This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



# A Cooperative Vehicle-Road System for Anomaly Detection on Vehicle Tracks with Augmented Intelligence of Things

Yuxin Zhang, Limei Lin, Yanze Huang, Xiaoding Wang, Sun-Yuan Hsieh, *Fellow, IEEE*, T Gadekallu, and Md. Jalil Piran, *Senior Member, IEEE*

***Abstract*—The Augmented Intelligence of Things (AIoT) is an emerging technology that combines augmented intelligence with the Internet of Things (IoT) to facilitate advanced decision-making processes. In this paper, we focus on the detection of vehicle trajectory anomalies in a vehicle-road collaboration system by AIoT, aiming to improve the traffic safety and road operation efficiency. We transmit collaboration data collected by sensors to an IoT server, which enables the effective data analysis for vehicle trajectory information. We propose a self-supervised learning augmented intelligence algorithm to achieve precise and efficient detection of trajectory anomalies. First, we models the traffic road network as a topology graph. Subsequently, we sample the relevant subgraph contexts for each target node through a random walk algorithm. And the subgraphs with higher intimacy scores are selected as the contextual background to be input along with the target node. After that, the anomaly score of each target node is computed through the generative learning module and the contrastive learning module. To evaluate the effectiveness of our anomaly detection approach, we initially conduct pre-training of the model using four widely utilized graph machine learning datasets. The experimental results reveal that our approach surpasses related methods in terms of the accuracy of identifying graph anomaly nodes. In addition, we carry out our proposed approach on two real traffic datasets with high accuracy of 86.47% and 85.2%, respectively. The results of this study demonstrate our proposal's effectiveness in detecting trajectory anomalies in real-life traffic scenarios, demonstrating the effectiveness of our approach.**

***Index Terms*—Vehicle road collaboration, Augmented Intelligence of Things, Anomaly Detection, Self-supervised Learning.**

(Corresponding authors: L. Lin, M. J. Piran)

Y. Zhang and L. Lin are with the College of Computer and Cyber Security, Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350117, China. E-mails: qsx20221353@student.fjnu.edu.cn, linlimei@fjnu.edu.cn.

Y. Huang is with the School of Computer Science and Mathematics, Fujian University of Technology, Fuzhou 350118, China. E-mail: yzhuang@fjut.edu.cn.

X. Wang is with the College of Computer and Cyber Security, Fujian Provincial Key Laboratory of Network Security and Cryptology, Fujian Normal University, Fuzhou 350117, China. He is also with the National Key Laboratory for Tropical Crop Breeding, Institute of Tropical Bioscience and Biotechnology, Chinese Academy of Tropical Agricultural Sciences, Sanya 572024, China. E-mails: wangdin1982@fjnu.edu.cn.

Sun-Yuan Hsieh is with the Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan 701, Taiwan. E-mail: hsiehsy@mail.ncku.edu.tw.

T. Gadekallu is with Division of Research and Development, Lovely Professional University, Phagwara, and Center of Research Impact and Outcome, Chitkara University, Punjab, India. email: thippareddy@ieee.org.

M. J. Piran is with the Department of Computer Science and Engineering, Sejong University, Seoul 05006, South Korea. Email: piran@sejong.ac.kr.

## I. Introduction

THE development of Intelligent Transportation Systems (ITS) makes road safety and traffic efficiency a top priority in urban planning and traffic management [1]. Effectively detecting and identifying abnormalities such as traffic congestion, accidents, and road damage is crucial for maintaining the integrity and smooth operation of the transportation network. Vehicle-road collaboration systems, which allow vehicles and infrastructure to exchange real-time data, attract widespread attention. These systems first capture information from sensors deployed on vehicles and infrastructure, and then transmit the data to IoT data centers for in-depth analysis and to effectively respond to road anomalies [2], [3]. This collaborative environment not only offers the possibility of making the most of the data collected, but also creates an excellent opportunity to ensure traffic safety and improve the efficiency of road operations.

On the other hand, the rapid development of the Internet of IoT and Artificial Intelligence (AI) makes remarkable achievements in various fields [4], and this trend leads to the convergence of a variety of information technologies, such as computers, electronics, and automation, bringing about tremendous changes and opportunities for modern society. Against this backdrop, the Augmented Intelligent Internet of AIoT emerges as a key approach to achieving comprehensive intelligent network coordination and management. It combines IoT technology with augmented intelligent algorithms, particularly machine learning algorithms, to significantly enhance data processing and analysis capabilities [5]. With the rapid development of vehicle-road collaboration technology, the application of AIoT in the field of transport also attracts widespread attention. As shown in Fig. 1, in practical applications, augmented intelligence methods such as deep learning are usually used to perform data analysis and pattern recognition on traffic data collected by large-scale IoT devices to achieve intelligent traffic management and decision support. However, traditional methods typically use a grid-based approach to model traffic data, dividing the traffic network into regular grid cells and computing attributes such as average speed or flow rate in each cell. The limitation of this approach is that many real transport networks (e.g., complex road networks and subway networks) are inherently graph-structured. Consequently, conventional grid methods may encounter difficulties when dealing with complex road structures and irregular traffic flows [6].

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023




Fig. 1. Vehicle-road collaboration system applying AIoT. This is achieved by using wireless communication devices (e.g., sensors installed on the road) to collect vehicle trajectory information (e.g., average speed and flow rate of passing vehicles), after which the collected data is processed and analysed by augmented intelligence (e.g., deep learning or machine learning).

Roads within a city should not be viewed as isolated entities since changes in traffic volume on one road are influenced by the topology of neighboring road networks. Specifically, the traffic state of an upstream road impacts the downstream road, and the condition of the downstream road reflects the accessibility of the upstream road. This interconnected relationship implies that traffic flow within the network is influenced not only by the attributes of individual roads but also by the traffic conditions of neighboring roads [7]. The interrelated nature of traffic flow within the network introduces significant complexity to its propagation and dynamics. To capture this complexity and interactivity more accurately, we model it as a topology graph rather than treating it as separate entities such as grids or line segments [8]. This graph-based representation enables the incorporation of traffic information as node attributes, along with the contextual factors mentioned above. Subsequently, we can employ anomaly detection algorithms on the graph to detect and address road anomalies within the traffic network.

In recent years, significant advancements are being made in anomaly detection of graphs, particularly through methods like Graph Convolutional Neural Networks (GCN) [9]. GCN is a deep learning model based on graphs, which can process graph data efficiently and learn the representation of nodes from the network. Through information propagation and fusion, GCN captures the contextual associations within a graph and incorporates information from neighboring nodes during the representation process to capture complex dependency relationships between nodes. This makes GCN a powerful tool for processing traffic data and detecting abnormal behaviour. In addition, in the context of the traffic network, anomaly labels are often scarce, and supervised anomaly detection algorithms suffer from the problems of uneven distribution and inability to cover all anomaly types during the training process, which leads to poor training results. Therefore, unsupervised detection algorithms are more suitable for the practical application of traffic network data.

To address the above problems, this paper proposes a vehicle track anomaly detection method that leverages the collaboration between vehicles and road infrastructure. We employ a self-supervised anomaly detection technique for graphs to detect anomalous nodes. Our approach utilizes the rich sensor data collected by vehicles and combines it with the information provided by the infrastructure to construct a comprehensive representation of the road network, where each node represents a road segment and edges represent the connectedness between nodes. The node attributes encompass various features related to the road segment, such as average speed, vehicle density, or vehicle flow rate. Vehicle trajectory anomalies are also abstracted as node anomalies that differ significantly from the behaviors of other nodes in the graph. To accurately detect anomalies, we construct different environments (subgraphs) around target nodes and employ a self-supervised learning

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



strategy to compare and obtain anomaly scores for each node. Specifically, we first select the subgraphs with high intimacy scores to the target node from the subgraphs of the target node generated by the random walk and input them into the GCN encoder along with the target node as contextual information for the target node input to the GCN encoder to learn the latent representation of each node. After that, we employ two modules, namely Generative Learning with Attribute Reconstruction and Contrastive Learning with Structure Prediction, for anomaly detection. By reconstructing node attributes through the decoder, the generative learning module can capture anomalies in the attribute space. The contrastive learning module captures anomalies in the structural space and also considers the mixture of structural and content information by directly comparing the target node and its surrounding context. Finally, we combine the output of the generative and contrastive modules to compute the anomaly score for each node.

