GraphBGP: BGP Anomaly Detection Based on Dynamic Graph Learning

Zheng Wu, Yanbiao Li, *Member, IEEE*, Xin Wang, *Senior Member, IEEE*, Zulong Diao, *Member, IEEE*, Weibei Fan, *Member, IEEE*, Fu Xiao, *Senior Member, IEEE*, and Gaogang Xie, *Senior Member, IEEE*

***Abstract*—Detecting anomalous BGP (Border Gateway Protocol) messages is critical for securing inter-domain routing systems over autonomous system (AS)-level networks. The dynamic nature of routing policies, massive scale of global routes, and incomplete global topology visibility make BGP anomalies exceptionally challenging to identify—let alone trace back to malicious or misconfigured ASes. To effectively overcome these barriers, this paper proposes *GraphBGP*, a novel BGP anomaly detection method that dynamically constructs real-time AS-level topologies, achieves precise anomaly detection and classification, and accurately traces malicious or misconfigured ASes. Specifically, to address the evolving nature of BGP routing status, *GraphBGP* constructs an attributed AS-level graph that dynamically integrates node and edge attributes. It intelligently tracks BGP updates to refresh this graph efficiently. Leveraging this enriched, up-to-date representation, *GraphBGP* employs tailored detection and tracing models grounded in graph convolutional networks (GCNs), enabling precise anomaly identification and source tracing. Comprehensive experiments with real-world and synthetic datasets demonstrate that *GraphBGP* achieves state-of-the-art anomaly detection accuracy while significantly reducing inference time, even under partial BGP network visibility. Furthermore, *GraphBGP* precisely traces malicious or misconfigured ASes within a short time period of 7 milliseconds after anomaly detection, enabling rapid mitigation.**

***Index Terms*—Border gateway protocol, anomaly detection, graph convolutional networks, incremental update.**

Received 3 February 2024; revised 20 August 2024 and 7 March 2025; accepted 14 August 2025. Date of publication 8 September 2025; date of current version 24 September 2025. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFB3104800; in part by the Natural Science Foundation of China under Grant 62402234, Grant 62102196, and Grant 62372248; and in part by the Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications under Grant XK0040923168. The associate editor coordinating the review of this article and approving it for publication was Dr. Abdallah Shami. *(Corresponding authors: Gaogang Xie; Yanbiao Li.)*

Zheng Wu, Weibei Fan, and Fu Xiao are with the College of Computer, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing 210049, China (e-mail: zwu@njupt.edu.cn; wbfan@njupt.edu.cn; xiaof@njupt.edu.cn).

Yanbiao Li and Gaogang Xie are with the Computer Network Information Center, Chinese Academy of Sciences, Beijing 100083, China (e-mail: lybmath@cnic.cn; xie@cnic.cn).

Xin Wang is with the Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY 11794 USA (e-mail: x.wang@stonybrook.edu).

Zulong Diao is with the School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: diaozulong@ict.ac.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TIFS.2025.3607239, provided by the authors.

Digital Object Identifier 10.1109/TIFS.2025.3607239

I. Introduction

THE Border Gateway Protocol (BGP) is one of the most important protocols for computer networks, which is applied to ensure the global network reachability between Autonomous Systems (ASes) [1]. However, the protocol does not include the authentication and validation steps, making the global networks more vulnerable to malicious attacks and misconfiguration, such as prefix hijack, route leak, network outage and so forth [2]. Due to the global nature of BGP, these attacks would generally cause much more severe consequences than a normal attack, leading to a wide and fast spread throughput the whole network [3]. For example, a Facebook anomaly event caused by a BGP misconfiguration on October, 4, 2021 led to a large scale network outage, as a result of which, the service of Facebook was disconnected from the global network for almost six hours, and its market value shrunk by nearly 5% (an estimated loss of more than $6 billion) [4].

To improve the security of global networks, two types of methods are generally adopted: proactive defense and passive detection. The proactive defense methods aim to filter out abnormal routes according to the out-of-band information, or alter the protocol by introducing security procedures such as Resource Public Key Infrastructure (RPKI), Autonomous System Provider Authorization (ASPA) and so on [5]. Nevertheless, it is hard for these methods to fully function as it is very costly and time-consuming to deploy these schemes at a large scale, and different ISPs have their different business interests [6].

Therefore, ISPs generally prefer to employ passive detection approaches to ensure the BGP security [7]. These methods can help detect anomalies and locate where they occur for subsequent processing by network operators. Passive detection schemes can be further divided into two categories, rule-based and Artificial Intelligence (AI)-based. Generally, rule-based methods [8] construct a database to match each incoming update message, and the messages not matched by the database are recognized as abnormal behaviors. This type of method is efficient for some simple anomalies, such as prefix hijack or one-hop hijack. The dynamic network degrades its performance because it uses rigid rules and is sensitive to incomplete information.

The success of artificial intelligence (AI) techniques in various domains (such as natural language processing and image

1556-6021 © 2025 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

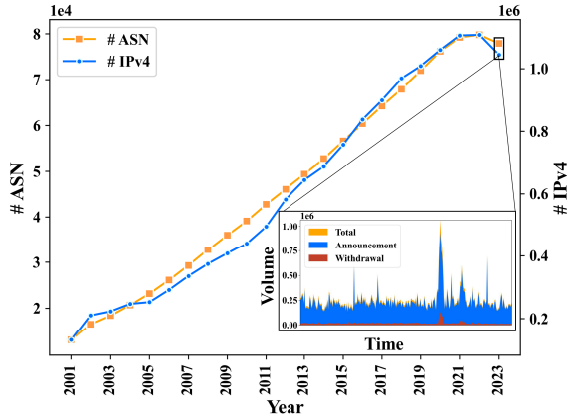

Fig. 1. The increasing and dynamic route status of BGP observed by the collector routeview2 from 2001 to 2023. (The sub-figure is the number of route updates messages comprising the total, announcement and withdrawal during the whole day of May, 11th, 2023.)

processing) has sparked an interest in leveraging AI-based methods, such as Graph Convolutional Networks (GCN), to enhance BGP anomaly detection. In this paper, we will realize online BGP anomaly detection method using the customized GCN.

A. Motivation and Challenges

Implementing online BGP anomaly detection in real-world deployments presents several challenges:

Firstly, detecting abnormal behaviors in a dynamic, large-scale network is a significant challenge [9]. As of 2023, the global network consists of approximately 70 thousand ASes, hundreds of thousands of interconnected edges, and millions of IP prefix routes. At peak time, the number of route update messages may even exceed 10^6 per second, as illustrated in Fig. 1. Given this scale and complexity, an essential prerequisite is developing precise models to capture AS-wise behaviors in such a vast and dynamic network.

Secondly, the incomplete BGP routes observed from the global vantages make it challenging to infer the overall operational state of BGP. Major AS-level Internet stakeholders typically keep their BGP routing information proprietary and are unwilling to disclose it [10]. Consequently, obtaining comprehensive and complete route data is often difficult. Therefore, addressing the limitations posed by restricted visibility into the global BGP network topology is a critical challenge.

Furthermore, merely detecting anomalies offers limited practical value for operation and maintenance (O&M) personnel. Designing separate methods for different functions may improve specificity but can hinder system efficiency and responsiveness. Therefore, a unified framework needs to be designed to integrate anomaly detection, type identification, and root cause localization, ensuring a more effective and streamlined approach.

B. Key Contributions

To address these challenges, this paper proposes a novel BGP security scheme named *GraphBGP*. We first model the BGP network as an attributed graph and propose an incremental update mechanism adapting to dynamic status. We construct a BGP anomaly detection framework based on graph learning, combining local and statistical knowledge. Besides, we present a root cause location method based on an auto-encoder framework for tracing misbehaved ASes. Thus, our method can not only efficiently model the evolving changes of networks but also detect anomalies and pinpoint the root cause accurately with incomplete information.

The contributions of our work are fourfold as follows:

- We propose a global BGP graph representation method for online anomaly detection. To efficiently adapt to the dynamic changes in the BGP network, we also design a novel incremental update mechanism that incorporates key decision-making factors from routers.
- We propose a novel method for both BGP anomaly detection and root cause location, built upon the above BGP graph representation. By carefully designing and aggregating rich knowledge, our approach effectively detects anomalies even with partial visibility and accurately localizes root causes at the AS level.
- We have developed a workflow for online BGP anomaly detection and root cause location, which includes the following steps: data collection and labeling, graph building and update, anomaly detection and root location. This process enables us to accomplish multiple tasks, including detecting anomalies, identifying their nature, and pinpointing their root causes.
- We evaluate our method using both real-world and synthetic BGP datasets containing ten BGP anomaly events, comparing it with six baseline methods. Experimental results demonstrate that our method outperforms state-of-the-art approaches in terms of detection accuracy and time efficiency, particularly when dealing with the incomplete information.

The remainder of this paper is organized as follows. Section II reviews the related works. Section III provides the preliminary knowledge and defines the problems to be addressed. Section IV presents a novel BGP anomaly detection method. The performance of the proposed method is evaluated by comparing with other methods for BGP anomaly detection in Section V. Finally, Section VI concludes this paper.

II. Related Work

Existing methods on BGP anomaly detection generally include two categories, namely, passive and active detection. Most ISPs prefer to use the passive detection methods because they will not occupy the additional bandwidth resource.

