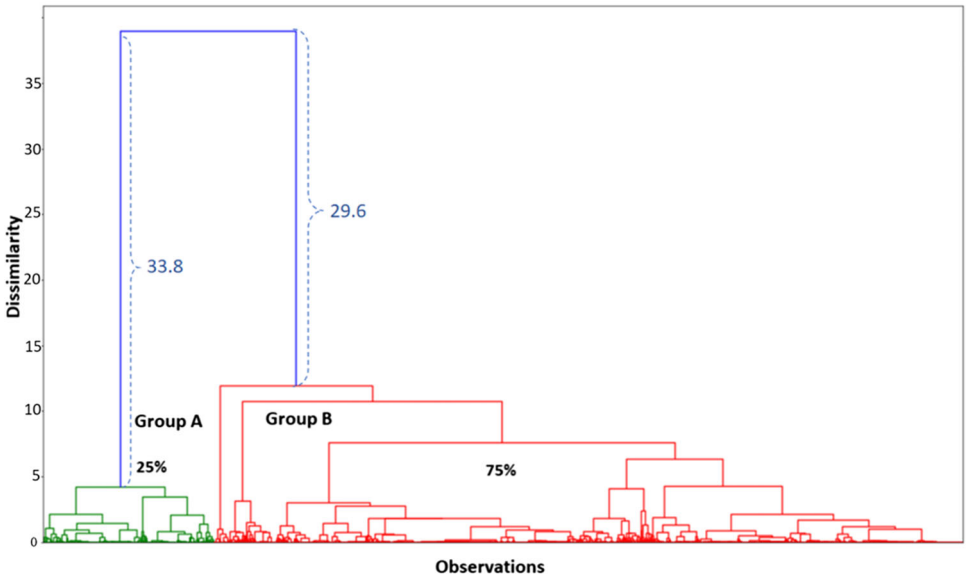
4.1.1 Hierarchical model

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 to which 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 (Fig. 2a) and within (Fig. 2b) the attack interval.

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



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



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

Fig. 2 Dendrogram graphs obtained from the data set proposed in Sect. 3

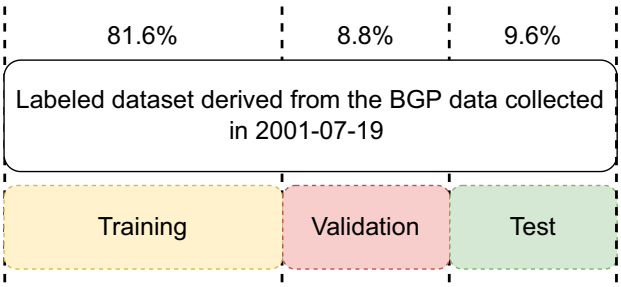


Fig. 3 Splitting strategy adopted in this work to train the algorithms, validate the automatic adjusting, and evaluate its performance

Comparing the dissimilarities evaluated in Fig. 2b and a 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 A regarding group B within the attack interval in Fig. 2b is larger (25%) than in Fig. 2a (14%), outside the attack interval, reinforcing the hypothesis that it tends to contain the most malicious or attack-related BGP messages.

4.2 Supervised tests

The non-supervised tests described in Sect. 4.1 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 Sect. 3, we also need to label the data set observations, according to their potential relation to a malicious attempt.

Table 1 Comparing f1-score metric with similar works

Work	f1-score	Year	#features
SVM-based [30]	0.96	2019	37
SVM-LSTM [31]	0.72	2016	37
Graph [32]	0.93	2021	17
Multi-view [33]	0.96	2021	46
This work	0.79	2022	15

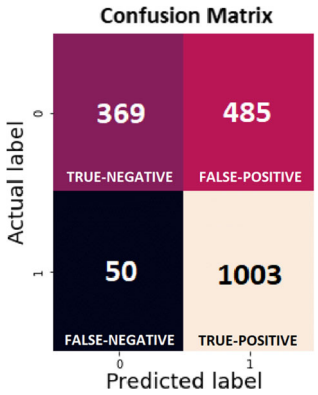
As mentioned in Sect. 3, our data set was built from the public data set containing BGP data from ASs 513, 559, and 6893 collected by RIPE RRC04, detailed in [28]. Taking the Code Red day in July 19^{th} 2001 as our reference, we extracted direct and indirect attributes from the header of each BGP update announcement, according to Sects. 3.1 and 3.2.