Our proposed anomaly detection algorithm adopts a self-supervised learning approach, eliminating the need for manual data labels during training. This reduction in manual labeling results in lower energy consumption compared to traditional rule-based or manually designed feature methods. Additionally, by incorporating local subgraphs, our algorithm can operate without requiring the entire graph information. This approach significantly reduces computational and communication requirements compared to traditional centralized or global computation methods, thus alleviating the burden on network traffic. By achieving higher efficiency in anomaly detection on real traffic datasets, our algorithm avoids the resource overhead associated with monitoring and analyzing the entire road network. This results in savings in computational and storage resources, ultimately improving overall resource utilization efficiency.

The major contributions of this paper are listed as follows:

- In this paper, a self-supervised learning augmented intelligence algorithm is innovatively used for vehicle trajectory anomaly detection, which applies AIoT technology to the vehicle-road collaboration system. The algorithm effectively analyses vehicle trajectory information using collaborative vehicle and infrastructure data collected by sensors in the system, thereby improves traffic safety and road operation efficiency.
- We model the traffic road network as a topology graph to capture the complex network structure of vehicle tracks and improve the previous self-supervised learning algorithm by introducing the concept of intimacy score. By selecting subgraphs with higher intimacy scores as the contextual background, we provide a more accurate and comprehensive data basis for subsequent anomaly detection and also improve the efficiency of anomaly detection.
- We provide a comprehensive experimental validation of the proposed self-supervised learning augmented intelligence algorithm. Firstly, the model is pre-trained with four widely used graph machine learning datasets, and it is demonstrated that the method outperforms previous methods in the accuracy of identifying graph anomaly nodes. In addition, test results on two real traffic datasets show that the accuracy of the method is as high as 86.47% and 85.2%, which fully demonstrates the effectiveness and practical application value of the algorithm in detecting trajectory anomalies in real traffic scenarios.

The rest of this paper is organized as follows. Section II reviews related work. Section III describes the proposed method. Section IV discusses the experimental results and we conclude the paper in Section V .

## II. Related Work

Augmented Intelligence is widely regarded as an extension of Artificial Intelligence, with a specific focus on enhancing human cognitive abilities rather than replacing them [10]. Artificial Intelligence encompasses a multitude of techniques and methods used to enable machines or computer systems to simulate human intelligence. In contrast, Augmented Intelligence places a stronger emphasis on fostering collaboration between humans and machines to augment human cognition and decision-making. This is achieved by combining the computational power and intelligence of machines with the unique insights and judgement of humans. In the field of the Internet of Things (IoT), the application of augmented intelligence has given rise to the emerging concept of AIoT, effectively addressing a series of problems caused by the proliferation of IoT data. With the help of cloud computing and big data analysis, AIoT not only provides more accurate and personalized services for modern industries [4], but also shows broad application prospects in areas such as network edge collaborative computing. For example, in the study of unmanned aerial vehicles (UAVs) on-board life extension, the application of AIoT has significantly improved the operational efficiency and lifetime of the UAVs [11]. The Vehicle-Road Collaboration field is also benefiting from the deeper application of AIoT technology. By integrating vehicles, road infrastructure, and cloud-based data processing with augmented intelligence, AIoT can facilitate real-time data analysis, decision support, and traffic optimization, leading to a safer and more efficient travel experience for drivers and traffic managers. The combination of AIoT technology and vehicle road collaboration enables the development of a smarter, more efficient, and safer transportation system. Since our goal is to apply self-supervised learning algorithms in augmented intelligence for efficient anomaly detection in a vehicle-road collaboration system, we will describe the background in three related sections of work, namely vehicle-road collaboration system, anomaly detection and self-supervised learning.

### A. Vehicle Road Cooperation System

Vehicle-to-road cooperative systems, through vehicle-to-internet (V2X) technology, have greatly expanded the connectivity capabilities between intelligent vehicles and the external environment, resulting in a revolutionary advancement in the field of intelligent transportation [16]. V2X covers a variety of interaction modes, including vehicle-vehicle (V2V), vehicle-road setup (V2I), vehicle-internet (V2N), and Vehicle-Person (V2P). V2N is the vehicle-vehicle information interaction and

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



TABLE I
COMPARISONS AMONG OUR METHOD WITH EXISTING SELF-SUPERVISED LEARNING METHODS.

| Approach | Subgraph selection | Generative learning | Comparative learning | Multi-view contrast |
|---|---|---|---|---|
| DGI [12] | ✗ | ✗ | ✓ | ✓ |
| MVGRL [13] | ✗ | ✗ | ✓ | ✓ |
| CoLA [14] | ✗ | ✗ | ✓ | ✓ |
| SL-GAD [15] | ✗ | ✓ | ✓ | ✓ |
| Ours | ✓ | ✓ | ✓ | ✓ |

reminder, which can be used to enhance the vehicle collision avoidance coefficient. V2I refers to the interaction between the vehicle and the base setup, through which the V2I can obtain the information from the roadside monitoring devices such as road sensors, climate sensors, traffic lights in front of the driving, barricades, and other road sensing devices. This effectively improves the efficiency and safety of road traffic. V2N allows vehicles to connect to the cloud through mobile networks and use the navigation, monitoring, safety warnings, etc. provided by the cloud. V2P is the interaction between vehicles and pedestrians, which can provide safety warnings to pedestrians and non-motorized vehicles on the road. Notably, Oh et al. [17] proposed an advanced traffic information system based on V2V communication. In this system, the vehicle first passes the traveling data measured by acceleration sensors, and the GPS positioning module, and then detects unsafe driving maneuvers by the designed detection algorithm. The development and application of this system further demonstrate the immense potential of vehicle-road collaboration systems in enhancing road traffic safety and efficiency.

### B. Anomaly Detection

Anomaly detection is a consistent research hotspot [18]. In recent years, as deep learning gains prominence, researchers are gradually shifting from traditional statistical methods to deep learning techniques to solve anomaly detection problems. Although these methods mainly target data in the Euclidean domain, anomaly detection of graph-structured data from outside the Euclidean space is also receiving extensive attention in recent years [19]. How to measure graph anomalies is an important issue in graph anomaly detection, Perozzi and Akoglu [20] proposed the AMEN anomaly detection method, which considers the ego network information of each node, and uses residual analysis for anomaly detection to find anomalous neighbors on the attribute network. After this, deep encoders are also being applied to detect graph anomalies [21], [22]. These methods typically use graph autoencoders to embed nodes into the potential space, reconstruct the graph information, and then detect anomalies using the reconstruction error between the original and reconstructed graph information. Recently, SL-GAD [15] scores nodes to detect anomalies by sampling relevant subgraphs of the target node followed by two self-supervised learning tasks of attribute learning structure learning respectively. However, SL-GAD only uses a random wandering approach when selecting subgraphs for the target nodes, which cannot ensure a high correlation between the subgraphs and the target nodes, which restricts the subsequent acquisition of information from the contextual information of the subgraphs.