The passive detection methods can be further classified into two types, the rule-based and AI-based methods. The rule-based methods need to construct the rule sets, and then match them against each BGP update message. These methods [11] can be implemented in real time and provide explanatory insights into the detected anomalies. For example, Sermpezis et al. [8] propose a self-operated control-plane approach Artemis. This method classifies the BGP anomaly into N types according to the location of the suspicious AS on an AS path

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

and then customizes the construction of information bases and matching schemes for each type of anomaly. To further reduce the false positive rate and filter out legitimate MOAS (Multiple Origin ASes) conflicts, Qin et al. [12] analyse the causes of legitimate MOAS in depth and devise a filtering mechanism. However, this method is only effective for detecting specific anomaly types, such as prefix hijacks and one-hop hijacks. In addition, this type of method is very rigid in rule matching, and hard to adapt to the dynamic changes and work in the decentralized BGP network.

The AI-based methods usually utilize the classifiers of a machine learning or deep learning algorithm to infer the BGP anomalies within a time window according to different features extracted from the BGP update behaviors. This type of methods can learn the BGP anomaly patterns better in the presence of network dynamics, thus having a better generalization performance. For example, Hoarau et al. [13] propose a feature extraction tool that can obtain 46 BGP-related features to capture the BGP anomaly behaviors. Cheng et al. [14] put forward a Long Short Term Memory (LSTM) model for BGP anomaly detection using wavelet transformation to extract the multi-scale time series features. Although these methods could identify between normal and abnormal samples, they generally cannot well determine the anomaly type and provide the cause for anomaly.

To better capture the rich information, the AS topology can be modeled using a graph model. Although the graph knowledge has been exploited to analyze the network features of BGP [15], [16], it can not be easily employed to achieve quick anomaly detection, as it is also very hard to deal with such large-scale graph and obtain the up-to-date global graph topology of ASes. There have been some studies on graph-based BGP anomaly detection. For example, Peng et al. [17] present a multi-view model, in which, the temporal and statistical features are linked into one fully-connected graph, and then a graph attention network (GAT) is used to extract the abnormal representation. However, the large topology of BGP network makes it time-consuming to obtain these graph features.

In terms of root cause tracing, Feldmann et al. [18] propose a method to localize the stability of Internet routing by assuming that the root causes appear either on the new AS path or the old substituted AS paths, but this assumption does not stand in the pesence of the incomplete visibility of topology. To address this issue, Javed et al. [19] design Poiroot, a real-time system allowing ISPs to isolate the root causes of any path change that affects the prefixes. Poiroot uses the BGP data but also combines the active measurement to gain the visibility of global networks. Unfortunately, this method can be only used to trace the root cause for a given AS, but not to adapt to the global BGP monitoring for its high complexity, and not focus on detecting the anomaly types. Besides, an ontological graph of ASes is proposed for locating the IP prefix hijacking [20], which possesses considerable complexity when the graph with large numbers of edges.

Several graph-based studies are also dedicated to anomaly detection. Kreuzer et al. [21] propose a transformer-based method for graph anomaly detection. Another transformer-based anomaly detection framework presented by Liu et al. [22] well learn discriminate knowledge from coupled spatial-temporal dynamic graphs. Facing a large graph like BGP network, this type of method generally consumes a lot of computing and memory resources, with the complexity of $O(n^2)$ or $O(e^2)$ where n and e are the numbers of nodes and edges respectively. Ma et al. [23] survey the graph anomaly detection with deep learning. However, these existing methods cannot be directly adopted for BGP anomaly detection due to the BGP network's intrinsic characteristics, such as its large-scale and decentralization nature.

Our approach differs from the existing methods in two key ways. First, we perform anomaly detection across the global BGP network while simultaneously tracing root cause by modelling the global ASes as an AS-level graph, which includes about 70 thousand nodes and hundreds of thousands of edges. Second, we introduce an incremental BGP graph update mechanism to adapt to dynamic BGP networks in real time. Additionally, by leveraging GCN, we aggregate extensive neighboring information and statistical knowledge to assess the state of each AS and the global routing status, thereby enabling early anomaly detection even with incomplete route information.

TABLE I
KEY FIELDS AND THEIR MEANINGS IN BGP RIB AND UPDATE MESSAGE

Field	Meaning
AS Path	AS sequence through which an update message passes in turn.
Prefix	The destined IP prefix comprising IPv4 and IPv6.
Peer AS	The AS from where the update message is received.
Operation	The action to the routing information with the update message, which usually is withdrawal or announcement.

III. PRELIMINARIES AND PROBLEM STATEMENT

Before describing our scheme in detail, we introduce some notations and concepts first, and then formulate our problem. Table I lists some important data fields in route information base (RIB) or BGP update messages.

A. Classical Malicious Attack Models

1) BGP Anomaly Models: There are three main types of BGP anomalies: prefix hijack, route leak and network outage. See Fig. 2 for some example.

Prefix Hijack: An attacker configures an AS router and announces the prefix(es) owned by other AS(es) to the networks, in order to hijack the traffic destined to the prefix(es) into the attacker ASes. In Fig. 2(a), the attacker AS4 forges the ownership of AS1 to prefix P. From the perspective of the vantage point (VP), the path destined to prefix P changes from the AS path (AS3, AS2, AS1) to the AS path (AS3, AS2, AS4), causing the path offset.

Route Leak: According to the RFC 7908 [24], this type of BGP anomaly is the propagation of BGP routing announcements beyond their intended scope. In Fig. 2(b), attacker AS2 leaks the prefix P owned by AS1 to its neighboring provider AS4. Compared to the expected route, this detouring path has a direct negative impact on the network traffic flow.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

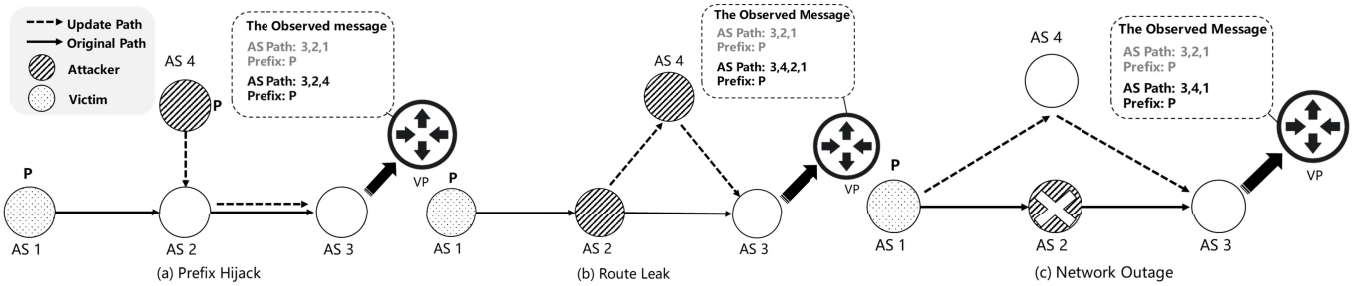

Fig. 2. Examples of each type of BGP anomaly.

Network Outage: An AS exchanges the network-layer reachability information with a neighboring AS by building BGP to keep alive messages within the hold time. However, due to misconfiguration or other issues like natural disasters or political events, the ASes are more likely to disconnect from their peers, leading to unreachable paths and even disconnections with the other ASes. In Fig. 2(c), AS2 is disconnected from its neighbors AS1 and AS3, causing the path (AS3, AS2, AS1) to be unreachable. Therefore, AS1 has to select an alternative path to guarantee a reachable path to prefix P.

2) Observations: The above examples show some obvious irregular changes as follows when the anomaly events occur.

- BGP anomaly events significantly affect the logical topology of AS. Different types of anomalies lead to distinct changes in the AS topology. These changes in topology have consequences, such as modifying AS's k-hop neighbors and introducing noticeable geographical and topological detours in the paths to prefixes.
- In the context of BGP anomalies, it is essential to note that if the malicious attacks or misconfiguration occurs, the victim AS may not be immediately aware of the negative changes occurring, because these changes typically originate from its surrounding neighbors first, such as the addition of new links or withdrawal of old links. Thus how to evaluate the status of one AS according to the information of its surrounding neighbors could be important.
- The changes of prefixes carried in the AS links can more reflect the anomalous activities of networks.

By exploiting these observations and incorporating the AI knowledge, we can achieve more precise detection of large-scale events from the global view even though under incomplete data.

B. Problem Statement

To illustrate the problem more clearly, we formulate the anomaly detection problem for BGP as follows.

Given the time window size T, the sequence of BGP updates is obtained from the collectors of RIPE NCC[1] or RouteViews,[2] namely, $U^t = [u_i]_{i=t}^{t+T}$, where u_i represents the i-th update message from the starting time t. In a given period, we can get multiple time windows. According to these sliding windows, the output of anomaly detection is to infer whether these series of windows are normal or not, i.e., $\{\tilde{y}^t \in Y | \tilde{y}^t \in C\}$ holds the normal label and other fine-grained BGP anomaly types, where Y is the label vector, and C represents the set of anomaly classes. Additionally, after the fine-grained anomaly detection, the set of ASes that more likely have anomalous behaviors $V_a = \{v_i\}, V_a \subset V$ should be inferred and identified, where V represents the whole set of global ASes. This process is formalized as follows,

$$\begin{aligned} &g(U^t) \rightarrow \tilde{y}^t, \tilde{y}^t \in C, \\ &f(U^t) \rightarrow V_a, V_a \subset V, \end{aligned} \tag{1}$$

where $g(\cdot)$ is a mapping function for identifying the anomaly type y_i of each incoming BGP update sequence $\tilde{u}_i$ and $f(\cdot)$ is used to trace the problematic AS set if anomaly occurs.

