Feature analysis and selection for BGP anomaly detection

Rifqy Hakimi*† and Martin J. Reed*
*School of Computer Science and Electronic Engineering, University of Essex, Colchester, United Kingdom
†School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung, Indonesia
*Email: {rh21215, mjreed}@essex.ac.uk

***Abstract*—Automated incidence reporting requires accurate and timely anomaly detection. This paper considers this in the context of the Border Gateway Protocol (BGP). BGP is crucial for Internet routing but is vulnerable to attacks due to a lack of widespread authentication. While rare, BGP disruptions can be highly detrimental to Internet performance and security. Detecting BGP anomalies helps network operators protect their networks and improve Internet reliability. In this work, we investigate and develop anomaly detection for BGP using Machine-Learning. We extract relevant features of BGP control plane messages from RIPE RIS and RouteViews public datasets. After extracting features, we employ various feature selection algorithms to extract the most relevant features and explore balancing the datasets. Finally, we explore detection latency, an important operating parameter for automated anomaly detection. The results show that BGP anomalies can be detected in the order of seven minutes when using real attack data and that the observation point for the BGP data has a significant effect on anomaly detection.**

***Index Terms*—anomaly detection, BGP, feature analysis, machine learning, detection latency**

I. Introduction

Border Gateway Protocol (BGP) is an essential network routing protocol for the Internet. Any failure or malicious modification of BGP can have serious implications for Internet performance and the security of end systems on the Internet. This paper investigates anomaly detection in BGP by using Machine Learning (ML) to detect problems that may be malicious or caused by misconfiguration errors. BGP is the routing protocol that facilitates data packet communication between different Autonomous Systems (ASes) on the Internet, determining the most efficient paths for data to travel. Standardised by the Internet Engineering Task Force (IETF) with RFC 1105 in 1989, BGP has served as the core routing protocol for the global Internet since 1994, connecting over 70,000 ASes globally [1]. The decentralised nature of BGP relies heavily on trust among ASes, which can lead to significant routing and security issues, particularly through peering relationships with large Internet Service Providers (ISPs) and other BGP routers. Managing BGP is thus an important aspect of any network/security operation centre.

Over the years, BGP has encountered various problems that can adversely affect Internet users, including large-scale incidents that cause outages and instability. These issues can arise from natural disasters, software and physical attacks, misconfigurations, censorship, software bugs, and hardware failures [2].

For example, on April 2, 2014, a routing leak involving an Indonesian telecommunications provider redirected a significant portion of global Internet traffic through its network, leading to widespread disruptions and degraded performance for numerous popular websites and services [3]. At the time, it was a manual process to recover from this problem, and even recently, there has been a reported lack of automation for this type of critical incident [4].

To enhance BGP security, researchers have developed mechanisms such as Resource Public Key Infrastructure (RPKI), BGPsec, and, in the future, Autonomous System Provider Authorisation (ASPA) [5]. RPKI, helps prevent BGP route hijacking and leaks by allowing network operators to create Route Origin Authorizations (ROAs) that verify the legitimacy of route announcements. Despite its potential, RPKI faces challenges like gradual adoption and complexity in managing cryptographic keys. As of July 2024, the global ROA adoption rate is 52.6%, with only 6% of the average degree of protection of network users against incorrect announcements using RPKI Route Origin Validation (ROV) [6]. BGPsec also faces slow adoption and has not been widely implemented globally, primarily due to its high computational demands and the need for extensive router upgrades across networks [7].

While large-scale Internet attacks are rare, smaller incidents persist, with between 1,700 and 2,500 BGP leaks or hijacks reported annually [8] despite RPKI's availability [9]. Consequently, the security operations for any autonomous organisation using BGP need additional anomaly detection solutions even as RPKI, BGPsec, and ASPA are fully adopted.

The problem addressed by this paper is to apply an ML approach to anomaly detection for BGP. While this has been investigated previously, we explore the procedure for obtaining an effective model through various aspects, including analysing the dataset, selecting features, comparing classifiers, balancing the dataset, and measuring detection time. Although there is extensive work on ML in general, anomaly detection is less frequently addressed and presents unique challenges

due to the typically small number of anomalous class events compared to normal ones. Additionally, the issue of detection time has been less explored in earlier work, representing a significant research gap since it is an essential parameter for incident reporting. We measure the detection time through precision-recall evaluation and a method of counting true positives (detected anomalies) using an appropriate probability threshold.

The rest of this paper is organised as follows: Section II reviews related works, Section III proposes the solution for the BGP anomaly detection with the results of applying this proposal given in IV. Then, Section V discusses some observations from the work and proposes future work before we conclude.

II. Literature review

BGP anomaly detection has evolved significantly over the years, employing a wide range of techniques to identify issues such as route hijacking and misconfiguration. Early work in this field focused on traditional statistical methods. For instance, Mai et al. [10] proposed a framework using wavelet analysis to provide both temporal and spatial localisation of anomalies. Similarly, Zhang et al. [11] developed signature-based and statistics-based approaches for identifying anomalous routing dynamics.

More recently, ML techniques have gained prominence in BGP anomaly detection. Al-Musawi et al. [12] provided a comprehensive survey of these techniques, categorising them based on their approaches and effectiveness. Further advancing the field, Sanchez et al. [13] compared various ML algorithms using graph features, demonstrating the potential of these advanced techniques. The ongoing innovation in this area is exemplified by Peng et al. [14], who proposed a multi-view framework using graph attention networks for BGP anomaly detection.