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 among competing models, based on its performance with the validation set.

Figure 3 provides a graphical idea about the strategy adopted in this paper to split our dataset.

Fig. 4 Confusion matrix and the respective values of performance



(a) Confusion matrix obtained from new data.

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

(b) Performance results obtained from the confusion matrix in Figure 4a.

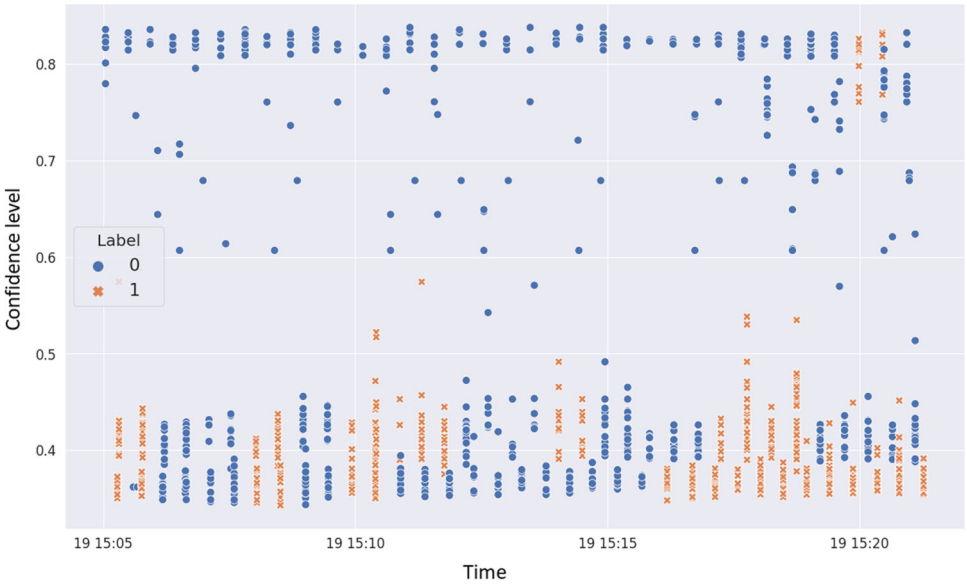


Fig. 5 The blue points—indicating no attack—are expected to be concentrated on the top, while the orange points—indicating attack—should be concentrated on the bottom

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 [18], which is 98% accurate, according to the results presented in [29]. $ATTACK = 0$ indicates the observation is unrelated 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* ($0 \leq C_L \leq 1$) 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 Eq. 3:

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

The validation phase also refers to choosing the best machine learning model in terms of performance, comparing the *ATTACK* field is settled by using Eq. 3 with the *ATTACK* label, obtained from the work in [18]. 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 in the first hidden layer after the input layer, 88 hidden layers, and 128 neurons for the last hidden layer, before the output layer.