There are some different places from other networks. To start with, each AS has its ranking level to represent its importance of AS. Then the link between ASes also have weights. Ultimately, this graph continuously evolves as BGP behaviors change, comprising the numbers of nodes and edges. Therefore, there are some key questions to answer, as follows:

- Q1: How to update the dynamic graph in time for online anomaly detection?
- Q2: How to model this large-scale dynamic networks considering both effectiveness and accuracy?
- Q3: How to mitigate the negative effects caused by limited visibility?

For the convenience of readers, the notation used in this paper is summarized in Table II.

IV. Methodology

In this section, we first illustrate how to use our scheme to incrementally update BGP graph in a timely manner in Section IV-A and construct enriched graph in Section IV-B, and then describe how our methods can achieve accurate detection of BGP anomalies and trace the sources in detail in Section IV-C. Our method is open source and available on the Github repository.[3]

A. The Building and Update of BGP Graph

First of all, we transform the global BGP network into a dynamic graph representation $G(t) = (V(t), E(t), F(t), W(t))$, where these variables stand for the following:

[1]https://www.ripe.net/

[2]https://www.routeviews.org/routeviews/

[3]https://github.com/wuzheng1994/BGPgraph.git

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

TABLE II
LIST OF NOTIONS

Notion	Meaning
C	The class set comprising normal, and three anomaly types.
$E(t)$	The edge set of the graph at the timestamp t.
$F(t)$	The node attribute matrix of the graph at the timestamp t.
$G(t)$	The graph modeling by the global ASes at the timestamp t.
I	The split interval of bins function using for encoding edge attribute.
R_p^t	The route set of prefix p in BGP route information table R at the time stamp t.
T	The size of sliding window.
$W(t)$	The edge weight matrix of the graph at the timestamp t.
Y	The label sequence, i.e., $[y_0, \cdots, y_N]$.
U^t	The update sequence constructed by the update set within the time window t.
BCE	Binary Cross Entropy.
BN	Batch Normalization.
FCN	Fully-Connected Network.
MLP	MultiLayer Perception.

- $V(t) = \{v_i\}_{i=1}^N$ is the node set at time t and each node stands for a AS in global networks, where N denotes the number of ASes;
- $E(t) = \{e_{ij}\}$ is the edge set at time t, where e_{ij} denotes an edge between nodes v_i and v_j;
- $F(t) \in \mathbb{R}^{N\times K}$ is the node attribute matrix at the tth timestamp composed of the representation of each node where K is the number of node attributes;
- $W(t) \in \mathbb{R}^{C\times B}$ is the edge attribute matrix at the tth timestamp composed of the representation of each edge, where C and B is the number of edges and the dimension of edge attribute respectively.

In prior works [25], [26], a full update of the BGP graph was made based on the complete RIB, which requires significant time to update the graph and calculate the attributes, especially for large-scale graphs. As a result, this approach is not well-suited for BGP anomaly detection.

Compared to full update, in incremental update mechanism, it only needs to infer the changes of links and then modify the AS graph according to these changes and last snapshot of topology, which is critical for perceiving the incremental changes. In this paper, we focus on BGP update messages related to prefix announcement or withdrawal that are driven by a series of BGP router decisions. These decisions depend on the route export policies derived from the contractual relationship between ASes. According to these decisions, we can further infer whether there is a route link between ASes for a prefix. We make a simple and practical assumption that the routers always select the best path to substitute the old path according to their route policy, and announce this update message to their neighbors [19]. It is noted that we do not make specific assumptions about the router policy for the universality of our method. Considering all the update behaviors related to the prefix change, we induce three rules to accurately capture the update behaviors and modify the evolving graph.

RULE 1: Explicit withdrawal: When an withdrawal update message is received with the path to the prefix in RIB not being null, it denotes that the routing information in the local RIB of the sending AS has been removed. Therefore, the links appearing in the AS path need to be eliminated from the graph. Formally, when a update message u_i^t ($u_i^t \in U^t$) coming, we define the explicit withdrawal as:

$$(u_i^t.operation == W) \wedge (R_p^t \neq \emptyset), \tag{2}$$

where R_p^t denotes the route set of prefix p at the timestamp t, and $u_i^t.operation == W$ represents that the operation of the i-th update is withdrawal.

RULE 2: Explicit announcement: If an update message is an announcement and the path to the prefix in the RIB is empty, this is an explicit announcement to indicate that there is no path to the prefix before, thus the announced AS path is the best route and should be added. It indicates that there is no route to the prefix. Formally, it is defined that

$$(u_i^t.operation == A) \wedge (R_p^t = \emptyset), \tag{3}$$

where $u_i^t.operation == A$ denotes the update message u_i^t is an announcement BGP update. Therefore, the edges in $u_i.as\_path$ could be added in the AS topology $G(t)$.

RULE 3: Implicit withdrawal: If an update message is an announcement and the path to the prefix in RIB is not empty, it means that either the original AS changes its route preference or some downstream AS on the AS path changes its export policies. Formally,

$$(u_i^t.operation == A) \wedge (R_p^t \neq \emptyset \wedge u_i^t.\text{as}\_path \notin R_p^t). \tag{4}$$

Therefore it should remove the edges in the previous AS path R_p^{t-1} from the graph $G(t-1)$, and add the edges in the current AS path R_p^t, i.e., $u_i.as\_path$.

Fig. 3 shows the incremental update mechanism. The raw data from BGP collectors include BGP RIBs and update messages. We initialize the graph $G(0)$ according to the AS path in the initial RIB R^0. Then we update the RIB R^t at the t-th timestamp utilizing the sequence of update messages U^t received at the t-th timestamp and R^{t-1} the RIB of last

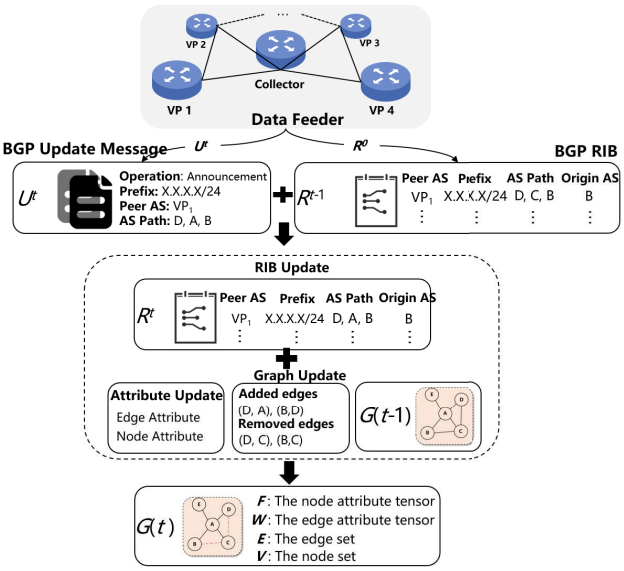

Fig. 3. The incremental update of BGP graph.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

timestamp. At the same time, we convert the graph $G(t-1)$ to $G(t)$ according to the changes of edge sets and the update rules. To reduce the update bias of AS as much as possible, we perform the topology correction using RIB tables at the interval of 8 hours.

B. The Design of BGP Graph

Considering the evolving nature of the global inter-domain network over time, it is crucial to capture the changes in BGP behaviors as comprehensively as possible. To this end, we incorporate the node and edge attributes that reflect these changes into the AS network in the following manner.

First, the node attributes include two distinct parts: the node state information (NSI) and node dynamic information (NDI). NSI includes the long-term state information obtained from authorized databases, comprising the geographic location and AS rank. NDI reflects the real-time information calculation from the graph of each snapshot, including the AS degrees, PageRank values of ASes, and the top values of the degree and PageRank of neighbors. Below, we provide detailed descriptions of these attributes.

The geographic location attributes consist of the latitude and longitude values of an AS obtained from geolocation databases. As mentioned earlier, the BGP anomalies often observe detouring behaviors. It proves to be an effective way to capture these detouring behaviors by using the geolocation data [27]. The AS rank collected from Caida[4] is used to reflect the impacts of one AS, which is calculated by the customer cone size and the degree of AS.

The AS degree indicates the number of its neighbors. The PageRank value of AS is well-known for its simple calculations and exemplary performance in the Internet search engines, which can reflect the importance of one AS. In this paper, this value is calculated in the incremental way [28].

We also add the top values of the degree and PageRank within the neighbors of the AS. In large-scale anomaly events, the loss of connections from influential neighbors of the AS may affect the service of AS.

All these node attributes are calculated or fetched from databases directly, thus the time overhead is negligible. The node attribute matrix F can be formulated as:

$$f_i = [f_{i1}, \cdots, f_{iK}]$$
$$F = [f_i]_{i=1}^{N}, \tag{5}$$

where f_i stands for the feature tensor of node AS_i; N denotes the number of nodes.