In this work, we refer to harmful changes in BGP behaviour as anomalies. Al-Musawi et al. [12] classify BGP problems into four categories:

1) *Direct intended anomaly*: involves various types of BGP hijacking where attackers claim ownership of prefixes belonging to other ASes.
2) *Direct unintended anomaly*: refers to BGP misconfigurations by network operators, such as origin and export misconfiguration; see incidents affecting Indosat [15] and Turk Telekom [16].
3) *Indirect anomaly*: involves malicious activities from web servers, like worm attacks (e.g., Nimda, Code Red, Slammer), causing routing overload [17]–[19].
4) *Link failure*: refers to physical infrastructure issues affecting BGP, such as the blackout in Moscow [20].

To illustrate the need for BGP anomaly detection, consider the routing leak on 12 June 2015. At 08:43 UTC, AS4788 (an ISP in Malaysia) announced a significant portion of the global routing table, causing widespread traffic rerouting [21]. This incident primarily affected Level3 (AS3549) and its customers, resulting in around 179,000 incorrect prefixes

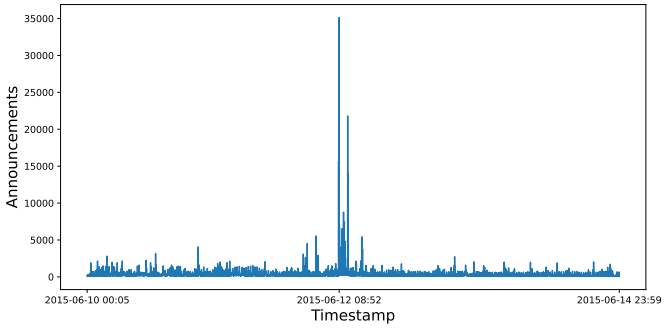

Fig. 1. Prefix announcements received by AS513 during AS4788 routing leak

[22]. The root cause was AS4788 re-advertising peer-learnt routes to upstream providers, creating a route hijack due to an error in an eBGP router's policy. The anomaly caused global Internet slowdowns and significant packet loss, particularly between the Asia-Pacific region and Level3's network. BGP data from RIPE collectors indicated that AS4788's prefix announcements rose from 1,126 to 2,722 during this period, with one new prefix advertised to all peers, leading to over 50,000 announcements at the peak (see Fig. 1). This incident highlights the urgent need for effective BGP anomaly detection to prevent global disruptions.

While such incidents underscore the importance of BGP anomaly detection, there is a notable gap in the literature regarding the measurement of anomaly detection latency for BGP. Most studies focus on developing detection techniques and improving accuracy but do not quantify detection times. The authors in [11] highlight the lack of a real BGP runtime environment for thorough evaluation, complicating actual latency measurements. Even research claiming near real-time capabilities lacks specific latency data [23]. One work that considers detection latency is by [24], which quantifies latency similarly; however, we expand upon it by considering both recall and precision rather than just the F1 score. Additionally, we propose measuring detection latency by counting true positives (detected anomalies) at a specified threshold.

III. Proposed BGP Anomaly detection

This section proposes an ML-based anomaly detection for BGP, addressing key issues such as dataset preprocessing, feature extraction and selection, choosing the best classifier, handling dataset imbalance, and measuring detection time. This paper concentrates on using approaches that do not use Deep Learning, as we are looking for efficient approaches that do not need support from specific hardware, such as GPUs.

A. Dataset preprocessing

A literature survey in [12] shows that the most commonly employed type of anomaly detection uses BGP control plane messages as its dataset. There are two primary sources of such a BGP dataset, which can be accessed publicly in [25] and [26]. However, these BGP messages do not provide information about past states or significant behavioural changes across several ASes; they only indicate incremental changes. Thus,

TABLE I
DATASET OF BGP INCIDENTS WITH TIME IN UTC

Incidents	Collector	Peers	Begin	End
AS9121 leak	rrc05	1853, 12793, 13237	09:20, 24/12/2004	10:03, 24/12/2004
AWS leak	rc04	15547, 25091, 34781	17:10, 22/4/2016	20:00, 22/4/2016
AS4788 leak	rrc04	513, 25091, 34781	08:42, 12/6/2015	10:24, 12/6/2015
Nimda worm	rrc04	513, 559, 6893	13:00, 18/7/2001	12:00, 21/7/2001
Slammer worm	rrc04	513, 559, 6893	05:31, 25/1/2003	19:59, 25/1/2003
Code Red worm	rrc04	513, 559, 6893	10:00, 19/7/2001	20:00, 19/7/2001
Blackout in Moscow	rrc05	1853, 12793	04:40, 25/5/2005	07:40, 25/5/2005
Earthquake in Japan	rv-sydney, rrc06	10026, 2497	09:13, 11/3/2011	15:39, 11/3/2011

feature extraction (described below) must be used on the BGP updates to extract the relevant information.

A high-level view of BGP incidents used in this paper to test anomaly detection, drawn from the RIPE-RIS and RouteViews, is presented in Table I.