### C. Self-supervised Learning

Self-supervised learning (SSL) is a new learning paradigm that aims to train models by automatically generating goals or labels from unlabeled data and using these generated goals. Compared to traditional supervised learning methods, self-supervised learning eliminates the need for manual labeling of extensive data and thus finds wide applications in scenarios where labeled data is scarce. In the past few years, self-supervised learning made significant research progress in areas such as computer vision (CV) [23] and natural language processing (NLP). Recently, SSL is being extended to the graph domain. For example, Deep Graph Infomax (DGI) [12] considers the representation of nodes and graph-level representation vectors as contrasted instance pairs and generates corrupted negative samples, which is the first algorithm based on unsupervised patterns of contrast learning. On the basis of DGI, MVGRL [13] treats the original graph structure and graph diffusion as two different views and proposes a multi-view contrast learning model. CoLA [14], a self-supervised learning method based on localized information, which learns node representations by comparing nodes with their contextual information in graphs to learn node representations by comparing them and calculating anomaly scores.

Based on our proposal, we present an anomaly detection algorithm that leverages augmented intelligence and self-supervised learning to enhance the efficiency of the vehicle-road cooperative system. Table I provides a comparison between our algorithm and existing self-supervised learning methods. Specifically, our algorithm initially samples subgraphs associated with the target nodes and selects those subgraphs that exhibit strong correlations with the target nodes. Subsequently, we employ a combination of generative and comparative self-supervised learning tasks to enable more comprehensive anomaly detection. Notably, the comparative self-supervised learning employs multi-view comparisons to effectively utilize the information embedded within the subgraphs.

## III. METHODOLOGY

The utilization of augmented intelligence for detecting vehicle trajectory anomalies through vehicle-road collaboration has broad applications in intelligent transportation systems. This technology not only enables real-time monitoring and anomaly detection of vehicle trajectories in urban traffic,

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



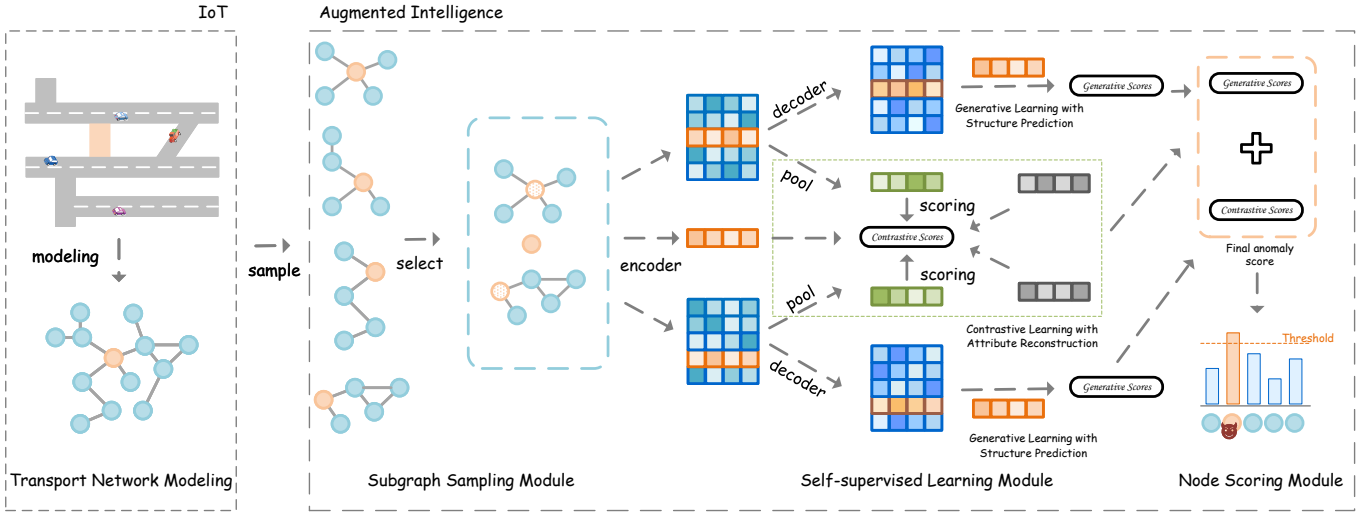



Fig. 2. The general framework of our model. Initially, we represent the road segment and traffic attribute information as a graph, and then several subgraphs are randomly sampled from the original graph. From these subgraphs, we select the ones with high intimacy scores to be used as inputs to the GCN encoder, along with the target nodes. This is followed by the self-supervised learning module, which consists of the generative learning with attribute reconstruction module and the contrastive learning with structure prediction module together, with the contrastive learning module framed by a green rectangle in the middle part. Finally, generated anomaly scores and compared anomaly scores are collected to generate the final anomaly score. According to the given threshold, nodes with an anomaly score greater than the threshold are determined as anomalous.

thereby enhancing the efficiency and safety of traffic management, but also plays a critical role in smart agriculture. For instance, monitoring and detecting anomalies in the trajectories of agricultural vehicles are equally vital. The trajectories of agricultural vehicles in the fields need to be monitored to ensure that tasks such as planting, fertilizing, and spraying are carried out according to plan. By implementing augmented intelligence technology, real-time monitoring and anomaly detection of agricultural vehicles can be achieved, assisting farmers in operating more efficiently in agricultural production, reducing resource wastage, and increasing yields.

In this section, we introduce the general framework of our proposed AIoT algorithm for detecting vehicle track anomalies in traffic network maps. As shown in Fig. 2, our approach consists of five parts, including transport network modeling, subgraph sampling, a self-supervised learning module consisting of generative learning with attribute reconstruction and contrast learning with structure prediction and graph anomaly scoring. First, we model a traffic road network as a topology graph, for the road segment graph modeled as a graph, we use an augmented intelligence anomaly detection method on the graph to detect it. Specifically, for each node to be detected, we first select a number of subgraphs for it using the random walk algorithm, and later select the subgraphs from among them that have high intimacy scores with the node. After that, for the rich node- and subgraph-level information we have taken, inspired by [15], we construct two different self-supervised objectives to detect anomalous nodes, that is, generative learning with attribute reconstruction and constrastive learning with structure prediction. The generative learning with attribute reconstruction aims to reconstruct the feature vector of the target node using neighboring attribute information of the. By doing so, any attribute mismatches between the selected node and its surrounding context can be reflected as a regression loss through a regression loss computed between the reconstructed feature vector and the original feature vector. The constrastive learning with structure prediction, on the other hand, is proposed to more fully utilize the structural information of the input graph, and unlike the node-level objective of generating attribute reconstruction, this hybrid-level contrast objective, compares the target node with its surrounding context directly in the embedding and structure space. Therefore, our model optimizes with these two self-supervised objectives that are closely related to node anomaly detection. During inference, we craft two scoring functions based on these two objectives, which tend to assign higher anomaly scores to attribute anomalies and structural anomalies in graphs. As shown on the right of Fig. 2, we compare the obtained anomaly score with the selected threshold. Nodes with scores higher than the threshold are considered abnormal, while nodes with scores lower than the threshold are considered normal.

### A. Transport Network Modelling

First, we model a traffic road network as a topology graph, where each road segment in the network represents a node and the connectivity between road segments is represented by edges in the graph. The attribute information of the node can be abstracted from the traffic information of the roadway, where the traffic information is collected by the sensors and sent to the IoT server for access. After modeling the traffic road network as a graph, since the structural information of the graph is the road information and the node attribute information is the traveling information of the vehicles on the road, the problem of detecting anomalies in vehicle trajectories is

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



transformed into the problem of detecting anomalous nodes on the graph, and then we use an augmented intelligence anomaly detection method on the graph to detect it. We give a notation representation of the graphs commonly used in the article as follows. A graph $G$ can be described as $G = (V, E, X, A)$, where $V = \{v_1, v_2, \cdots, v_N\}$ is the set of nodes in the graph, and $E$ is the set of edges. $X \in \mathbb{R}^{N\times D}$ and $A \in \mathbb{R}^{N\times N}$ denote the node features and graph adjacency matrix, where $D$ is the numbers of dimensions of node features.