In addition, the importance of a link can be reflected by the number of prefixes. The more IP addresses carried in the AS link, the more significant the AS link is. However, the calculation of the number of IP addresses for graphs with a large number of prefixes is time-consuming, which can not meet the real-time demand. Therefore, we devise a simple measurement method based on the mask length as follows.

[4]https://www.caida.org/

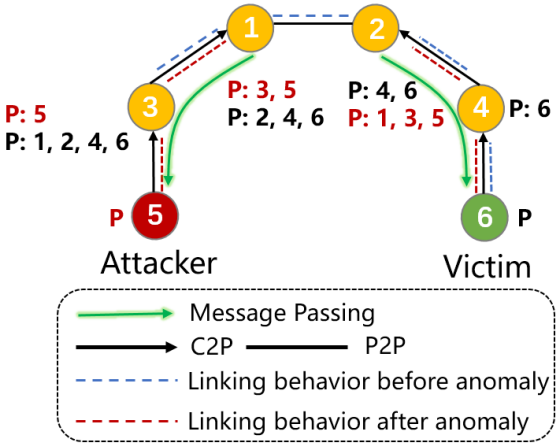

Fig. 4. An example of a AS-level topology and the effect of an BGP anomaly on the message passing of GCN.

The weight of AS link between ASi and ASj is calculated as follows,

$$w_{ij}{}' = \sum_{n \in P} \frac{m_n - l_n}{m_n}, \tag{6}$$

where m_n is the length of the IP prefix n, and m_n is 32 for IPv4 and 128 for IPv6; l_n is the mask length of IP n; P is the IP prefix set that the link (ASi, ASj) carries.

Large value range and variance may lead to the gradient explosion of our model. We further encode the weight w' as:

$$w_{ij} = bins(w', I), \tag{7}$$

where the function $bins(\cdot)$ means that the value range of w is binned averagely and this function returns the indices of the bins. I is the vector of the split intervals.

C. Anomaly Detection Model

Given BGP prefix hijacking as an example, we explain how to perceive the changing state of nodes just using BGP updates in Fig. 4. Before anomaly occurs, all the traffic destined to prefix P is routed to AS6. When AS5 announces the prefix P deceptively and launches a prefix hijack, AS3 and AS1 are more likely to select the false route to AS5 due to the shorter AS path. Thus the changes of linking behaviors of AS3 and AS1 take place. GCN model is known for transmitting and aggregating node information, namely message passing. Furthermore, this kind of AS linking behaviors comprising the addition of new link (AS3, AS5) and the withdrawal of old link (AS1, AS2) could alter the way of message passing of AS nodes in the topology.

To better work with the proposed embedded graph, we present an anomaly detection and cause tracing framework as shown in Fig. 5. The whole model comprises graph learning, graph pooling, and readout modules for anomaly detection, as well as attribute reconstruction and bias computing modules for root cause tracing.

1) Overview: Our GraphBGP method aims to learn the changes of AS-level graph and then identifies the types of BGP anomaly. Besides, if a BGP anomaly occurs, we try to trace the problematic AS.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

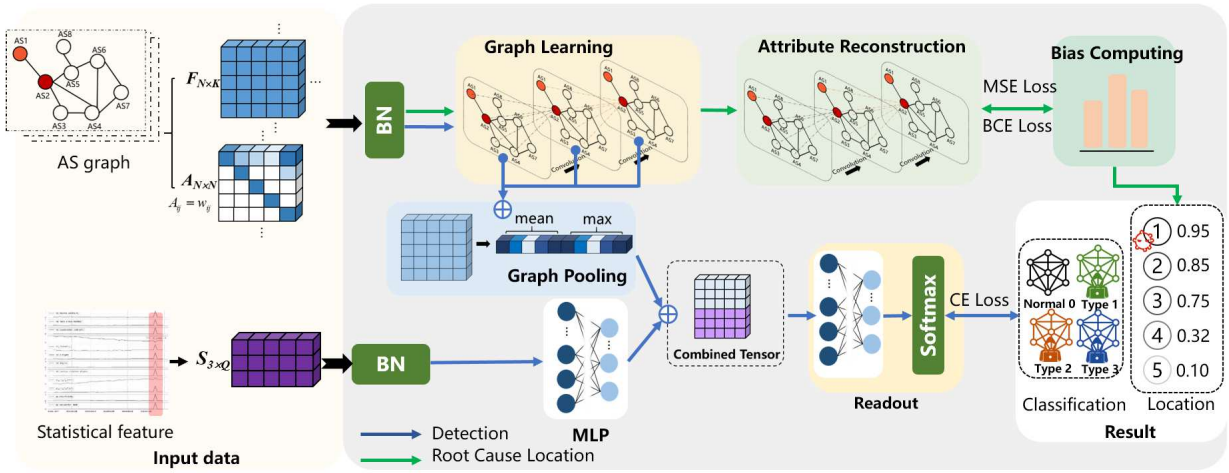

Fig. 5. The framework of the proposed BGP anomaly detection and cause tracing.

To capture the spatial interactions among different ASes, we exploit an M-layer GCN model with a node representing an AS and an edge representing the connection between two ASes. We perform the graph pooling operations on the outputs of each layer to obtain the joint multi-order neighboring graph representation. In addition, we incorporate the BGP behavioral statistical knowledge into this graph and input the combined tensor into readout layers to identify BGP anomalies. Upon detecting the BGP anomalies, a novel auto-encoder framework is designed to reconstruct the node attributes and trace the abnormal ASes using the specially designed loss function and compute the biases. The specific design is as follows.

2) Graph Learning: Considering the AS topology in BGP infrastructures, we combine the corresponding adjacency matrix and attribute matrix into the graph convolution neural network and learn a layer-wise latent representation by the following function.

$$F^{(l+1)} = \sigma\left(D^{-\frac{1}{2}}\tilde{A}D^{\frac{1}{2}}F^{(l)}\theta^{(l)}\right) \in \mathbb{R}^{N\times H}, \tag{8}$$

where D is the degree matrix and A is the adjacency matrix. $F^{(l)}$ is the node attribute tensor of the lth layer. $\tilde{A}$ is self-looped adjacency matrix and $\theta^{(l)} \in \mathbb{R}^{N\times H}$ is the trainable parameters of the lth layer, and H is the size of hidden layers. Note that $\sigma(\cdot)$ is the non-linear activation function, i.e., $\text{relu}(x) = \max(0, x)$.

To enable our framework to perceive the importance of each edge, we configure the adjacency matrix $\tilde{A}$ according to edge attributes and Eq. (9).

$$\tilde{A} = \begin{cases} w_{ij} & if\ e_{ij} = 1 \\ 0 & otherwise. \end{cases} \tag{9}$$

As a result, the detection model could pay more attention to the changes of the prefixes carried in AS links.

3) Graph Pooling: To obtain each layer's graph representation, we use the average and max functions as the pooling functions to generate the graph representation tensor $F^{(l)}$, which has been widely used in the previous works [29]. Mathematically,

$$F^{(l)}_{mean} = \frac{1}{N}\sum_{i=1}^{N} F^{(l)}_{ih} \in \mathbb{R}^{1\times H},$$

TABLE III
LIST OF STATISTICAL FEATURE USED IN ANOMALY DETECTION MODEL

Anomaly Type	Feature
Prefix Hijack	the occurrence of original AS; the occurrence of repeated announcement and withdrawal; the number of pairs of original AS and peer AS; the max occurrence of the pairs of original AS and peer AS; the number of duplicate update message;
Route Leak	the shortening path of update; the ratio of the number of detouring paths and the total number of update message; the occurrence of repeated announcement and withdrawal; the occurrence of the 3rd AS of the path that changes; the ratio of the message with shorter path and the whole number of messages;
Network Outage	the number of different AS; the difference number of the $2nd$ AS in path; the number of different prefixes; the number of implicit withdrawal; the edit distance with the same tuple (peer, prefix);

$$F^{(l)}_{\max} = \max_i F^{(l)}_{ih} \in \mathbb{R}^{1\times H},$$
$$F^{(l)}_{graph} = pooling(F^{(l)}_{ih}) = F_{mean} \oplus F_{\max} \in \mathbb{R}^{1\times 2H}, \tag{10}$$

where F^{l}_{ik} represents the k-th attribute value of the i-th node in the tensor $F^{(l)}$, and the graph representation $F^{(l)}_{graph}$ is calculated by the $F^{(l)}_{\max}$ and $F^{(l)}_{mean}$ after the max and mean pooling.

Given the attributed matrix F, the m-th hop neighborhood of each node can be effectively captured by M-layers graph convolution. To absorb the knowledge of all M neighboring nodes, the output matrixes $F^{(l)}_{graph}$ of M layers are concatenated vertically to get the whole tensor F_{graph} representing the spatial knowledge of each time window as in Eq. (11).

$$F_{graph} = [F^{1}_{graph}, \cdots, F^{M}_{graph}]^T \in \mathbb{R}^{M\times 2H}. \tag{11}$$

4) Feature Combination: The statistical features can reflect anomaly behaviors in general and are often accompanied by a relatively time-saving extraction process. Therefore, we choose to extract some statistical features using a consistent time window size for more accurate detection.