B. Comparing the classifiers

Comparing multiple classifiers is crucial in ML, especially for complex tasks like BGP anomaly detection. This process helps identify the most effective algorithm for our dataset, as classifier performance can vary based on data characteristics. We evaluated six supervised learning classifiers from the scikit-learn library [27], including:

1) *Decision Tree*: A model that makes decisions based on asking a series of questions about the features, splitting the data into increasingly homogeneous subsets [28].
2) *K Nearest Neighbours (KNN)*: A non-parametric algorithm that classifies data points based on the majority class among their K nearest neighbours in the feature space [29].
3) *Logistic Regression*: A statistical method that estimates the probability of a binary outcome based on one or more independent variables, using the logistic function to model the relationship [30].
4) *Naive Bayes*: A probabilistic classifier based on Bayes' theorem that assumes independence between features, making it computationally efficient and effective for text classification tasks [31].
5) *Random Forest (RF)*: An ensemble learning method that combines multiple decision trees to improve accuracy and reduce overfitting [32].
6) *Support Vector Machine (SVM)*: An algorithm that finds the optimal hyperplane to separate classes in high-dimensional space, maximising the margin between the closest points of different classes [33].

C. Feature Selection

Feature selection is a crucial step in ML that involves identifying the most relevant features from a dataset [34]. This process reduces data dimensionality, improving model performance and computational efficiency. It also enhances model interpretability by focusing on key features and mitigates overfitting by removing irrelevant or redundant features that may introduce noise. In this work, we employed several feature selection techniques available in scikit-learn [35], including:

1) *Chi-square*: measures feature-target dependence, with higher scores indicating greater feature relevance [36].
2) *Analysis of Variance (ANOVA)*: analyses group mean differences to determine feature significance in predicting the target variable [37].
3) *Mutual Information*: quantifies feature-target dependence, higher values indicate more informative features [38].
4) *Recursive Feature Elimination (RFE)*: iteratively eliminates the least important features based on the scores [39].
5) *Random Forest Feature Importance (RFFI)*: Feature importance is determined by measuring how model performance changes when each feature is randomised or excluded [32].

Those methods above offer a balanced approach, combining univariate techniques (Chi-square, ANOVA, and Mutual Information) that assess individual feature relationships with the target variable and multivariate techniques (RFE and RFFI) that consider feature interactions. These methods were selected for their efficacy, computational efficiency, and suitability for the complex, high-dimensional data typical in BGP systems. This diverse approach provides a comprehensive understanding of the importance of features for anomaly detection.

While techniques like Principal Component Analysis (PCA) are excellent for dimensionality reduction, our chosen methods focus on selecting and preserving original features. This approach is particularly valuable in BGP anomaly detection, where maintaining the interpretability of features and their direct links to specific BGP characteristics is crucial.

D. Addressing dataset imbalance

The challenge in BGP anomaly detection lies in the significant class imbalance, where normal traffic vastly outnumbers anomalous events. This imbalance can lead to biased models that perform poorly in detecting actual anomalies. The key questions to address are how balancing the BGP dataset impacts the overall performance of anomaly detection models and what the most effective techniques for achieving this are without introducing artificial patterns that could lead to overfitting. Several balancing techniques are employed to address this issue [40]. There are two main balancing techniques: oversampling and undersampling. Because the anomaly class proportion is already much smaller than the normal class, the undersampling method is unsuitable. Therefore, we implement some popular oversampling techniques from the imbalanced-learn [41], including:

1) *Random Oversampling*: duplicates minority class instances to balance the dataset, but it can lead to overfitting as it does not introduce new information [42].
2) *SMOTE*: generates synthetic minority instances by interpolating between existing ones and their nearest neighbors, increasing diversity without direct duplication [42].

3) *Borderline SMOTE*: an extension of SMOTE which generates synthetic minority instances near the decision boundary to address likely misclassifications [43].
4) *ADASYN*: an adaptive SMOTE variant, generates more synthetic instances for harder-to-learn minority samples based on density distribution [44].

We compare these methods for their practical advantages in handling BGP anomaly detection. These methods efficiently create synthetic samples of minority class data, allowing us to balance datasets without complex computational requirements. They require no special hardware like GPUs, making them accessible and easy to implement.

E. Detection latency

Rapid detection of BGP anomalies is crucial for network operators for several reasons related to network security and management. For instance, hijacking and misconfiguration incidents can propagate rapidly, affecting up to 90% of the Internet within minutes. This highlights the need for real-time detection mechanisms to prevent widespread disruptions. By quickly identifying and addressing BGP anomalies, network operators can maintain Internet reliability and stability, ensuring continuous service operation. Furthermore, rapid detection allows network operators to respond swiftly to potential threats, minimising the risk of data breaches and unauthorised access. The detection latency is measured by running the ML test over increasing time windows from the start of the known attack time.

IV. Results

This section presents the results of our proposed anomaly detection in BGP, highlighting key aspects to enhance performance. First, we collected the BGP messages from the BGP public datasets and performed preprocessing [45] to extract 47 features. These features, described in Table II, are categorised as volume-based or AS path-based and used as input for our ML-based anomaly detection system. Each dataset is collected from a 5-day observation period with anomaly durations ranging from approximately 1 hour to 4 days, as shown in Table I. Each feature in a dataset is captured over 1 minute, and a count of events is used directly or as a mean over the minute as indicated by the feature description.

Due to space limitations, we present a representative sample of anomalies, providing brief insights into the broader range of results where appropriate. This approach offers a comprehensive overview while highlighting the most significant outcomes.

Next, we compare multiple classifiers using the scikit-learn library in Python to identify the most effective algorithm for detecting BGP anomalies. Feature selection analysis is then conducted to determine the significant features that optimise model efficiency. Additionally, we address dataset imbalances to ensure robust classifier performance. Finally, we measure the detection time for each approach, which is critical for real-time BGP anomaly detection.