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

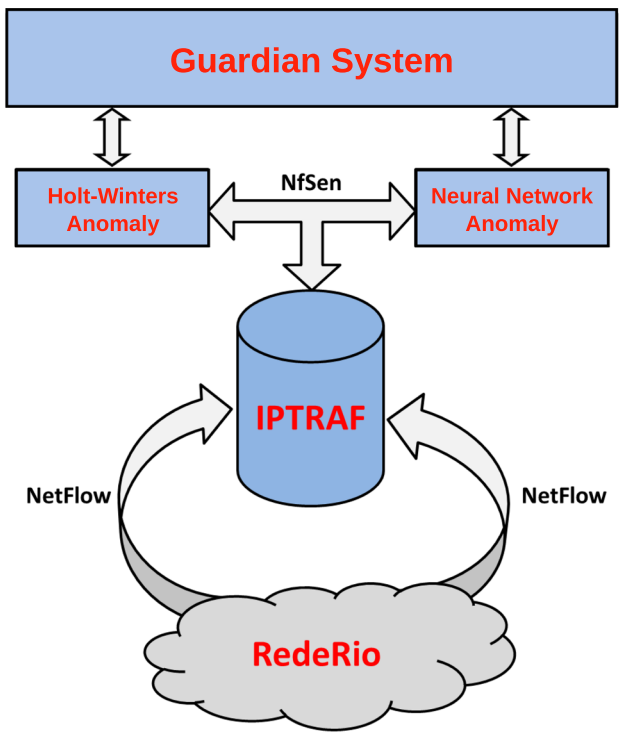


Fig. 6 IPTraf architecture as part of the security solutions of the RedeRio Project

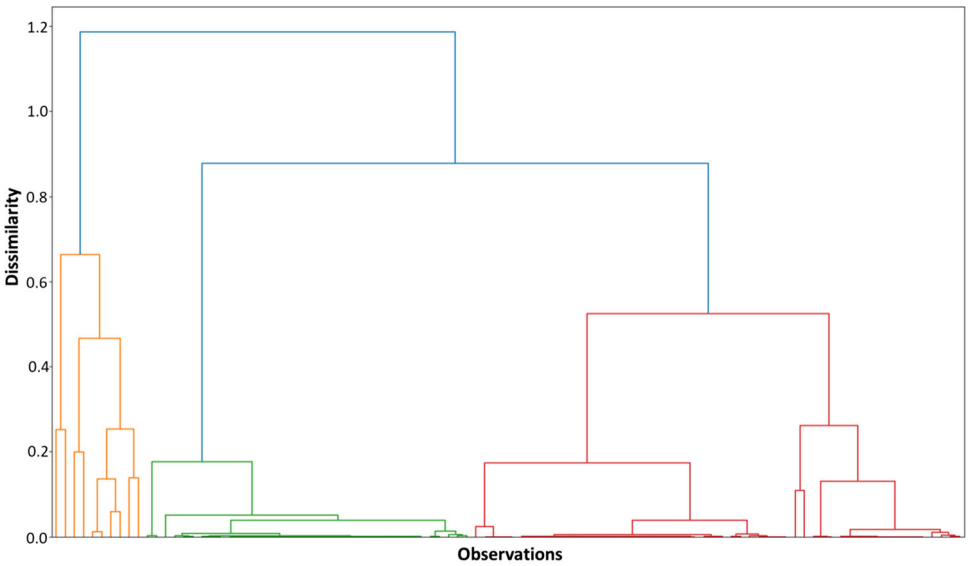
Fig. 4a and by (*ii*) calculating the performance metrics from its numbers, presented in Fig. 4b.

The performance metrics in Fig. 4b 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 1.

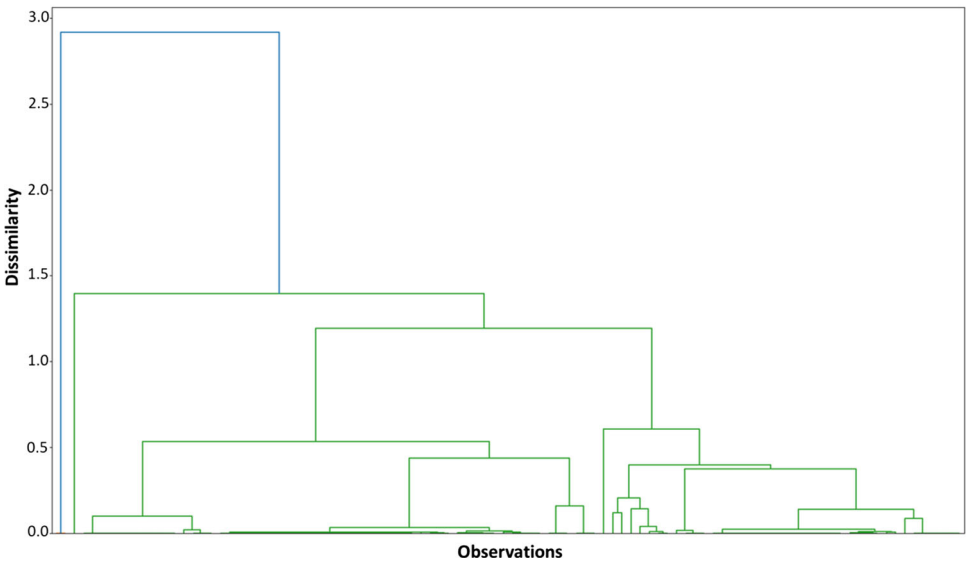
The *f1-score* metric is one of the most well-known statistical measures to compare ML models' performance. It can be defined as the harmonic mean between precision and recall. Therefore, in the case of imbalanced datasets, f1-score can be considered as a reliable metric.