### B. Subgraph Sampling Module

Similar to the vision domain, self-supervised learning methods for graphs can be broadly classified into two branches: generative method and comparative method. In the generative branch, most existing work focuses on attribute prediction, typically at the same scale [24], such as "node vs. node" (attribute regression) and "graph vs. graph" (structure prediction). On the other hand, comparative learning can distinguish instances not only at the same level but also at different scales, "node vs. graphs". As mentioned in [14], in graph anomaly detection, an anomaly generally manifests itself as a mismatch between it and its surrounding context, so we sampled subgraphs for each node to construct its contextual background.

In our approach, we first establish the process of selecting context subgraphs for the target nodes and propose two self-supervised learning objectives. Specifically, the anomaly detection in the attribute space is performed first, and the feature vector of the target node is reconstructed using the GCN encoder and decoder, and then it is compared with the original feature vector of the node. Furthermore, to detect anomalies existing in the structure space, we construct a hybrid hierarchical comparison between the target node and its local subgraphs. This comparison involves analyzing positive and negative samples, enabling us to identify structural anomalies. Based on this, we first sample the context information of the target node from the input graph and then select the one with a high score as the sampled subgraph of the target node by calculating their respective closeness scores.

For our attribute targets, the discriminant pairs are the original and reconstructed target nodes. On the other hand, in our comparison target, the two discriminant pairs are the target node and the two sampled graph views, respectively. We now elaborate on the aforementioned processing steps:

(1) **Target node sampling.** Since we are mainly concerned with detecting node-level anomalies in the graph, we first sample the target nodes. In this paper, we sample the target nodes from the given input graph by uniform sampling.

(2) **Graph view sampling.** It is then a matter of sampling the target node context subgraph, which is key to learning a high-quality representation of the center node. Due to the different importance of different neighbors and to control the randomness of sampling to some extent, we make improvements to the traditional random wandering sampling method [25]. Specifically, we first sample multiple subgraphs with the random walk algorithm and then calculate the importance scores of the neighbor nodes of the target nodes in these subgraphs. In this, for the computation of the importance of the neighbor nodes, we follow the subgraph sampling based on the personalized PageRank algorithm of [26], as described in [27], the importance score matrix $S$ can be denoted as

$$S = \alpha \cdot (I - (1-\alpha)\cdot \overline{A}), \tag{1}$$

where $I$ is the identity matrix, $\alpha \in [0, 1]$is a parameter usually set to 0.15. Term $\overline{A} = AD^{-1}$ denotes the adjacency matrix normalized by columns, $D$ is a diagonal matrix with $D(i,i) = \sum_j A(i,j)$ on its diagonal. $S(i,j)$ measures the importance scores between node $v_i$ and $v_j$. Based on this, we define the intimacy score for the measure of picking subgraphs:

**Definition** *1:* (Intimacy scores of subgraphs): For each subgraph $g_i^t$ generated by random walks of target node $v_i$, we calculate its intimacy scores $I_i^t = \sum_j S(i,j)$ separately, where $v_j \in g_i^t$.

And then the subgraphs $g_i^{\varphi_1}, \cdots, g_i^{\varphi_k}$ with scores in the top k are selected as the inputs for the training task afterwards.

(3) **Node anonymization.** In order to increase the difficulty of the two predefined self-supervised learning tasks and to prevent the original information from interfering with the learning results, the target nodes in the sampled graph views are anonymized [28](their original features are set to zero). Using this mechanism not only prevents information from leaking but also encourages the model to rely only on contextual information to identify anomalies.

### C. Generative Learning with Attribute Reconstruction

As noted earlier, the anomalies of an instance are usually expressed as the degree of disagreement between its original and reconstructed information. This disagreement can be quantified by measuring the $l^2$-norm distance. A greater distance indicates greater reconstruction error and a greater likelihood that the current instance is anomalous. Therefore, we would like to reconstruct the node's attribute information in an unsupervised manner and compare the reconstructed information with its original feature information for the purpose of detecting anomalies, and it is this model that is used in deep autoencoders( AEs). The traditional AEs have two parts, the encoder $Enc(\cdot)$ and the decoder $Dec(\cdot)$, the encoder first projects the input node features into a low-dimensional feature space, and then the decoder tries to restore them to the original data, the learning process of AE can be represented as follows:

$$\hat{x} = Dec(Enc(x)), \tag{2}$$

where $\hat{x}$ is the restored feature vector. The optimisation goal of the AEs is to make $\hat{x}$ and $x$ as close as possible. This can be achieved by minimising their $l^2$-norm distance, which encourages the AE to learn latent invariant patterns between inputs. We have enhanced the conventional AE architecture to account for the ability to incorporate the fundamental topological information in attribute reconstruction. Specifically, we build GCN-based encoders and decoders.

**GCN-based encoder.** $g_i^{\varphi_1}, \cdots, g_i^{\varphi_k}$ are the selected subgraphs of $v_i$. As mentioned earlier, we first have to map the features of the nodes in the subgraphs to a low-dimensional

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



vector space, and the process implemented by the one-layer GCN encoder can be denoted as:

$$H^{\varphi_j} = Enc_{GCN}(X_i^{\varphi_j}, A_i^{\varphi_j}) = \sigma(D^{-\frac{1}{2}}(A_i^{\varphi_j} + I)D^{-\frac{1}{2}}X_i^{\varphi_j}W_e), \tag{3}$$

where $H^{\varphi_j}$ denote the node embedding matrix of $g_i^{\varphi_j}$, $X_i^{\varphi_j}$ and $A_i^{\varphi_j}$ are the adjacency and feature matrices of the corresponding subgraphs, respectively. In the equation on the right, $\sigma(\cdot)$ represents an activation function, such as the the often-used ReLU, $D(i,i) = \sum_j A(i,j)$, and $W_e$ is the matrix of parameters used in the encoding training process.

**GCN-based decoder.** In a similar way, our graph decoder is constructed using a single layer of GCN. We take the embedding matrix obtained in the encoder as input to the decoder, the specific decoding process is as follows:

$$\tilde{X}_i^{\varphi_j} = Dec_{GCN}(H^{\varphi_j}, A_i^{\varphi_j}) = \sigma(D^{-\frac{1}{2}}(A_i^{\varphi_j} + I)D^{-\frac{1}{2}}H^{\varphi_j}W_d), \tag{4}$$

where $W_d$ is the matrix of parameters used in the decoding training process.

**Generative graph anomaly detection.** The reconstruction of attributes for anonymous node $v_i$ relies on obtaining information about the subgraphs of $g_i^{\varphi_1}, \cdots, g_i^{\varphi_k}$ through the GCN encoder and decoder, which by anonymising the target node, can make more full use of the attribute information from the surrounding context. Therefore, we set the objective function in the generative attribute learning process to minimise the Mean Square Error between the original features and the reconstructed features:

$$L_g^{\varphi_j} = \frac{1}{N}\sum_{i=1}^{N}(\tilde{X}_i^{\varphi_j}[-1,:], x_i)^2, j \in \{1, \cdots, k\}, \tag{5}$$

where $L_g^{\varphi_j}$ is the generative loss of the loss of all nodes on the sampled subgraph $s_i^{\varphi_j}$, the feature vectors for the reconstruction of node $v_i$ are obtained from the reconstructed feature matrix $\tilde{X}_i^{\varphi_j}$ generated by the decoder, and $x_i$ is the original feature vector of the node $v_i$. The final objective function is the average value of $L_g^{\varphi_j}$ on the subgraphs :

$$L_g = \frac{1}{k}\sum_{j=1}^{k}(L_g^{\varphi_j}). \tag{6}$$

### *D. Contrastive Learning with Structure Prediction*

As mentioned earlier, due to the strong correlation signals that precede nodes and their contextual subgraphs, node anomalies often manifest as disparities between a node and its environment. The aim of the generative module in the previous section was to identify anomalies in the attribute space; however, the structural information within the graph remained underutilised. To address this limitation, we suggest implementing a multiple-view comparison learning module. In the previous section, we discussed the aim of node-level generative attribute learning, whereas the comparison learning within the current module blends various graph topological scales to highlight semi-global information. The comparison module is made up of three key components: the graph encoder, the pooling module and the contrastive module.