TABLE II
List of BGP features

Index	Features	Category
1	Announcements to longer path	AS path
2	Announcements to shorter path	AS path
3	Number of announcements	Volume
4	Average of AS path length	AS path
5	Maximum AS path length	AS path
6	Number of duplicate announcements	Volume
7	Average edit distance	AS path
8-18	Edit distance with k=0-10	As path
19	Maximum edit distance	AS path
20-30	Edit distance with k=0-10 (unique)	AS path
31	Number of flaps	Volume
32	Number of implicit withdrawals (all)	Volume
33	Number of implicit withdrawals with same path	Volume
34	Number of implicit withdrawals with different path	Volume
35	Number of new announcements after withdrawal	Volume
36	Number of plain new announcements	Volume
37	Number of announced prefixes	Volume
38	Number of rare ASes	AS path
39-41	Number of IGP/EGP/INCOMPLETE messages	Volume
42	Number of ORIGIN changes	AS path
43	Average number of rare ASes	AS path
44	Maximum number of rare ASes	AS path
45	Average announcements per prefix (unique)	Volume
46	Maximum AS path length (unique)	AS path
47	Number of withdrawals	Volume

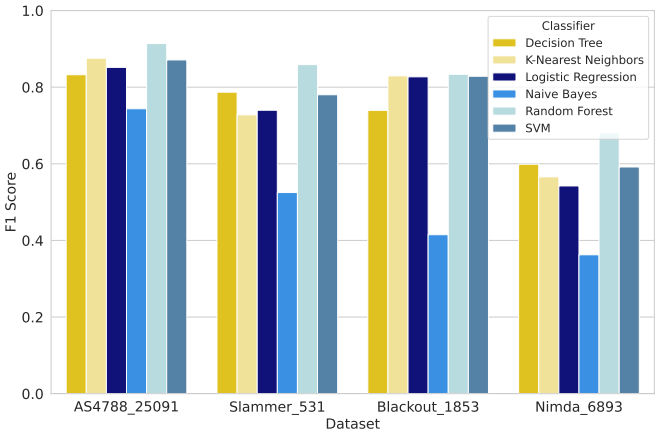

Fig. 2. Performance comparison from various classifiers

A. Classifier comparison and analysis

In our proposed solution, we evaluated six supervised learning classifiers from the scikit-learn library. As illustrated in Figure 2, which displays selected results due to space limitations, the Random Forest (RF) classifier demonstrated superior performance across four distinct network incidents: the AS4788 route leak (observed from AS513), Slammer worm attack (observed from AS2497), Nimda worm attack (observed from AS13237), and a blackout incident (observed from AS3257). RF's effectiveness was quantified by its consistently lower Mean Absolute Error (MAE), derived from the highest F1 scores among all classifiers tested. Table III provides a detailed breakdown of these performance metrics. The RF classifier's ability to handle complex, high-dimensional data and its resistance to overfitting likely contributed to its robust performance across these varied network anomalies.

TABLE III
PERFORMANCE EVALUATION USING MAE ON VARIOUS CLASSIFIERS

Parameters	DT	KNN	LR	NB	RF	SVM
MAE of F1 score	0.0708	0.0790	0.0879	0.2468	0.0059	0.2084

TABLE IV
PERFORMANCE EVALUATION OF FEATURE SELECTION

Parameters	Chi	ANOVA	Mutual	RFFI	RFE
Execution time (s)	0.011	0.021	1.318	0.265	10.319
MAE of F1 score	0.0108	0.0082	0.0086	0.0012	0.0057

B. Feature selection and analysis

Feature selection occurs during the training phase; consequently, computational performance is less of an issue than during the run-time; however, it is still useful for evaluating relative performance. We evaluated six feature selection methods based on average execution time and Mean Absolute Error (MAE) from the best F1 score. As shown in Table IV, RFFI is generally preferred over Chi-square for feature selection, despite Chi-square being faster. This is because RFFI performs feature selection offline during model training, not affecting live detection runtime. Additionally, RFFI achieves lower MAE scores than other methods, making it a better choice for applications where predictive performance is crucial.

After choosing RFFI for feature selection, we applied it to various datasets to find the top 10 features, as shown in Table V. Although the top features vary slightly even within the same anomaly type, our analysis in Table VI reveals that the number of announcements (feature 3) is the most significant across all BGP incidents: worm attacks, routing leaks, blackouts, and earthquakes. This feature spikes before and during these anomalies, indicating increased BGP activity as routers adjust to topology changes. In routing leaks, the announcement to shorter path (feature 2) feature is also a key indicator, aligning with BGP path selection, where the shortest AS path is preferred. Routing leaks often propagate incorrect information, appearing as shorter paths. For worm attacks, blackouts, and earthquake incidents, the number of withdrawal features (features 47, 32, and 33) is also significant, as these events often lead to disruptions in network connectivity. Similarly, in the case of blackouts and earthquakes, physical damage to infrastructure can result in BGP outages, prompting routers to withdraw routes that are no longer viable.

To evaluate our feature selection approach, we employed RFFI based on the F1 score to determine the optimal number of features for each dataset. Table VII presents selected results, indicating that the number of features yielding top performance varies significantly, influenced by both the type of anomaly and the observation point (AS peer). Our analysis revealed a notable difference in performance between the baseline (all features) and the selected features (RFFI). Specifically, we observed an average difference of 0.0451 (or 4.51%) in F1 score across all datasets. This performance gap underscores the complexity of BGP anomaly detection and highlights the importance of combining appropriate feature selection with effective oversampling techniques. These results demonstrate the substantial benefits of RFFI in enhancing overall detection accuracy while highlighting the need for adaptive approaches in BGP anomaly detection systems.