Considering its efficient performance, we use a well-known feature selection method mRMR [30] based on mutual information theory to achieve the top Q statistical features from a BGP-related feature set listed in Appendix Table A1. In

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

this paper, Q is set to 5 by grid search considering both the time cost and accuracy, and the detailed features are listed in Table III.

Therefore, $3 * Q$ features are used to construct the matrix carrying the statistical information on the updates messages and then map it into the same scale with the spatial tensor by stacking simple linear transformations.

After computing the statistical and spatial tensors, the combined tensor F_{com} is obtained by concatenating these two tensors, where softmax$(\cdot)$ is the softmax function used to get the predicted probability.

$$p_i = \text{softmax}(MLP(F_{com})) \tag{12}$$

5) Loss Function: The detection model uses the cross-entropy (CE) loss in Eq. (13) to optimize the parameters of model, which is the widely used in multi-classification problem.

$$L_c = -\frac{1}{N}\sum_{i}^{N}\sum_{c=1}^{C} y_{ic}\log(p_{ic}), \tag{13}$$

where C is the number of classes and N is the number of training samples. y_{ic} and p_{ic} are the label and output logit calculated by the softmax$(\cdot)$ function respectively.

6) Attribute Reconstruction: An AS-level anomaly tracing model is devised to pinpoint the abnormal ASes. This model uses the auto-encoder framework to reconstruct the attributes of each node in the normal status. If an anomaly event takes place, the bias between these two tensors is more likely to be large, and furthermore the nodes with the first k biggest biases are probably problematic ASes or the most affected ASes. The specific process is as follows.

In this paper, we reuse the trained multi-layer GCN in the detection model to as the encoder part of the pre-trained model and use well-trained parameters as the initial values to shorten the training time. The decoder part uses the inverted structure, i.e., the last layer of encoder is devised as the first layer of the decoder module, and so on for reconstructing the node attributes.

To restore the node attribute F of the normal label as accurately as possible, the mean square error (MSE) loss function is employed as follows,

$$\begin{aligned} b_i^t &= \left\| \hat{f}_i^t - f_i^t \right\|_2 \\ l_b &= \sum_{i \in V} b_i \end{aligned} \tag{14}$$

where $\hat{f}_i^t$ is the predicted attribute vector of node i at time t by the decoder model and f_i^t is the input to the encoder of node i at time t. The reconstruction bias b_i is the $l2$-norm between the true value and predicted value. Note that l_b is trained by the only normal instances (i.e., the true label $y_i = 0$ in this paper).

Furthermore, the binary cross entropy (BCE) is used for discriminating the normal and anomaly types instances as follows,

$$l_c = -\frac{1}{N}\sum_{i=0}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i), \tag{15}$$

where $\hat{y}_i$ and y_i are the predicted logit and the true label of instance i in the training set respectively. With the above two losses, the total loss L_t of the tracing model is calculated,

$$L_t = \begin{cases} l_c & if\ y_i \neq 0 \\ l_c + l_b & if\ y_i = 0, \end{cases} \tag{16}$$

D. Bias Computing

When locating the cause of anomalies, it is found that a larger AS usually has a greater loss because it aggregates more bias from its large number of neighbors. To accurately locate the cause, it is necessary to normalize bias as

$$\hat{b}_i = \frac{b_i - \hat{\mu}_i}{\hat{\sigma}_i}, \tag{17}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i$ are the estimated values of the median and the inter-quartile range (IQR2) for node v_i when the corresponding label is normal. The anomalous AS set V_a is obtained by

Algorithm 1 Framework of BGP Anomaly Detection and Root Cause Tracing

Start with:

$\theta_a = (\theta_g, \theta_r)$: the parameters of anomaly detection including graph learning and the remaining parameters

$\theta_t = (\theta_e, \theta_d)$: the parameters of cause tracing comprising encoder and decoder modules

(X_{tr}, Y_{tr}) and (X_{te}, Y_{te}): the training and test datasets of ASes and their corresponding labels

Initialization:

$\theta_a \leftarrow RandInit(|\theta_a|)$, $\theta_d \leftarrow RandInit(|\theta_d|)$; \\ randomly initialize new parameters

$G_{tr} = ConstrGraph(X_{tr})$ \\ construct graphs for the training data

Training:

$p_i \leftarrow Detector(G_{tr}, Y_{tr}, \theta_a)$

$\theta_a^* \leftarrow \arg\min_{\theta_a} \left(L_c = -\frac{1}{N}\sum_{i}^{C}\sum_{c=1} y_{ic}\log(p_{ic}) \right)$

$\theta_d \leftarrow \theta_g^*$; \\ pre-train the encoder part

$\hat{G}, \hat{Y} = Tracer(G_{tr}, Y_{tr}, \theta_t)$

$\theta_t^* \leftarrow \arg\min_{\theta_t}(l_v(G_{tr}, \hat{G}_{tr}) + l_c(Y_{tr}, \hat{Y}))$; \\ train the parameters of Tracer

Training Output:

θ_a^* and θ_t^* \\ output the best parameters of Tracer and Detector

Inference:

$G_{te} = ConstrGraph(X_{te})$

$\hat{o} \leftarrow Detector(G_{te}, \theta_a^*)$; \\ infer the type of anomaly

if $\hat{o} != 0$ **then**

$v_a \leftarrow Tracer(G_{te}, \theta_t^*)$

Inference Output:

$\hat{o}$ and $\hat{t}_{as}$ \\ output the type of anomaly and the anomalous ASes

$$V_a = \{v_i\} = \arg\underset{i}{\text{top-k}}(\hat{b}_i). \tag{18}$$

The whole process of our BGP anomaly detection and location framework is shown in Algorithm 1. The algorithm

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

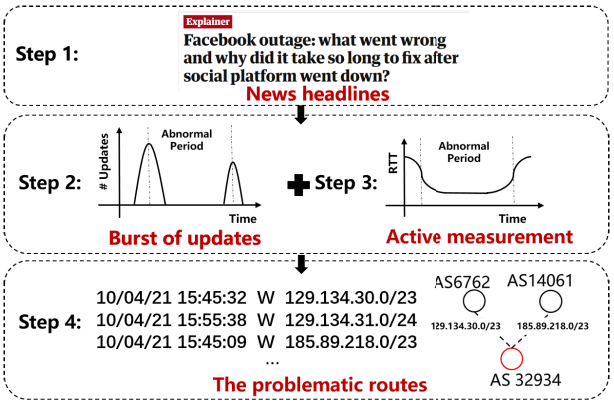

Fig. 6. The process of instance labelling.

is input with each series of BGP updates within the time window and the type of anomaly as well as the potential anomalous AS set are output. In the first place, it needs to update the graph and randomly initialize the parameters of model. Furthermore, it uses the training dataset to optimize the parameters of detector and root cause tracer respectively. Ultimately, each BGP update set is detected within a time window, if the result is an anomalous type, it needs to further trace the root cause of the anomaly.

E. Complexity Analysis

In this subsection, we analyze the time complexity of each component in the GraphBGP framework. For BGP incremental update, the complexity is mainly caused by edge update, which is $O(|U|)$ where $|U|$ is the number of updates within each time window. In the anomaly detection model, GCN brings the complexity of $O(ENH * M)$, where E, N, H and M are the number of edges, nodes, node attributes and GCN layers respectively. Besides, the calculation of statistical features also brings complexity. Because these features can be calculated by multi-process computing, we only need to focus on the maximum computational complexity, i.e., $O(|U|)$. For root cause tracing, it mainly comprises encoder and decoder parts, the time complexity is $O(ENH * M)$. Thanks to this process deployed on GPU, the root cause tracing is fast with multiprocessing. To sum up, the over-all time complexity is $O(2 * |U| + ENH * M)$ if the detection results is normal, otherwise, the time complexity is $O(2 * |U| + ENH * M * 2)$.

V. Experiments

We conduct all the experimental evaluations using the Python language on a server running Linux with configuration of Intel°*ledR* Xeon™ CPU 8352Y@2.2GHZ and an NVIDIA A100 80G graphics card. We implement the proposed method with CUDA 12.0 and PyTorch-Geometric Library.[5]

A. Dataset and Baselines Description

1) Data Collection: The raw data are collected from multiple vantage points from RouteView and RIPENCC.[6] These vantage points are selected by using MVP [31] to maximize the utility of the global collectors. Each collector captures two types of information, i.e., the BGP update packets created every five minutes and the entire RIB obtained every eight hours stored in the format of MRT (Multi-threaded Routing Toolkit).

2) Data Cleaning: To reduce the distracting information existing in raw data, we clean up the BGP routes before making inference. First, the private ASNs appearing in AS path are removed. Second, the aggregated AS paths are decomposed into multiple corresponding paths. Furthermore, we also remove the repetitive ASes on AS paths resulted from AS path prepending. Finally, the ASes owned by IXPs are removed from AS paths, where these IXP ASNs are collected from PeeringDB.[7]

3) Data Labelling: To label each BGP instance as accurately as possible, the whole process of labelling data involves four steps to determine the period of BGP anomalies. First, we collect the authorized headlines to determine a rough period. Then, the bursts of updates are further used to determine the starting and end time. Sometimes the bursts may be not obvious and hidden in the background data. To cope with this problem, the active measurement by probes is used to confirm the anomaly period through RTT (round-trip time) and the number of up/down routes for the involved prefixes using Ark data [32]. Finally, in the above period, we search for the problematic routes in the BGP update message to refine the anomaly time range. The whole process of labelling instances is demonstrated in Fig. 6. The obtained dataset comprises 10 real events with three types of anomalies, i.e., prefix hijack, route leak and network outage for comprehensive experiments. The details of each BGP anomaly event are presented in Table IV.