**GCN-based encoder.** In this module, our goal is to establish the comparison between the target node and its contextual subgraph, therefore, the inputs to the GCN encoder are the feature matrix of the subgraph and the feature vectors of the nodes, the process of encoding the feature matrix of the subgraphs has been shown in Equation (3), the encoding process of the individual nodes is slightly different from it:

$$h_i = \sigma(x_i W_e), \tag{7}$$

where $x_i$ is the feature vectors of target node $v_i$, $h_i$ is the corresponding embedding vectors. Unlike the graph eigenvector representation, the graph adjacency matrix is not used as an input term in the generation of the node's representation. This is because individual nodes do not contain graph structures themselves. Furthermore, the parameter matrix $W_e$ is shared with Equation (3).

**Pooling module.** Since our goal is to compare the target node with its contextual background information, for this comparison to work, we need a pooling function $\Phi(\cdot)$ to aggregate the obtained subgraph features to obtain a graph representation vector, as shown in the middle of Figure2. Common pooling functions are average pooling and bilinear pooling, etc. To simplify the operation, we use in this paper the average pooling function, formulated as follows:

$$s_j = \Phi(H^{\varphi_j}) = \frac{1}{n}\sum_{k=1}^{n} H^{\varphi_j}[k,:], \tag{8}$$

where $n$ is the number of nodes in sampled subgraphs, and $s_j$ is the graph representation of subgraph $g_i^{\varphi j}$.

**Contrastive module.** Since self-supervised contrastive learning does not rely on the original label information but is trained by an encoder comparing positive and negative examples, the selection of positive and negative samples is crucial in contrastive learning. We begin by representing the positive and negative examples as follows:

$$e_i^j = (h_i, s_j), \tag{9}$$

$$\tilde{e}_i^j = (h_i, \tilde{s}_j), \tag{10}$$

where $e_i^j$ denotes the positive example of the target node $v_i$, and $e_i^j$ is the corresponding negative example, in which $\tilde{s}_j$ is generated by corrupting $s_j$, specifically, it is generated by disrupting the embedding matrix $H^{\varphi_j}$, and then aggregated by Equation (8).

As described in [14], we adopt a discrimination to contrast the positive and negative examples and score the samples:

$$d_i^j = \sigma(h_i W_d s_j), \tag{11}$$

$$\tilde{d}_i^j = \sigma(h_i W_d \tilde{s}_j), \tag{12}$$

where $d_i^j$ and $\tilde{d}_i^j$ are the discrimination scores for positive and negative sample pairs $e_i^j$ and $\tilde{e}_i^j$, respectively. And $\sigma(\cdot)$ is a non-linear function, here we adopt the sigmoid function to keep the resulting discrimination score in the interval [0, 1].

**Contrastive graph anomaly detection.** Since nodes depend on their regional contexts and different nodes have different contextual subgraphs. If a node is anomalous, it should

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



be more distinct from its surrounding contexts, numerically should be that $d_i^j$ is significantly higher than $\tilde{d}_i^j$. Here, we adopt Jensen-Shannon divergence [12] to measure the value of the difference between these two scores:

$$L_c^{\varphi_j} = -\frac{1}{2N}\sum_{i=1}^{N}(\log(d_i^{\varphi_j}) + \log(1-\tilde{d}_i^{\varphi_j})), j \in \{1, \cdots, k\}, \tag{13}$$

where $L_c^{\varphi_j}$ is the contrastive loss value on subgraph $s_i^{\varphi_j}$. Similar to the previous module, the final contrastive objective function is the average of all subgraph losses at node $v_i$:

$$L_c = \frac{1}{k}\sum_{j=1}^{k}(L_c^{\varphi_j}). \tag{14}$$

### E. Node Anomaly Scoring

The above describes the generative loss and contrastive loss during training, and the next describes the computation of specific anomaly scores during final inference. Since most of the nodes in a graph are normal nodes, the graph encoder and decoder we choose should be able to do the job of mapping the feature vectors of the normal nodes to the appropriate embedding space after training, and vice versa. Whereas an anomalous node, either structural or attribute anomalies, will distort its features when being mapped to the latent space.

Since in the process of generative learning, the attribute reconstruction for a target node only relies on the attribute information in the subgraphs of its contextual neighbors, as mentioned above, the degree of difference between the original features and the reconstructed features can be used as an indicator to determine whether the node is abnormal or not. In the specific application, we use the $l^2$-norm distance. as the scoring function to calculate the degree of difference, and the process is formulated as follows:

$$score_g(v_i) = \frac{1}{k}\sum_{j=1}^{k}(\theta(\|\tilde{X}_i^{\varphi_j}[-1,:] - x_i)\|_2^2), \tag{15}$$

where $score_g(\cdot)$ is the scoring function of the generative anomaly detection module.$\theta$ is the parameter that adjusts the final score to fall into [0, 1], $score_g(v_i)$.

As for the constructive module, since we have already computed the scores of the respective sample pairs when generating the positive and negative sample pairs, according to the Equation (11) and (12), we give the final scoring function for constructive learning:

$$score_c(v_i) = \frac{1}{k}\sum_{j=1}^{k}(d_i^j - \tilde{d}_i^j), \tag{16}$$

where $score_g(v_i)$ is the contrastive score of target node $v_i$. In the sample pairs we have taken to, as described by Equations (9) and (10), positive pairs are composed of the target node with its neighboring subgraphs, and negative pairs are composed of the target node with the disrupted subgraphs, so if a node is abnormal, it may exhibit a mismatch with its neighboring subgraphs, and this mismatch is similar to the one shown with its negative sample pairs, thus $d_i^j$ and $\tilde{d}_i^j$ should be numerically close to each other, and the $d_i^j - \tilde{d}_i^j$ approaches 0. Conversely, if a node is normal, the difference between $d_i^j$ and $\tilde{d}_i^j$ is larger, manifesting as $d_i^j$ converging to 1 and $\tilde{d}_i^j$ converging to 0. To summarize, the range of $d_i^j - \tilde{d}_i^j$ is in [0, 1].

The final anomaly score is obtained by combining equations (15) and (16) :

$$score(v_i) = \lambda score_g(v_i) + \gamma score_c(v_i), \tag{17}$$

where $\lambda$ and $\gamma$ are two adjustable parameters used to adjust the weights accounted for by the comparison and generation scoring functions.