TABLE V
TOP 10 FEATURES FROM RFFI

Datasets	Top 10 Features by RFFI
Japan_2497	9, 21, 39, 3, 32, 37, 33, 7, 11, 23
Moscow_1853	9, 21, 33, 39, 3, 1, 32, 2, 37, 7
Moscow_12793	9, 33, 21, 3, 42, 39, 32, 11, 6, 20
Moscow_13237	37, 39, 3, 32, 47, 33, 43, 1, 42, 2
Nimda_513	47, 39, 3, 8, 37, 20, 35, 31, 9, 32
Nimda_559	39, 47, 37, 3, 7, 32, 33, 20, 8, 35
Nimda_6893	39, 37, 3, 32, 9, 2, 33, 21, 1, 24
Slammer_513	35, 47, 41, 39, 8, 20, 34, 42, 3, 9
Slammer_559	47, 31, 8, 20, 41, 39, 3, 37, 9, 42
Slammer_6893	47, 35, 39, 20, 8, 31, 9, 21, 3, 2
Code_513	39, 47, 37, 3, 32, 9, 35, 20, 33, 21
Code_559	31, 39, 47, 37, 9, 3, 32, 8, 20, 24
Code_6893	34, 39, 37, 20, 8, 3, 32, 9, 47, 31
Malaysia_513	31, 47, 8, 35, 20, 27, 37, 3, 34, 15
Malaysia_25091	8, 32, 20, 37, 3, 34, 39, 9, 33, 21
Malaysia_34781	31, 35, 8, 20, 2, 37, 6, 3, 24, 47
AWS_15547	31, 35, 9, 3, 21, 47, 2, 20, 8, 33
AWS_34781	31, 9, 8, 2, 21, 32, 47, 26, 11, 24
AWS_25091	33, 12, 6, 24, 3, 11, 37, 8, 2, 31
AS9121_13237	13, 25, 2, 42, 33, 24, 3, 12, 14, 11
AS9121_1853	47, 14, 26, 39, 13, 35, 11, 24, 3, 19
AS9121_12793	13, 37, 25, 3, 33, 39, 32, 11, 2, 23

TABLE VI
TOP 5 MOST FREQUENT FEATURES FOR ALL ANOMALIES

Anomaly type	Top 5 features
Worm attack	3, 39, 9, 47, 8
Routing leak	3, 2, 8, 31, 20
Blackout, earthquake	3, 32, 33, 39, 9

C. Analysis of dataset balancing

Most anomaly detection problems suffer from a highly imbalanced class distribution. In the AS4788 dataset, the normal class accounts for 98.57% of the data, while in the AS9121 and Blackout datasets, it represents 98.9% and 97.62%, respectively. Therefore, we employ several oversampling techniques to address the dataset imbalance, such as Random Oversampling, SMOTE, Borderline SMOTE, and ADASYN.

TABLE VII
TOP FEATURE SELECTION BY RFFI WITH THE HIGHEST F1 SCORE

Datasets	Number of top features, x	F1 score	
		All features	Top x features
Nimda_513	34	0.6372	0.6448
Nimda_559	23	0.5208	0.533
Nimda_6893	45	0.6374	0.6416
Slammer_513	4	0.7922	0.8091
Slammer_559	17	0.7246	0.7468
Slammer_6893	17	0.6862	0.7149
Code_513	33	0.6844	0.7042
Code_559	18	0.4229	0.4694
Code_6893	27	0.7329	0.7432
AS4788_513	40	0.5992	0.6102
AS4788_25091	22	0.7594	0.7985
AS4788_34781	43	0.9053	0.9169
AS9121_1853	11	0.2735	0.3093
AS9121_12793	28	0.3775	0.4398
AS9121_13237	30	0.9805	0.9864

TABLE VIII
PERFORMANCE EVALUATION ON VARIOUS OVERSAMPLING TECHNIQUES

Metrics	Without	Random	SMOTE	Borderline	ADASYN
MAE	0.0322	0.0175	0.0114	0.0177	0.0166

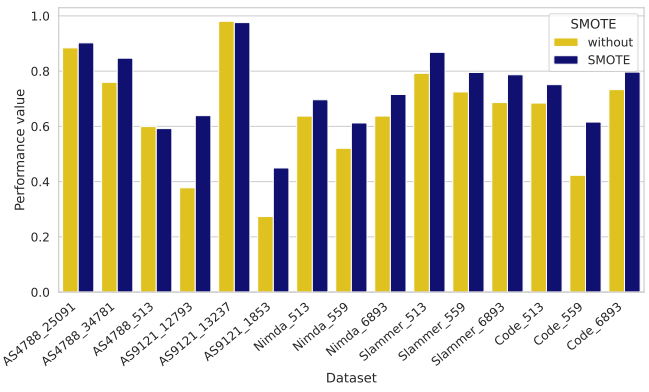

Fig. 3. Performance evaluation for SMOTE

We use the F1 score to evaluate the performance of various oversampling techniques. To better understand their impact, we also calculate the Mean Absolute Error (MAE) based on the best F1 score achieved by each method. Table VIII shows that most oversampling techniques improve F1 scores compared to the baseline (without oversampling). Among the methods tested, SMOTE stands out as the best choice. It achieves the lowest MAE while maintaining a high F1 score.