TABLE IV
The Detailed Descriptions of the Proposed Anomaly Dataset

Event	Type	Start Time	End Time	Message Volume
Google	Route Leak	2018-11-12	2018-11-13	28166180
Verizon	Route Leak	2019-06-24	2019-06-25	11370636
Rogers1	Prefix Hijack	2020-07-31	2020-08-01	77634072
Facebook	Outage	2021-10-04	2021-10-05	127353272
AZURE	Outage	2021-12-16	2021-12-17	49880795
Cablevision	Route Leak	2021-02-11	2021-02-12	16013397
Twitter	Prefix Hijack	2021-03-28	2021-03-28	8120173
Cloudflare	Outage	2022-06-21	2022-06-22	2636896
Rostel	Prefix Hijack	2022-07-26	2022-07-28	192941662
Rogers2	Outage	2022-07-08	2022-07-10	185887189
# Instance	14400	**Data Volume**	315.73 GB	
Ave. Node	73146	**Ave. Edge**	141528	

In addition, the parameters of the proposed GraphBGP are set to the best value using the grid search. More specifically, we set the time window size to one minute, where the grid search process is presented in Appendix. We compare the performances of our proposed method with six state-of-the-art anomaly detection methods based on AI techniques in Table V. To accelerate the training process in parallel, we apply the mini-batch mechanism [33] to optimize our GCN model.

[5]https://pytorch-geometric.readthedocs.io/en/latest/

[6]https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris/archive/ris-raw-data

[7]https://www.peeringdb.com/

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

TABLE V
THE DETAILS OF THE COMPARED METHODS

Method	Alert & Classification	Location	Key Techniques
Graph Feature [34]	✓	×	Graph Feature + SVM[1]
Statistical Feature [13]	✓	×	Statistical Feature + kNN[2]
Self-Operate [35]	✓	×	Statistical Feature + Self-attention + LSTM
MSLSTM [14]	✓	×	Wavelet Transform + LSTM
MultiView [17]	✓	×	Statistical Feature + GAT
BGPVector [36]	✓	×	Skip Gram + CBOW[3]
PoiRoot	✓	✓	Path changes + Activate measurement
GraphBGP	✓	✓	GCN + Node-wise encoding

[1] SVM: Support Vector Machine;
[2] kNN: k-Nearest Neighbors;
[3] CBOW: Continuous Bag Of Words.

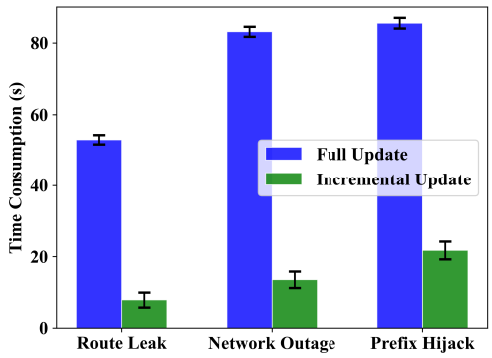

Fig. 7. The average time comparison between full and incremental update.

4) Performance Metric: To evaluate the performances of the proposed method and the baselines, we employ F1-score (F1), precision, and recall scores as the metrics, which are widely used in evaluating the anomaly detection performance. We also assess the time consumption for the BGP anomaly detection and cause tracing. In addition, to measure the timeliness of anomaly detection, we calculate the average early detection time n, where n indicates that the anomaly is detected in the n-th time window after the anomaly event occurs. In addition, the training and test datasets are split by five-fold cross validation to guarantee the credibility of the experimental results.

B. Performance of Graph Update

Fig. 7 compares the time taken for full graph updates and incremental graph updates for three types of events: route leak, network outage, and prefix hijack. Because the proposed method makes modifications only on the corresponding changed part of the graph, it can be observed that the time consumption by the proposed incremental method can be reduced to 10% ~ 30% of that required by the full update.

C. Performances of Anomaly Detection

1) Overall Performance Evaluation: In the training and validation processes as shown in Fig. 8(a), there are rapid drops of losses in the early stage, suggesting that GraphBGP has a good convergence performance during the training period. The confused matrix is shown in Fig. 8(b). This result proves that the proposed method can identify each type of BGP anomaly with an accuracy above 95%.

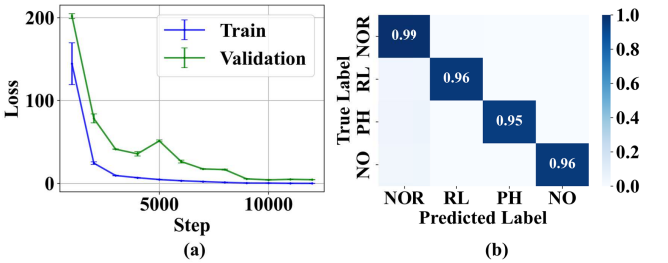

Fig. 8. The performances of anomaly detection using *GraphBGP*. (a) The process of training and validation; (b) The confusion matrix of classification results (NOR: Normal; RL: Route leak; PH: Prefix hijack; NO: Network outage).

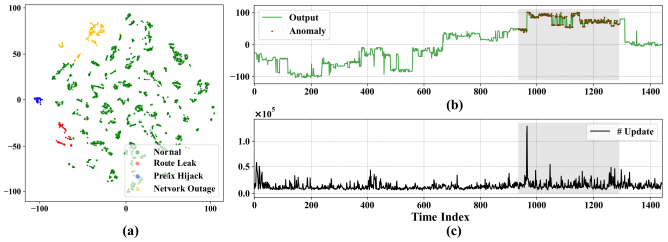

Fig. 9. The t-SNE plot of the readout layer of our model. (a) 2-D compressed representation of BGP anomaly type using t-SNE; (b) 1-D compressed output by t-SNE in the Facebook event where the anomaly period is shown with grey shadow; (c) The number of BGP updates messages in the Facebook event.

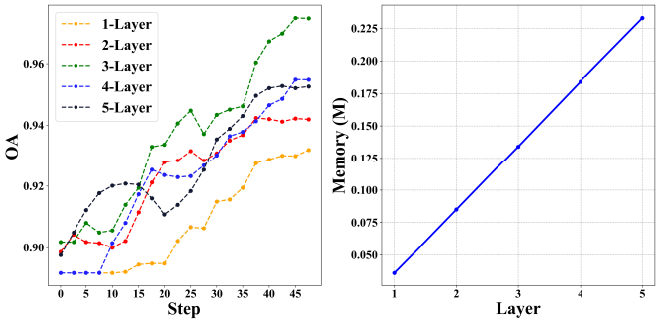

Fig. 10. The performances of OA (left) on validation and occupied memory (right) with different GCN layer.

Fig. 9(a) shows a t-SNE [37] plot with 2 dimensions. The disparity of different types of samples and the similarity of the same type denote that *GraphBGP* can extract discriminative representation for identifying different types of anomaly events. In Fig. 9, we take the Facebook event as the test dataset and present the temporal outputs of the readout layer. *GraphBGP* can detect the anomaly throughout the events because *GraphBGP* captures the changes from both statistical and spatial views, while it is difficult to detect the negative changes in topology using only statistical features, e.g., the number of updates as shown in Fig. 9.

2) Parameter Selection: The layers of GCN reflect the size of the perceptive field in the graph, so the parameter M is important to our model. To select the proper number of layers of GCN, we uniformly use the same number of epochs 50 to train the model, considering both OA and memory occupied by the prediction model. The obtained results are shown in

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

TABLE VI
ABLATION ANALYSIS

Multi-order Perception	Edge Feature	Statistical Feature	OA
✓	✓	✓	0.990
✓	✓		0.984
✓		✓	0.965
	✓	✓	0.973
✓			0.959
	✓		0.960
		✓	0.955
			0.950

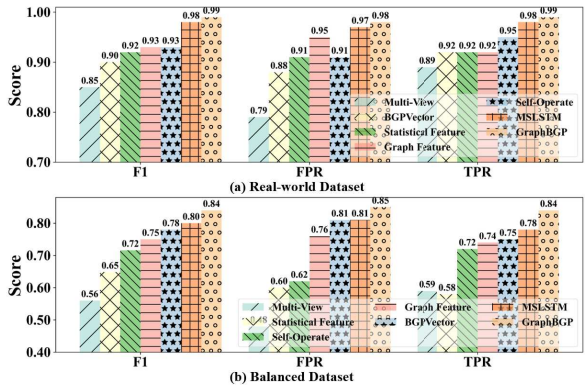

Fig. 11. Performances of our methods and its baselines.

Fig. 10. We can observe that when the number of layers is three, the detection model gets the best performance. It may be because the combination of three layers of perspective can obtain the most accurate information, including the local (the 1st, the 2nd layers) and wider perceptual field (the 3rd layer) information. With the increase of layers, some negative factors will be introduced, such as the redundant information and the rising memory occupied by model. The presence of redundant information leads to a decrease in both training efficiency and accuracy of detection model, while the rising occupied memory increases the deployment difficulty. In this paper, we apply the GCN model with 3 layers to carry out the following experiments.