### F. Model Optimization and Algorithm

Our optimization objective is composed of the Eq.(6) and (14):

$$L = \lambda L_g + \gamma L_c, \tag{18}$$

where $L$ is the total training loss we want to minimize. The steps of our proposed methodological process are summarized in Algorithm 1. We first sample a batch of nodes. For each node, we generate k subgraphs with the highest intimacy score to the current target node based on the importance score matrix. Then, we choose GCN as the encoder to encode the nodes in the batch and the corresponding sampled subgraphs. Our goal is to reconstruct the feature vectors of the anonymized target nodes in each subgraph, and to compute the generative loss incurred in this process, we decode the node embeddings of the subgraphs by the GCN decoder and compare the reconstructed feature vectors of the nodes with their original feature vectors. The comparison loss is then computed by first aggregating the node embeddings in the subgraph to generate the subgraph representation, which is then generated by comparing it with the target node embeddings. The final training loss is then computed by combining two different targets. The anomaly scores for each node in $G$ are repeated $R$ times during inference, and the subgraphs are chosen randomly from a several subgraphs with the highest intimacy scores, ensuring randomness and ensuring that the final anomaly scores are statistically stable.

## IV. SIMULATION STUDIES

In this section, we conduct experiments on six real-world datasets to validate the effectiveness of our scheme for node classification in an unsupervised manner. We compare our method with the state-of-the-art anomaly detection and self-supervised learning methods. We conduct an ablation study and parameter sensitivity experiments to further investigate the properties of the model.

### A. Dataset Description

We conduct pre-training on four datasets, namely BlogCatalog, Flickr, Cora, and CiteSeer, and compare our model with existing supervised and unsupervised learning methods. Subsequently, we conduct experiments on real traffic datasets Los-loop and SZ-taxi to validate the effectiveness of our

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



**Algorithm 1** Anomaly Score Calculation based on Self-Supervised Algorithm.

**Input:** A graph $G$ with input feature $V$; Batch size $B$; Number of training epochs $N$; Number of evaluation rounds $R$.
**Output:** Anomaly score for each node in the graph $G$.

Compute importance score matrix S according to Eq. (1);
// Training stage
**for** $n \in (1, 2, \cdots, N)$ **do**
  $B \leftarrow$ Split $V$ into batches;
  **for** batch $b = (v_1, \cdots, v_b) \in B$ **do**
    Randomly sample t subgraphs;
    Calculate the intimacy scores for each subgraph and select k subgraphs with the highest scores among them;
    Encode target node via Eq. (3) and Eq. (7) and associated subgraph via Eq. (8) to obtain latent representation;
    Contrast positive sample pairs and negative sample pairs, calculate respective scores via Eq. (11) and Eq. (12);
    Decode the node representation via Eq. (4) to obtain the reconstruction node feature;
    Calculate the loss $L$ via Eq. (6) and Eq. (14);
  **end for**
**end for**
// Inference stage
**for** $v_i \in V$ **do**
  **for** evaluation rounds $r \in 1, 2, \ldots, R$ **do**
    Calculate $score(v_i)$ via Eq. (17);
  **end for**
  Calculate the average score of the $R$ rounds of node $v_i$;
**end for**

model. In many deep learning tasks, the pre-training needs larger datasets, which can provide better model representation capabilities. Moreover, the traffic datasets with features are small-scale and hard to obtain, so we opted for a pre-training approach on BlogCatalog, Flickr, Cora, and CiteSeer datasets. By the pre-training, we can train the model with richer and more comprehensive data, allowing it to learn richer features and patterns. Thus, the pre-trained model presents a better initial state on the new dataset. Therefore, we can deduce that our method has better performance and results on the traffic dataset. The corresponding descriptions of these datasets are as follows:

- **BlogCatalog**: The BlogCatalog [29] dataset contains the data of thousands of blog users, where each user is represented by a node. Each user's information includes his/her profile, interest tags as node features, and the relationships with other users as edges. These blog users and the relationships between them form a social network.
- **Flickr**: Flickr is an image-sharing website and similar to the BlogCatalog dataset. It is also a social network where users' interest tags for sharing images are used as nodes' features attributes in this dataset.
- **Cora**: Cora [30] is a commonly used academic literature citation dataset for machine learning. In this dataset, the nodes represent academic papers, the features of the nodes include paper titles, abstracts, etc., and the connecting edges between the nodes indicate the previous citation relationships of the papers.
- **CiteSeer**: Similar to the Cora dataset, CiteSeer is a citation dataset. It contains academic papers and their citation relationships in the field of computer science and is used to study the association and citation patterns among academic literature.
- **Los-loop**: We choose the real dataset collected by loop detectors on freeways in Los Angeles in [31]. This dataset is selected from 207 sensors and their traffic speeds from March 1 to March 7, 2012, and the traffic speeds are tested every five minutes. We select only one day's data from this dataset as the feature matrix input in our experiments, and each row of the adjacency matrix in the data represents one road, and the values in the matrix represent the distance between sensors on the road.
- **SZ-taxi**: The dataset is about cab trajectories on January 1, 2015, in Shenzhen [31]. The study area comprises 156 major roads in Luohu District. The dataset has two parts, the first being a 156*156 adjacency matrix that provides information on the spatial relationship between roads. Each row represents a road, and the matrix values denote connectivity between roads. The second part is a feature matrix that describes the changes in speed over time for each road. Each row represents a road, and each column denotes the speed of traffic on the road at different times. The traffic speed on every road is summarized every 15 minutes.

The above datasets do not contain anomalies and require us to manually inject anomalies into them for validation evaluation. Here, we adopt the anomaly generation strategies in [21] and [14], which generate anomalies by perturbing the node features in the graph and the structural information of the graph, respectively. In this case, attribute anomalies are generated by randomly selecting distant node attributes in place of the original attributes of the selected nodes, followed by randomly selecting a portion of the nodes so that they are completely connected to generate structural anomalies. We summarize the above six datasets into Table II.

### *B. Experimental Settings*

We describe the experimental settings in this subsection, including the baseline methodology, evaluation metrics and parameter settings.

**Baseline methodology.** We compare our model with other anomaly detection methods on four datasets, and the following is a description of the comparison methods:

- **DOMINANT** [21] uses the depth autoencoder of the graph to reconstruct the adjacency and feature matrices of the graph, which are then compared with the matrices of the original graph, and the anomalies in the graph can be measured in terms of reconstruction errors of the nodes.

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



TABLE II
THE STATISTICS OF THE DATASETS

| Datasets | BlogCatalog | Flickr | Cora | CiteSeer | Los-loop | SZ-taxi |
|---|---|---|---|---|---|---|
| Nodes | 5,196 | 7,575 | 2,708 | 3,327 | 207 | 156 |
| Edges | 171,743 | 239,738 | 5,429 | 4,732 | 2,853 | 532 |
| Features | 8,189 | 12,047 | 1,433 | 3,703 | 288 | 96 |
| Anomalies | 300 | 450 | 150 | 150 | 15 | 13 |

- **Radar** [32] is an unsupervised anomaly detection framework that detects anomalies by combining the residuals of the attribute information and the intrinsic structure of the network information.
- **AMEN** [20] is also an unsupervised learning scheme. It detects anomalies by combining the attribute information of nodes and graph structure information in the ego network.
- **DSAD** [33] is a deep semi-supervised anomaly detection method. This method uses abnormal and normal labeled instances and unlabeled instances to learn an anomaly detector for independently and identically distributed data to perform anomaly detection on independently and identically distributed data.
- **SL-GAD** [15] constructs different contextual subgraphs for each node and employs two modules, namely, generative attribute regression and multi-view comparison learning, to detect attribute and structural anomalies, respectively.