To illustrate SMOTE's effectiveness, we included a graph comparing F1 scores between datasets with SMOTE and those without oversampling. Fig. 3 shows a substantial overall increase of 18.83% in F1 scores across all datasets. Improvements vary significantly, ranging from just 2.05% for the AS4788_25091 dataset to an impressive 69.21% for the AS9121_12793 dataset. This variation highlights SMOTE's adaptability to different dataset characteristics and its potential to significantly enhance detection performance, particularly in cases of severe class imbalance.

D. Anomaly detection latency

An important aspect of any anomaly detection is the latency between the start of the attack and the detection. BGP operates with convergence in the order of minutes; e.g., APNIC reported 2023 BGP data with mean convergence of the order of 1 minute and outliers up to five minutes [46]. Consequently, we would want detection latency to operate in the order of a few minutes, and we would not expect to detect anomalies in less than 1 minute.

In practice, the detection performance changes gradually over the detection timeframe. To highlight this, we show the evolution of precision and recall over time for AS4788 route leak in Fig. 4. These results were generated by training on the AS peer (AS34781) dataset and repeating the ML prediction on a different dataset with a different AS peer (AS25091). The metrics were measured over a period that included an increasing time window from the start of the attack in 1-minute intervals. Since no true positives were detected before the anomaly started, the precision/recall is undefined before this point in time and set to zero on the graph.

These results show the averaged precision/recall performance of many events across the test dataset. Generally, the graphs show better precision/recall metrics with increasing time after the attack. This is unsurprising as the volume of attack traffic increases as a portion of the total traffic is analysed so that any detection errors have a proportionally smaller effect. However, such general precision/recall data do not give good guidance on minimum detection latency, a fact that others do not report. Instead, we note that even one event that triggers a true positive is enough to signal an alarm, assuming that false positives are rare or non-existent. We note that the graph in Fig. 4 shows a sharp decline in recall and a slight decrease in precision at approximately 20 minutes after the anomaly starts. This was due to a significant reduction in BGP messages at that point in time for an unknown reason.

A more effective approach to investigating detection latency is to count true positive events within a specified time window. Figure 5 illustrates this for the dataset trained on AS34781 and tested on AS25091. The graph shows data from two days before the attack until the attack time, with each data point representing a one-minute increase in the window. The period two days before the attack is not visible as the zero values encompass the count for that time.

Figure 5 demonstrates that the number of true positives increases over time and varies with the probability threshold used to determine a positive anomaly. Higher probability thresholds reduce the likelihood of false positives. Our analysis shows that a threshold of 0.75 yields no false positives in the two days preceding the attack and detects the attack within seven minutes, which is a reasonable timeframe. In contrast, a threshold of 0.5 proves too low, resulting in false positives. This aligns with the poor precision observed in Figure 4, which uses the default 0.5 threshold.

We should also add that, although we had no false positives in the period before the attack – when a suitable threshold of 0.75 is set – our technique uses aggregated message statistics over a window; consequently, although the period of the attack (after detection delay) is correctly marked as true positive, there will be some messages during the window that are benign and mixed in with the malicious messages. The false negative events are a result of the detection delay, and consequently, any standard ML metrics are influenced by the attack duration (or measurement window), and this should be taken into account. Future work should consider the window size that is passed to the ML and the consequential influence on the detection results. In a practical deployment, a security team will need to trade off the detection probability threshold against the false positive rate, i.e., the amount of organisational risk posed by either a false negative or false positive. This is often a business-level decision and may even depend upon the business type of prefix/AS that is affected. This work aims to give information to aid this type of decision, not make the specific business decision about the precision/recall threshold selected.

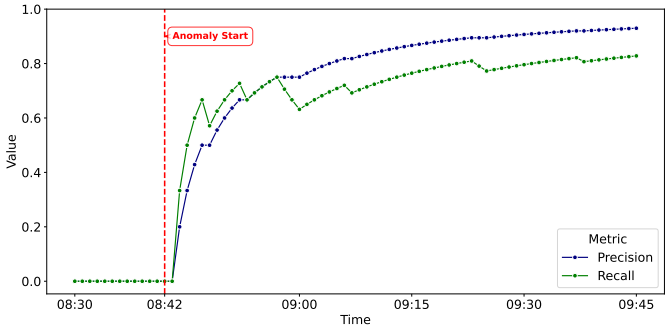

Fig. 4. Precision/recall performance for AS4788 route leak detection

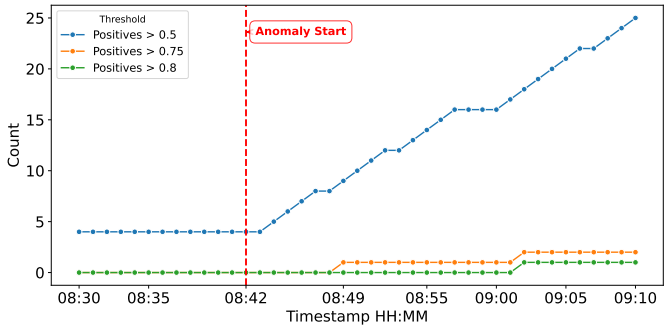

Fig. 5. Detection using true positive counting with probability thresholds

V. Discussion

Our comprehensive analysis of various classifiers and feature selection methods for BGP anomaly detection reveals that RF emerges as the best classifier, particularly when combined with RFFI for feature selection. This combination, along with the application of SMOTE for oversampling, yields the most robust and accurate results. The superior performance of the RF classifier can be attributed to its ensemble learning approach, which mitigates overfitting and effectively handles the complexity of BGP data. It would be interesting to explore the advantages of Deep Learning in future work while trying to keep the hardware costs compatible with practical deployment scenarios where GPU architectures are generally not available.