3) Ablation Analysis: The ablation analysis is shown in Table VI. Firstly, the original GCN model could obtain a good score on BGP anomaly detection, which shows an above-average detection performance. We also observe that the edge attribute are most helpful for improving the model performance, suggesting that it is conducive to capture the anomaly behaviors. The multi-order neighboring perceptions and statistical representation also help enhance the detection accuracy compared to using the original GCN.

D. Method Comparisons

1) Accuracy Comparison: We compare the performance using both the raw dataset and the balanced dataset, and the latter is adjusted to have the equal number of normal and abnormal instances for each event. The fine-grained classification of each anomaly type is more complex than the binary classification of normal and abnormal types. The balanced dataset amplifies the impact of different types of anomalies. Fig. 11 shows the comparison results.

TABLE VII
THE COMPARISON OF TIME PERFORMANCES

Method	Early Detection	Inference Time(s)	Tracing Time(s)
Graph Feature	2.56	0.056	×
Statistical Feature	2.20	0.085	×
Self-Operate	4.00	0.027	×
MSLSTM	0.25	0.028	×
MultiView	4.00	0.062	×
GraphBGP	**0.00**	**0.017**	**0.007**

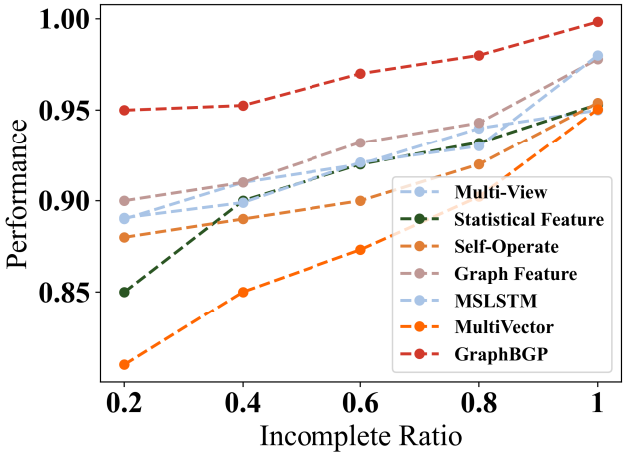

Fig. 12. The performance comparison of each method under different incomplete ratios.

According to Fig. 11, itGraphBGP has the highest detection accuracy in both datasets. The improvement of performance compared to the second best is more obvious on the balanced datasets. Moreover, the high recall and precision rates suggest good performances of our method on every anomaly type. However, the other methods can not detect these fine-grained AS-level behaviors by simply relying on the statistical information of the updated behaviors or the graph measures.

2) Detection Time Comparison: Table VII shows that the proposed *GraphBGP* has the shortest inference time and the earliest detection time, because the proposed method is able to perceive the anomaly through multiple neighbors using the GCN-based model. In addition, in our method, cause tracing can be employed almost seamlessly after detecting the BGP anomaly, only taking 0.007 second.

3) Methods Under Incomplete Information: To evaluate the performance of each method, we make sampling on BGP update messages within each time window randomly with the sampling ratios of 0.2, 0.4, 0.6, 0.8 and 1, respectively. The sampling ratio r means that the r percentage of update messages is randomly selected and retained, so a smaller r denotes the more incomplete information in each time slot. We use the well-trained models of each method with the complete updates, i.e., $r = 1$ to evaluate the performances under the incomplete information with different degrees. The specific results are shown in Fig. 12. Compared with other methods, our proposed method has a more stable performance because it can extract the spatio-temporal stable information and a large

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

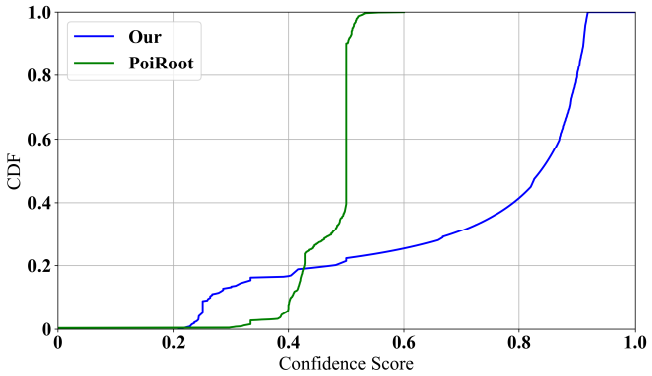

Fig. 13. The confidence comparison of root location.

perceptive field of neighbors. The other methods are affected by the incomplete information negatively to different degrees.

4) Root Location Comparison: To validate our location method, we further conduct a comparative experiment between our approach and PoiRoot. We calculate the root cause confidence as

$$C_t = \frac{C_{alert}(t_a, t)}{C_{sum}(t_a, t)}, \tag{19}$$

where $C_{alert}(t_a, t)$ and $C_{sum}(t_a, t)$ are the numbers of accurate and summation alerts respectively when anomaly occurs inside the time window t_a to t. A bigger value means a better root location performance.

We conduct experiments on the real-world dataset and label the attacker and victim as the root cause of an anomaly event. The confidence score of the cumulative distribution function (CDF) is calculated in the way shown in Fig. 13. It is observed that the scores of our method are mainly distributed from 0.6 to 0.9, higher than its rival PoiRoot. This result demonstrates that our method behaves better than PoiRoot in terms of root location.

E. Case Study

1) Facebook Outage: Facebook and its other platforms, including Instagram, WhatApp and Messenger went down globally for nearly six hours on October, 4, 2021. According to the statement issued by Facebook, this network outage was due to a misconfiguration on the backbone routers, which led to the disappearance of AS paths to Facebook. In this subsection, we study this serious network outage from the BGP perspective by using our GraphBGP method.

We train the model using the proposed dataset excluding the data of Facebook outage, and test the updates of each time window of Facebook outage.

GraphBGP considers the ASes with the top reconstruction biases to be problematic. Fig. 14 shows the time sequence of Facebook (AS32934) and Cogent (AS174) ASes' bias values. A bigger AS tends to get more severe losses aggregated from their large number of neighbors, leading to false location results. To alleviate this adverse effect, we normalize the reconstruction bias using the value of median and the second interquartile range (IQR2). AS174, one of the most impacted neighbors of AS32934, is thus taken as an example. Before

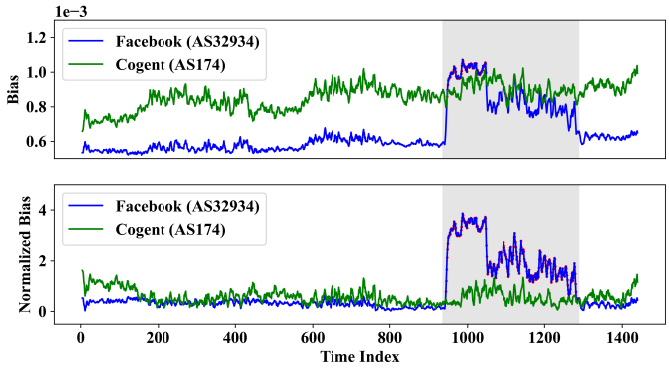

Fig. 14. Time sequence of bias and normalized bias of Facebook and Cogent AS in Facebook event. The area with the grey shadow is the period of anomaly event, and the red points are plotted when AS32934, the anomalous AS of Facebook, is determined as misbehaved AS.

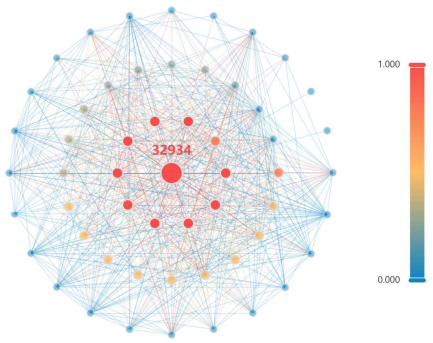

Fig. 15. The ego-subgraph of ASN32934 where the color and size of nodes both denote the bias loss of each node.

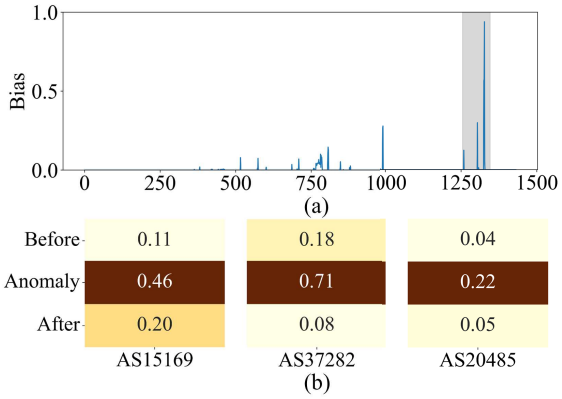

Fig. 16. The results of Google's route leak. (a) The reconstruction bias computed by our proposed method where the shadow area denotes the interval within which the anomaly occurred; (b) The heatmap shows the average reconstruction bias of suspicious ASes respectively before, during, and after the event.

normalization, during most time of this anomaly event, the reconstruction bias of the anomaly AS32936 is lower than that of the AS174, indicating that the true abnormal AS can not be identified at most time. After the normalization, it is noted that this negative impact has been significantly alleviated.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

To demonstrate the performance of locating the anomalous ASes, we further analyze the reconstruction bias of global AS when the anomaly takes place. The ego-graph of AS32934 with the bias loss is plotted in Fig. 15. It is easy to find that AS32934 has the most considerable bias, and its neighbors also have the reconstruction bias in different degrees when the BGP anomaly occurs. For other BGP events in the dataset, GraphBGP can trace all the anomalous ASes accurately when k is set to 10.