**Evaluation metrics.** We choose ROC-AUC as a comparative indicator to evaluate the performance of our model against other models [34]. In this case, the ROC curve is a curve with the true positive rate (anomalies identified as abnormal) as the vertical axis and the false positive rate (normal nodes identified as abnormal) as the horizontal axis, which is used to visualize the performance of the classification model. The value of AUC is the area under the ROC curve, and the closer its value is to 1, the better the method detects the samples, and it can more accurately distinguish positive and negative samples. On the contrary, the closer its value is to 0, the less effective it is. In addition, there are some other common machine learning metrics that we have chosen in order to show the detection effeciency of our method more comprehensively. We first select the point on the ROC curve that is closest to the (0, 1) coordinate. The point ensures that the true-positive rate is as large as possible and the false-positive rate is as small as possible, and the threshold at this point is considered to be the optimal threshold. Based on the true-positive (TPR) and false-positive rates (FPR) at this point, we can obtain the other metrics by calculation. Note that, the true positives (TP) are the number of positive case samples that the model correctly predicts as positive cases, false positives (FP) are the number of negative case samples that the model incorrectly predicts as positive cases, true negatives (TN) are the number of negative case samples that the model correctly predicts as negative cases, false negatives (FN) are the number of positive case samples that the model incorrectly predicts as negative cases.

- **Accuracy** is the ratio of the number of samples correctly predicted by the model to the total number of samples and is one of the most commonly used assessment metrics. The specific formula is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}. \tag{19}$$

- **Precision** is the proportion of all positive samples that are predicted to be positive cases by the model. The specific formula is:

$$Precision = \frac{TP}{TP + FP}. \tag{20}$$

- **F1-Score** is the reconciled mean of the precision and false positive (recall) rates, combining the accuracy and comprehensiveness of the model. The specific formula is:

$$F1\text{-}Score = 2 \times \frac{TruePositiveRate \times Precision}{TruePositiveRate + Precision}. \tag{21}$$

**Parameter settings.** Regarding the parameter settings in the experiments, if the subgraph is too small, we will get too little node context information, and if the subgraph is too large, it will introduce redundant information and increase the demand for computational resources. On balance, we choose the subgraph size to be 6, and the hidden dimension in the GCN encoder is unified to be 64. Then, the learning rate and the number of training epochs as well as the loss parameter settings are set differently in different datasets. The learning rate is chosen to be 0.001 for all the datasets except for the BlogCatalog dataset where it is 0.003. 100 epochs of training are performed on the Cora and Citeseer datasets for the performance evaluation, while a total of 400 epochs are performed on the BlogCatalog and Flickr datasets. The number of epochs evaluated is then uniformly 256. For the parameter settings, $lambda$ is fixed to be 1, while the value of $\gamma$ is chosen to be 0.4 for training on the CiteSeer dataset, and 0.6 for all the other datasets. For the other baselines, we retain the settings described in the corresponding papers.

### C. Experimental Results

We compared our model with the baseline methods described above to assess model performance. Among the experimental results, the AUC value is in Table III, while the ROC curve is shown in Fig. 3. Besides, we put our results for all indicators on the six datasets in Table IV. Based on the observation of the experimental results, we make the following conclusions:

- In the ROC curves in Fig. 3, we can see that the results of the pre-training of our model outperform the other models on the four datasets of BlogCatalog, Flickr, Cora, and

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023



TABLE III
ABNORMALITY DETECTION PERFORMANCE (I.E., AUC VALUE) WITH THE BEST RESULTS ON EACH DATASET IN BOLD

| Method | BlogCatalog | Flickr | Cora | CiteSeer | Los-loop | SZ-taxi |
|---|---|---|---|---|---|---|
| AMEN | 0.6385 | 0.6572 | 0.6260 | 0.6150 | N/A | N/A |
| Radar | 0.7384 | 0.7301 | 0.6439 | 0.6709 | N/A | N/A |
| DOMINANT | 0.7466 | 0.7442 | 0.8152 | 0.8247 | N/A | N/A |
| DSAD | 0.6412 | 0.7201 | 0.7240 | 0.8143 | N/A | N/A |
| SL-GAD | 0.8095 | 0.7850 | 0.8801 | 0.8773 | N/A | N/A |
| Ours | **0.8165** | **0.7865** | **0.9047** | **0.8963** | **0.8666** | **0.8849** |

TABLE IV
ANOMALY DETECTION PERFORMANCE OF OUR METHOD ON MULTIPLE METRICS AND ACROSS MULTIPLE DATASETS.

| Metrics | BlogCatalog | Flickr | Cora | CiteSeer | Los-loop | SZ-taxi |
|---|---|---|---|---|---|---|
| AUC | 0.8165 | 0.7865 | 0.9047 | 0.8963 | 0.8666 | 0.8849 |
| FPR | 0.2346 | 0.2637 | 0.1552 | 0.1728 | 0.1041 | 0.1048 |
| TPR | 0.6966 | 0.6933 | 0.8533 | 0.8066 | 0.8667 | 0.8468 |
| Accuracy | 0.6997 | 0.6865 | 0.8524 | 0.8043 | 0.8647 | 0.8502 |
| Precision | 0.9798 | 0.9763 | 0.9895 | 0.9898 | 0.9881 | 0.9889 |
| F1-Score | 0.8154 | 0.8124 | 0.9169 | 0.8893 | 0.9209 | 0.9119 |

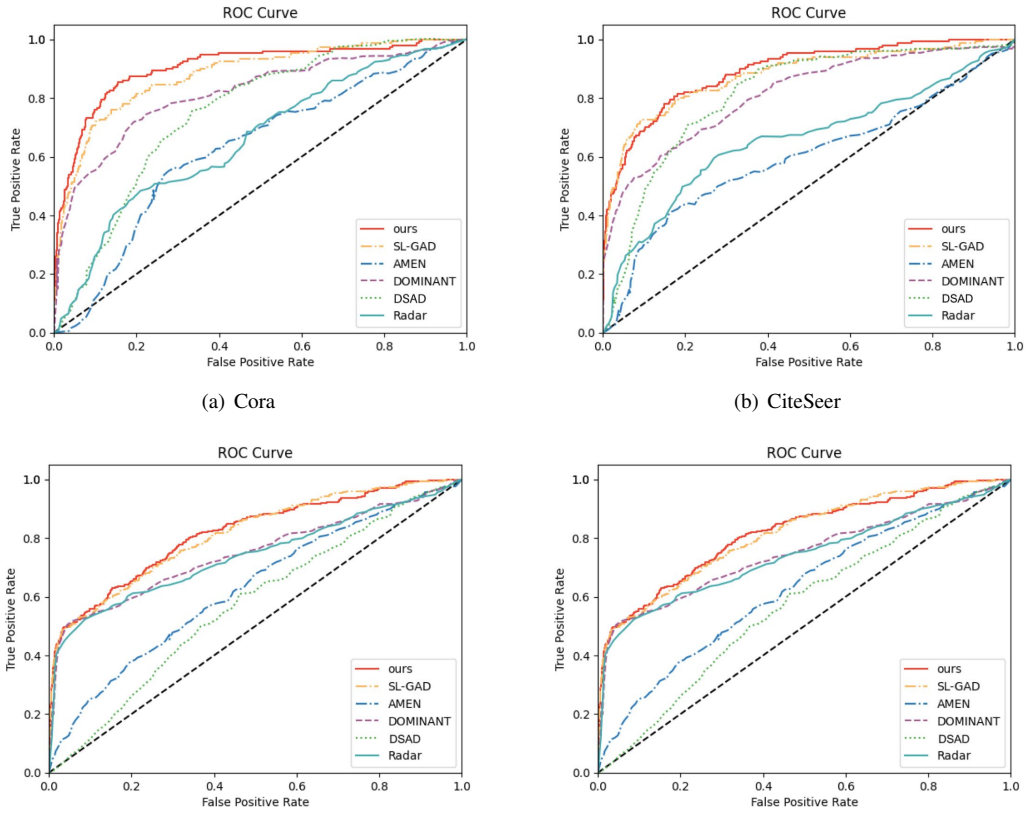
CiteSeer, and there is a more significant improvement on the Cora dataset and CiteSeer dataset compared to the previous scheme. Especially on the Cora dataset, we notice that the rising segment of the ROC curve is steeper, which implies that our model has a relatively high true positive rate at different thresholds, accompanied by a lower false positive rate. It suggests that our model has higher accuracy and robustness in the anomaly detection task on the Cora dataset. This validates the effectiveness of multi-task self-supervised learning combined with subgraph selection for anomaly detection.