One significant insight from the results relates to the fact that data is collected from multiple vantage points or different AS peers. The choice of observation point can significantly impact the features and patterns present in the data, consequently affecting the detection performance of the ML model. As illustrated in Figure 6, different observation points for the same BGP incidents lead to varying performance levels. Determining the most optimal observation points and their combination or exploring techniques to effectively integrate and leverage data from multiple sources could potentially enhance the model's ability to capture a more comprehensive view of the network dynamics, thereby improving anomaly detection performance. This is a promising area for future research, and this paper is the first to describe it.

Feature optimisation is highly dependent on the specific type of anomaly being targeted. Different types of BGP anomalies, such as route leaks, hijacks, or misconfiguration

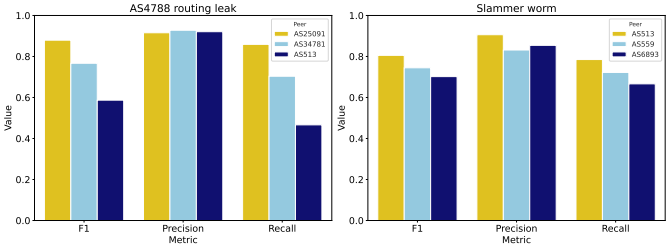

Fig. 6. Performance evaluation from different observation points (AS peer)

events, may exhibit distinct patterns and characteristics in the data. This raises the question of whether it is possible to develop robust ML models that can employ a common feature set for effectively detecting various anomalies or if tailored feature engineering is required for each specific anomaly type.

To tackle the challenge of finding the right features for different types of anomalies, we could look into techniques that can automatically select and extract relevant features. This way, we do not have to manually engineer features for each new anomaly type. Additionally, we could explore if the pattern learnt by models for one type of anomaly can be transferred and used to detect other types of anomalies. By using such *transfer learning* [47], it may be possible to develop more flexible and adaptable anomaly detection systems that can identify various kinds of anomalies without needing extensive retraining or feature engineering for each new anomaly. These considerations are for future research.

VI. Conclusion

Automated anomaly detection using ML for network operations is an emerging area of research. This paper has investigated this in the context of BGP anomalies. This work has found that detecting BGP anomalies faces several challenges to provide good detection, including feature selection and dataset balancing. The work is the first to observe that the choice of observation points or AS peers impacts the data features and patterns, affecting model performance. Future work determining the optimal combination of data sources or integrating data from multiple sources could improve the capture of network dynamics and detection performance.

Finally, this investigation is pioneering in its approach to measuring the BGP anomaly detection latency. We are the first to utilise the probability threshold of true positive anomalies to assess this latency. The results show that the detection can be achieved within seven minutes for real-world attack data, which is compatible with BGP update timescales, which are of the order of minutes. This work provides a workflow for future BGP anomaly detection in secure network operations.

VII. Acknowledgment

The author gratefully acknowledges the financial support provided by the Indonesian Education Scholarship Program from the Center for Higher Education Funding (BPPT) under the Ministry of Education, Culture, Research, and Technology, Republic of Indonesia, and the Indonesia Endowment Fund for Education (LPDP).

REFERENCES

[1] K. Lougheed and Y. Rekhter, "RFC 1105: Border Gateway Protocol (BGP)." RFC - Experimental on IETF Datatracker, June 1989, Available online https://datatracker.ietf.org/doc/rfc1105/.

[2] J. Sherman, "It's far too easy for countries like russia and china to hijack internet traffic," Nov. 2018. Available online https://slate.com/technology/2018/11/bgp-hijacking-russia-china-protocols-redirect-internet-traffic.html, accessed=Jan. 11, 2023.

[3] R. Wihelm, "BGP leaks in Indonesia." RIPE Labs, 4 April 2014, Available online https://labs.ripe.net/author/wilhelm/bgp-leaks-in-indonesia/.

[4] D. Shahjee and N. Ware, "Integrated network and security operation center: A systematic analysis," *IEEE Access*, vol. 10, pp. 27881–27898, 2022.

[5] A. Azimov and et al., "BGP AS_PATH Verification Based on Autonomous System Provider Authorization (ASPA) Objects." Active Internet-Draft (sidrops WG) on IETF Datatracker, 8 July 2024, Available online, https://datatracker.ietf.org/doc/draft-ietf-sidrops-aspa-verification/.

[6] "MANRS Observatory." Accessed online, 31 July 2024, https://observatory.manrs.org/#/overview. MANRS Observatory.

[7] M. Lepinski and K. Sriram, "RFC 8205: BGPSEC Protocol Specification." RCF on IETF Datatracker, Sept 2017, Available online https://datatracker.ietf.org/doc/html/rfc8205.

[8] A. Mathurin, "A Regional Look into BGP Incidents in 2020." MANRS Document, 12 March 2021, Available online https://www.manrs.org/2021/03/a-regional-look-into-bgp-incidents-in-2020/.

[9] N. H. Hammood and B. Al-Musawi, "Using BGP features towards identifying type of BGP anomaly," in *2021 International Congress of Advanced Technology and Engineering (ICOTEN)*, pp. 1–10, 2021.