2) Google Route Leak: Some of Google's major services unexpectedly went off for a period of time on Nov. 12, 2018. Referred to the News [38], ISP AS37282 leaked 212 Google's (AS15169) prefixes, covering a vast scope of Google services.

The results of the proposed method are shown in Fig. 16. We can observe that the biases during the events are significantly higher than the normal time. Additionally, the number of misbehaved ASes is higher than the other ASes when the event occurs, indicating that our method can accurately locate the problematic ASes.

VI. Conclusion

In this work, we propose a larger-scale BGP anomaly detection method based on dynamic graph learning. This method tries to model the update behaviors from the global topology and statistical perspectives, so as to well capture the behavioral patterns of BGP anomaly, even though under the incomplete visibility of the BGP networks. Specifically, the global AS-level graph is built by customizing the edge and node attributes, which can adapt to the dynamic BGP routes with an incremental graph update in a timely manner. Furthermore, the GCN-based BGP detection model is specially designed for the proposed AS-level graph. To locate the root causes of the BGP anomaly event, the cause tracing model reuses the GCN part of the detection model to carry out seamlessly cause tracing after detecting anomalies and timely identify the ASes with problems. Experiments reveal that the proposed method outperforms the baselines in terms of the detection accuracy, and can detect the anomaly event in the first time window. Besides, the anomaly AS can also be accurately traced within the interval of 0.007s after the BGP anomaly detection.

A normal route change may cause the false positive instances in our proposed system. To reduce this, we can update the model on a regular basis so that normal routing changes can be learned. In addition, we will investigate the cause of positive false in depth to make more accurate anomaly detection in the future.

References

[1] T. Krenc, R. Beverly, and G. Smaragdakis, "AS-level BGP community usage classification," in *Proc. 21st ACM Internet Meas. Conf.*, Nov. 2021, pp. 577–592.

[2] Q. Li, M. Xu, J. Wu, X. Zhang, P. P. C. Lee, and K. Xu, "Enhancing the trust of Internet routing with lightweight route attestation," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 691–703, Apr. 2012.

[3] B. Al-Musawi, P. Branch, and G. Armitage, "BGP anomaly detection techniques: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 377–396, 1st Quart., 2017.

[4] Wikipedia.*2021 Facebook Outage*. Accessed: Jul. 2025. [Online]. Available: https://en.wikipedia.org/wiki/2021-Facebook-outage

[5] T. Chung et al., "RPKI is coming of age: A longitudinal study of RPKI deployment and invalid route origins," in *Proc. ACM Internet Meas. Conf. (IMC)*, New York, NY, USA, 2019, pp. 406–419.

[6] T. Hlavacek, P. Jeitner, D. Mirdita, H. Shulman, and M. Waidner, "Behind the scenes of RPKI," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2022, pp. 1413–1426.

[7] M. Cheng, Q. Xu, W. Liu, Q. Li, and J. Wang, "MS-LSTM: A multi-scale LSTM model for BGP anomaly detection," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–6.

[8] P. Sermpezis et al., "ARTEMIS: Neutralizing BGP hijacking within a minute," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2471–2486, Dec. 2018.

[9] R. Oliveira, D. Pei, W. Willinger, B. Zhang, and L. Zhang, "The (In)Completeness of the observed Internet AS-level structure," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 109–122, Feb. 2010.

[10] T. Krenc, R. Beverly, and G. Smaragdakis, "Keep your communities clean: Exploring the routing message impact of BGP communities," in *Proc. 16th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2020, pp. 443–450.

[11] J. Schlamp, R. Holz, Q. Jacquemart, G. Carle, and E. W. Biersack, "HEAP: Reliable assessment of BGP hijacking attacks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 6, pp. 1849–1861, Jun. 2016.

[12] L. Qin, D. Li, R. Li, and K. Wang, "Themis: Accelerating the detection of route origin hijacking by distinguishing legitimate and illegitimate MOAS," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 4509–4524.

[13] K. Hoarau, P. U. Tournoux, and T. Razafindralambo, "BML: An efficient and versatile tool for BGP dataset collection," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[14] M. Cheng, Q. Li, J. Lv, W. Liu, and J. Wang, "Multi-scale LSTM model for BGP anomaly classification," *IEEE Trans. Services Comput.*, vol. 14, no. 3, pp. 765–778, May 2021.

[15] P. Mahadevan et al., "Lessons from three views of the Internet topology," 2005, *arXiv:cs/0508033*.

[16] Z. Jin, X. Shi, Y. Yang, X. Yin, Z. Wang, and J. Wu, "TopoScope: Recover AS relationships from fragmentary observations," in *Proc. ACM Internet Meas. Conf.*, Oct. 2020, pp. 266–280.

[17] S. Peng et al., "A multi-view framework for BGP anomaly detection via graph attention network," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109129.

[18] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating Internet routing instabilities," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 205–218, Aug. 2004.

[19] U. Javed, I. Cunha, D. R. Choffnes, E. Katz-Bassett, T. Anderson, and A. Krishnamurthy, "PoiRoot: Investigating the root cause of interdomain path changes," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 37–44, Oct. 2013.

[20] O. S. Alkadi, N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "An ontological graph identification method for improving localization of IP prefix hijacking in network systems," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1164–1174, 2020.

[21] D. Kreuzer, D. Beaini, W. L. Hamilton, V. Létourneau, and P. Tossou, "Rethinking graph transformers with spectral attention," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 21618–21629.

[22] Y. Liu et al., "Anomaly detection in dynamic graphs via transformer," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12081–12094, Dec. 2023.

[23] X. Ma et al., "A comprehensive survey on graph anomaly detection with deep learning," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12012–12038, Dec. 2023.

[24] K. Sriram, D. Montgomery, E. O. D. McPherson, and B. Dickson, *Problem Definition and Classification of BGP Route Leaks*, document RFC 7908, 2016.

[25] K. Hoarau, P. U. Tournoux, and T. Razafindralambo, "Suitability of graph representation for BGP anomaly detection," in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 305–310.

[26] K. G. Leyba, J. J. Daymude, J.-G. Young, M. E. J. Newman, J. Rexford, and S. Forrest, "Cutting through the noise to infer autonomous system topology," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 1609–1618.

[27] P. Winter, R. Padmanabhan, A. King, and A. Dainotti, "Geo-locating BGP prefixes," in *Proc. Netw. Traffic Meas. Anal. Conf. (TMA)*, Jun. 2019, pp. 9–16.

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.

[28] B. Bahmani, A. Chowdhury, and A. Goel, "Fast incremental and personalized PageRank," 2010, *arXiv:1006.2880*.
[29] Z. Zhang et al., "Hierarchical graph pooling with structure learning," 2019, *arXiv:1911.05954*.
[30] O. A. Alomari, A. T. Khader, M. A. Al-Betar, and M. A. Awadallah, "A novel gene selection method using modified MRMR and hybrid bat-inspired algorithm with β-hill climbing," *Int. J. Speech Technol.*, vol. 48, no. 11, pp. 4429–4447, Nov. 2018.
[31] T. Alfroy, T. Holterbach, and C. Pelsser, "MVP: Measuring Internet routing from the most valuable points," in *Proc. 22nd ACM Internet Meas. Conf.*, Oct. 2022, pp. 770–771.
[32] CAIDA. *The CAIDA, UCSD IPv4 Routed /24 Topology Dataset*. Accessed: Sep. 2007. [Online]. Available: https://www.caida.org/catalog/datasets/ipv4_routed_24_topologydataset/
[33] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch geometric," 2019, *arXiv:1903.02428*.
[34] J. Huang, M. Odiathevar, A. Valera, J. Sahni, M. Frean, and W. K. G. Seah, "Realtime BGP anomaly detection using graph centrality features," in *Proc. Int. Conf. Adv. Inf. Netw. Appl.* Cham, Switzerland: Springer, 2024, pp. 222–233.
[35] Y. Dong, Q. Li, R. O. Sinnott, Y. Jiang, and S. Xia, "ISP self-operated BGP anomaly detection based on weakly supervised learning," in *Proc. IEEE 29th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2021, pp. 1–11.
[36] T. Shapira and Y. Shavitt, "AP2Vec: An unsupervised approach for BGP hijacking detection," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 3, pp. 2255–2268, Sep. 2022.
[37] L. van der Maaten and G. E. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[38] Mutually Agreed Norms for Routing Security.*Route Leak Causes Major Google Outage*. Accessed: Nov. 2018. [Online]. Available: https://www.internetsociety.org/blog/2018/11/route-leak-caused-a-major-google-outage/

Authorized licensed use limited to: SUNY AT STONY BROOK. Downloaded on January 09,2026 at 07:41:37 UTC from IEEE Xplore. Restrictions apply.