As can be observed in Table 1, although our model presents a high recall value, the far-from-sensational precision of our model takes its *f1-score* down.

Figure 5 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 [18]. Although most of the orange points—meaning the BGP message is potentially related to an attack—are concentrated on the bottom part of Fig. 5, we also have blue points—meaning the BGP update message is not related to an attack—in the same area, indicating poor false-positive performance. The best-expected condition is having all the blue points on the top, holding the



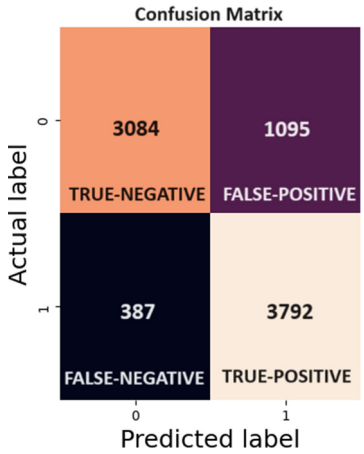
(a) Dendrogram graph considering BGP messages *outside* the attack interval.



(b) Dendrogram graph considering BGP messages *within* the attack interval.

Fig. 7 Dendrogram graphs obtained from the IPTraf data set proposed in Sect. 5

Fig. 8 Confusion matrix and the respective values of performance



(a) Confusion matrix obtained from new labeled data.

Metric	Value
Loss	0.43
Accuracy	0.82
Precision	0.77
Recall	0.91
AUC	0.85
PRC	0.79
f1-score	0.83

(b) Performance results obtained from the confusion matrix in Figure 8a.

orange points on the bottom. However, due to the lackluster precision of the model, one can see many blue points on the bottom and a few orange ones on the top.

5 IPTraf dataset analysis

As part of the security solutions of Rede-Rio, IPTraf plays the critical role of providing network administrators with online dashboards containing detailed traffic information. IPTraf collects flow information from the Rede-Rio routers using Netflow protocol [34]. Once stored in the IPTraf database, flow files are organized and processed in the Nfsen [35] to generate the online traffic dashboards. From Nfsen data, it is possible to extract statistical information that is simultaneously processed by two different anomaly detector systems, as shown in Fig. 6. The Guardian system is in charge of alarming any anomaly detected to the Rede-Rio's network operations center.

To build the new data set, we reproduced the same methodology used in Sect. 3, this time, using fresh BGP raw data, collected from the Rede-Rio routers. To label this new data set, we used the two anomaly detection systems that compose the IPTraf architecture, shown in Fig. 6. In other words, for each anomaly detected by both anomaly detectors, we link to the related BGP update data collected from the Rede-Rio BGP network, making $ATTACK = 1$. For the remaining row data not related to any anomaly detected, we mark $ATTACK = 0$. The matching process between the BGP data with the anomaly alarm from IPTraf was accomplished by using key fields, such as *timestamp* and *origin AS*. We made the dataset available through the following link (https://www.ravel.ufrj.br/files/papers/CSNet2023dataset.zip) or by mailing the authors.