[10] J. Mai, L. Yuan, and C.-N. Chuah, "Detecting BGP anomalies with wavelet," in *NOMS 2008 - 2008 IEEE Network Operations and Management Symposium*, pp. 465–472, 2008.

[11] K. Zhang, A. Yen, and et al., "On Detection of Anomalous Routing Dynamics in BGP," in *Networking 2004*, (Berlin, Heidelberg), pp. 259–270, Springer Berlin Heidelberg, 2004.

[12] 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.

[13] O. R. Sanchez and et al., "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*, Big-DAMA '19, (New York, NY, USA), p. 35–41, Association for Computing Machinery, 2019.

[14] S. Peng and et al., "A multi-view framework for BGP anomaly detection via graph attention network," *Computer Networks*, vol. 214, p. 109129, 2022.

[15] A. Toonk, "Hijack event today by Indosat." BGPmon-Hijack, News and Updates, 3 April 2014, Available online https://www.bgpmon.net/hijack-event-today-by-indosat/.

[16] A. Toonk, "Turkey hijacking IP addresses for popular global DNS providers." BGPmon Hijack, News and Updates, 29 March 2014, Available online, https://www.bgpmon.net/turkey-hijacking-ip-addresses-for-popular-global-dns-providers/.

[17] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang, "Analysis of BGP Update Surge during Slammer Worm Attack," in *Distributed Computing - IWDC 2003* (S. R. Das and S. K. Das, eds.), (Berlin, Heidelberg), pp. 66–79, Springer Berlin Heidelberg, 2003.

[18] L. Wang and et al., "Observation and Analysis of BGP Behavior under Stress," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, (New York, NY, USA), p. 183–195, ACM, 2002.

[19] S. Deshpande, M. Thottan, and B. Sikdar, "Early detection of BGP instabilities resulting from internet worm attacks," in *IEEE GLOBECOM 2004*, vol. 4, pp. 2266–2270 Vol.4, 2004.

[20] M. Dillon, "Re: More on Moscow Power Failure(was re: Moscow: Global Power Outage)." NANOG mailing list archives, 26 May 2005, Available online https://seclists.org/nanog/2005/May/710.

[21] N. Kephart, "Route leak causes global outage in level 3 network." ThousandEyes Outage Analysis, 15 June 2015, Available online https://www.thousandeyes.com/blog/route-leak-causes-global-outage-level-3-network.

[22] A. Toonk, "Massive route leak causes internet slowdown." BGPmon-BGP instability, 12 June 2015, Available online https://www.bgpmon.net/massive-route-leak-cause-internet-slowdown/.

[23] B. Al-Musawi, P. Branch, and G. Armitage, "RTBADT-real-time BGP anomaly detection tool v0.1," Technical Report 180129A, Swinburne University of Technology, Melbourne, Australia, Jan. 2018.

[24] K. Hoarau and et al., "BGNN: Detection of BGP Anomalies Using Graph Neural Networks," in *2022 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1–6, 2022.

[25] "RIS data access." https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/ris-data-access. Accessed: 2022-11-30.

[26] "MRT data files." http://www.routeviews.org/routeviews/index.php/mrt-data-files/. Accessed: 2022-11-30.

[27] F. Pedregosa and et al., "1. Supervised learning." scikit-learn, https://scikit-learn.org/stable/supervised_learning.html, note = [Online; accessed 1 Aug 2024].

[28] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 1993.

[29] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[30] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*. Wiley Series in Probability and Statistics, New York, NY, USA: Wiley-Interscience, 2 ed., 2000.

[31] Y. Wang and I. H. Witten, "Text classification using naive bayes," in *Proceedings of the International Conference on Machine Learning*, vol. 1, pp. 1–8, 1999.

[32] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, p. 1157–1182, mar 2003.

[35] F. Pedregosa and et al., "Feature selection." scikit-learn, https://scikit-learn.org/stable/modules/feature_selection.html. [Online; accessed 1 Aug 2024].

[36] H. Liu and R. Setiono, "Chi2: feature selection and discretization of numeric attributes," in *Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence*, pp. 388–391, 1995.

[37] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine Learning Proceedings 1992*, pp. 249–256, Morgan Kaufmann, 1992.

[38] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[39] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine learning*, vol. 46, no. 1-3, pp. 389–422, 2002.

[40] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, p. 1263–1284, sep 2009.

[41] G. Lemaître, F. Nogueira, and C. K. Aridas, "2. Over-sampling." imbalanced-learn, https://imbalanced-learn.org/stable/over_sampling.html, note = [Online; accessed 1 Aug 2024].

[42] N. V. Chawla and et al., "SMOTE: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[43] H. Han, W.-Y. Wang, and B.-H. Mao, "Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning," in *International conference on intelligent computing*, pp. 878–887, Springer, 2005.

[44] H. He and et al., "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 1322–1328, IEEE, 2008.

[45] P. Fonseca and et al., "BGP Dataset Generation and Feature Extraction for Anomaly Detection," *Proc. IEEE Symposium on Computers and Communications*, vol. 2019-June, 2019.

[46] G. Huston, "BGP in 2023 — BGP updates." APNIC Annual Report, 10 Jan 2024, Available online https://blog.apnic.net/2024/01/10/bgp-in-2023-bgp-updates/.

[47] S. Aburakhia and et al., "A transfer learning framework for anomaly detection using model of normality," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0055–0061, 2020.