5.1 Unsupervised tests

Figure 7 shows the hierarchical model, named dendrogram, of the new data set built using just the fresh BGP data collected from the Rede-Rio routers, without the $ATTACK$ label.

For both dendrogram graphs in Fig. 7, outside and within the attack interval, it is possible to realize three well-defined clusters. However, comparing the dendrogram in Fig. 7a with Fig. 7b, it is possible to find noticeable differences, mainly for the number of groups that changed in Fig. 7b. This finding reveals that the behavior of BGP messages changed in the anomaly interval.

5.2 Supervised tests

Following the same model described in Sect. 4, we submit the validation part of the labeled data sets to an optimization process to find the best model in terms of performance.

The results shown in Fig. 8 demonstrate that the machine learning model keeps performing well in classifying the confidence level of the BGP update messages that are received by the routers in the Rede-Rio. In other words, including using the new data set obtained from fresh BGP row data, it is possible to infer the confidence level of the BGP update messages by using just the mandatory fields of the BGP header.

6 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. This work considers the scenario proposed in [5],

where distributed IDSs cooperatively share detection information by using the BGP network, to address its security vulnerabilities related to internal attacks. This paper can also 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 [5]. Thus, we built our own data set from the indirect effects over the BGP network due to a widespread worm attack, already addressed in [18]. 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, other performance metrics, namely ROC and PRC, show that the model can 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 that 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 Sect. 4.2. We are also implementing the DIDS proposal described in [5], which will enable us to build a new data set using BGP update messages directly derived from intrusion detection events.

Acknowledgements The authors also thank our fellow Vitor Zanotelli; this work would not be possible as it is without your valuable help.

Author Contributions R.S., F.M, and E.L. wrote the main manuscript text, experiments, and prepared figures. L.F. provided guidance and support for accomplishing such work. All authors reviewed the manuscript.

Funding This study was partially funded by FAPERJ (grant number 150.134/2010).

Data Availability We made the dataset generated during the current study available through the following link (https://www.ravel.ufrj.br/files/papers/CSNet2023dataset.zip) or by mailing the authors.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Jabloner P (2015) A brief guide to recreational pyromania. Accessed on https://computerhistory.org/blog/the-two-napkin-protocol/
2. Bates TJ, Chandra R, Rekhter Y, Katz D (2007) Multiprotocol extensions for BGP-4. RFC Editor
3. Loibl C, Hares S, Raszuk R, McPherson D, Bacher M (2020) RFC 8955: Dissemination of flow specification rules, no. 8955. RFC Editor, p 36
4. Huston G, Rossi M, Armitage G (2010) Securing BGP—a literature survey. IEEE Comm Surveys 13(2):199–222
5. Silva RS, Moraes LF (2019) A cooperative approach with improved performance for a global intrusion detection systems for internet service providers. Annals of Telecom 74(3):167–173
6. Bush R, Austein R (2017) The resource public key infrastructure (RPKI) to router protocol, Version 1. RFC Editor
7. McPherson D, Amante S, Osterweil E, Blunk L, Mitchell D (2015) Considerations for internet routing registries (IRRs) and routing policy configuration, no. 7682. RFC Editor, p 18
8. Chung T et al. (2019) RPKI is coming of age: a longitudinal study of RPKI deployment and invalid route origins. In: Proceedings of the internet measurement conference - ACM, pp 406–419
9. Kirkpatrick K (2021) Fixing the internet. Commun ACM 64(8):16–17
10. Tsai C-F, Hsu Y-F, Lin C-Y, Lin W-Y (2009) Intrusion detection by machine learning: a review. Expert Syst Appl 36(10):11994–12000
11. Moore D, Shannon C, Claffy K (2002) Code-red: a case study on the spread and victims of an internet worm. In: Proceedings of the 2nd ACM SIGCOMM workshop on internet measurment. Marseille, France, IMW '02. ACM, 12, pp 273–284
12. Chindipha S, Irwin B (2017) An analysis on the re-emergence of SQL Slammer worm using network telescope data
13. Silva RS, Assis FMF, Macedo ELC, Moraes LFM (2023) Inferring the confidence level of BGP-based distributed intrusion detection systems alarms. In: 2023 7th Cyber security in networking conference (CSNet), pp 157–162. https://doi.org/10.1109/CSNet59123.2023.10339702
14. REDERIO (2023) Rede-Rio/FAPERJ. Available at https://rederio.br/. Accessed Jan 2024
15. Assis F, Coutinho M, Filho JS, Macedo E, Moraes L (2021) IPTraf: coleta e Detecção de Anomalias em Fluxos de Rede. In: Anais do XXVI Workshop de Gerência e Operação de Redes e Serviços. SBC, Porto Alegre, RS, Brasil, pp 96–109. https://doi.org/10.5753/wgrs.2021.17188. https://sol.sbc.org.br/index.php/wgrs/article/view/17188
16. Nordström O, Dovrolis C (2004) Beware of BGP attacks. Comput Commun Rev 34:1–8
17. Al-Musawi B, Branch P, Armitage G (2017) BGP anomaly detection techniques: a survey. IEEE Commun Surv Tutor 19(1):377–396
18. Fonseca P, Mota ES, Bennesby R, Passito A (2019) BGP dataset generation and feature extraction for anomaly detection. In: 2019 IEEE ISCC, pp 1–6
19. Urbina Cazenave IO, Köşlük E, Ganiz MC (2011) An anomaly detection framework for BGP. In: 2011 International symposium on innovations in intelligent systems and applications, pp 107–111
20. McGlynn K, Acharya HB, Kwon M (2019) Detecting BGP route anomalies with deep learning. In: IEEE INFOCOM workshops, pp 1039–1040
21. Kruegel C, Mutz D, Robertson W, Valeur F (2003) Topology-based detection of anomalous BGP messages, vol 2820
22. Shue CA, Kalafut AJ, Gupta M (2012) Abnormally malicious autonomous systems and their internet connectivity. IEEE/ACM Trans Networking 20(1):220–230
23. Fung C, Boutaba R (2013) Intrusion detection networks: a key to collaborative security. 11(9780429099922):1–237. https://doi.org/10.1201/b16048
24. Roh Y, Heo G, Whang SE (2019) A survey on data collection for machine learning: a big data-AI integration perspective. IEEE Trans Knowl Data Eng 33(4):1328–1347
25. Murphy SL (2006) BGP security vulnerabilities analysis. RFC Editor

26. Wang C, Li Z, Huang X, Zhang P (2016) Inferring the average as path length of the internet. In: 2016 IEEE IC-NIDC, pp 391–395
27. Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
28. RIPE NCC (2021) RIS raw data. Acessed on http://data.ris.ripe.net/rrc04/. Accessed 13 Sep 2021, 16:17:11
29. Rocha Fonseca PC (2020) A deep learning framework for BGP anomaly detection and classification. Phd thesis, Federal University of Amazonas, Manaus, Amazonas, Brazil
30. Dai X, Wang N, Wang W (2019) Application of machine learning in BGP anomaly detection. J Phys: Conf Ser 1176:032015
31. Ding Q, Li Z, Batta P, Trajković L (2016) Detecting BGP anomalies using machine learning techniques. In: 2016 IEEE SMC, pp 003352–003355
32. Paiva TBea (2021) BGP anomalies classification using features based on AS relationship graphs. In: 2021 IEEE LATINCOM. IEEE, pp 1–6
33. Peng S, Nie J, Shu X, Ruan Z, Wang L, Sheng Y, Xuan Q (2021) A multi-view framework for BGP anomaly detection via graph attention network. CoRR abs/2112.12793
34. Claise B (2004) Cisco systems NetFlow services export version 9. RFC Editor. https://doi.org/10.17487/RFC3954. https://www.rfc-editor.org/info/rfc3954
35. Sourceforge (2023) NfSen - Netflow sensor. Available at https://nfsen.sourceforge.net/. Accessed Jan 2024