- Table III is the specific AUC values corresponding to Fig. 3 for our model and other comparison methods on the four datasets of BlogCatalog, Flickr, Cora, and CiteSeer. Among them, AMEN and Radar methods perform poorly because of the limitations of their shallow mechanisms, which are less capable of handling complex graph structures and nonlinear data. This may be due to their inability to fully capture the complex patterns and contextual information in the graphs. On the BlogCatalog and Flickr datasets, the AMEN method achieves AUC values of 0.6385 and 0.6572, respectively, while the Radar method achieves AUC values of 0.7384 and 0.7301. In contrast, our model achieves higher AUC values of 0.8165 and 0.7865 on these two datasets, proving the superiority of our method in handling complex graph structures and nonlinear data. DSAD, due to the combination of labeled and unlabeled samples, utilizes information from unlabeled samples to improve model performance. This may have an impact on the performance of DSAD as the model is more likely to favor normal samples and may not be accurate enough to identify abnormal samples. And although DOMINANT is a deep method, its unsupervised learning model only considers the reconstruction of attributes and structures, and the subgraphs in SL-GAD are only obtained by random wandering, and the correlation between subgraphs and nodes may be slightly insufficient. In the table, the SL-GAD method achieves higher AUC values of 0.8801 and 0.8773 on the Cora and CiteSeer datasets, respectively. In contrast, our method achieves higher AUC values of 0.9047 and 0.8963 on these two datasets, further validating the effectiveness of our subgraph selection strategy. In summary, our method selects subgraphs by their proximity to the target nodes, and combines generative and contrastive self-supervised learning strategies to achieve better results in anomaly detection. Compared to other methods, our model achieves higher AUC values on multiple datasets, proving its superiority in dealing with complex graph structures and nonlinear data.
- In Table IV, our experimental results on both Cora and CiteSeer datasets have FPR below 0.2 and TPR above 0.8, which prove that our model has a low false alarm rate in detecting anomalous nodes and is efficient in identifying anomalous nodes. Precision of our model is higher than 0.97 on all four datasets, indicating the effectiveness of our algorithm for the identification of anomalous nodes. F1-Score is the reconciled average of Precision and Recall, which combines the accuracy and completeness of the model. The value of the F1-Score of our model is above 0.8 on each dataset, which means that our model strikes a good balance in the anomaly detection task with high precision and recall, i.e., it can identify anomalous nodes accurately and less likely to misclassify normal nodes as anomalous nodes. This suggests that our model can effectively balance the risk of false positives and omissions, and achieve a better performance in dealing with the anomaly detection problem. In addition to evaluating the performance of pre-trained models on the BlogCatalog, Flickr, Cora, and CiteSeer datasets, we further conduct experiments to validate the effectiveness of our proposed self-supervised learning algorithm on graph-based anomalous node detection tasks. Specifically, we extend our evaluation to include real-world traffic datasets, namely Lop-loop and SZ-taxi. The Lop-loop and SZ-taxi datasets capture the speeds of vehicles on various roadways, particularly neighboring roadways, which often exhibit similarities at the same time. Our objective is

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023






(a) Cora (b) CiteSeer

(c) Flickr (d) BlogCatalog

Fig. 3. ROC curves on the four datasets. The larger the AUC of the area under the curve, the better the performance of graph anomaly detection.

to identify and connect nodes on the road network that display significant differences. By applying our proposed algorithm, we aim to detect anomalous nodes effectively in this traffic context. The experimental results not only demonstrate the efficiency of our algorithm on traditional graph machine learning datasets, but also highlight its effectiveness on real-world traffic datasets. These findings reinforce the versatility and robustness of our model in diverse graph-based applications.

## V. Conclusion

The AIoT technology combines the advancements of IoT and augmented intelligence techniques, and enables synergistic utilization. In this paper, we apply AIoT technology to vehicle-road collaboration and propose a self-supervised learning algorithm for detecting anomalies in vehicle trajectories. We first modelled the vehicle travel data collected by the IoT sensors and the traffic road network as a topology graph. Then, we processed the topology graph in the IoT server using deep self-supervised learning algorithms for anomaly detection. Since the self-supervised learning algorithm is trained using unlabeled data, it does not need to manually label a large amount of data, which reduces the energy overhead and improves the efficiency of energy utilization. Specifically, we started by picking out some subgraphs of the target node using random walks, and the use of subgraphs avoids the computational and communication overhead of the need to transmit the entire graph information, reduces the burden on the network traffic, and reduces the use of computational resources. We calculated the intimacy scores between these subgraphs and the target nodes, and chose the ones with higher scores. Subsequently, two self-supervised learning modules were employed to detect attribute anomalies and structural anomalies of nodes, respectively, generating anomaly scores for each node. Higher scores indicate a higher likelihood of anomalies. Finally, to validate the effectiveness of the algorithm, we conducted pre-training on four commonly used datasets and evaluated the obtained model on two traffic datasets, and the accuracy of the experiments were 86.47% and 85.2%, respectively, indicating that our algorithm performs well.

## Acknowledgment

The authors declare that there is no conflict of interest regarding the publication of this paper. This work was supported in part by the National Natural Science Foundation of China under Grants 62171132, 62102088, and U1905211, in part by Fok Ying Tung Education Foundation under Grant 171061, in part by the Natural Science Foundation of Fujian Province under Grant 2021J05228.

## References

[1] B. Ducourthial, V. Cherfaoui, T. Fuhrmann, and S. Bonnet, "Experimentation of a road hazard anticipation system based on vehicle cooperation," *Vehicular Communications*, vol. 36, p. 100486, 2022.

Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2024.3398023




[2] X. Han, T. Grubenmann, R. Cheng, S. C. Wong, X. Li, and W. Sun, "Traffic incident detection: A trajectory-based approach," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 1866–1869, 2020.

[3] B. Jiang, J. Du, C. Jiang, Z. Han, and M. Debbah, "Underwater searching and multi-round data collection via auv swarms: An energy-efficient aoi-aware mappo approach," *IEEE Internet of Things Journal, DOI: 10.1109/JIOT.2023.3336055*, 2023.

[4] L. Zong, D. Qiao, H. Wang, and Y. Bai, "Sustainable cross-regional transmission control for the industrial augmented intelligence of things," *IEEE Transactions on Industrial Informatics, DOI: 10.1109/TII.2022.3230674*, 2023.

[5] J. Du, C. Jiang, J. Wang, Y. Ren, and M. Debbah, "Machine learning for 6g wireless networks: Carrying forward enhanced bandwidth, massive access, and ultrareliable/low-latency service," *IEEE Vehicular Technology Magazine*, vol. 15, no. 4, pp. 122–134, 2020.

[6] J. Ye, J. Zhao, K. Ye, and C. Xu, "How to build a graph-based deep learning architecture in traffic domain: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3904–3924, 2020.

[7] C. J. Dong, C. F. Shao, C. X. Zhuge, and M. Meng, "Spatial and temporal characteristics for congested traffic on urban expressway," *Beijing Gongye Daxue Xuebao/Journal of Beijing University of Technology*, vol. 38, no. 8, pp. 1242–1246, 2012.

[8] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligences (ICJAI)*, pp. 3634–3640, 2018.

[9] K. Sun, Z. Lin, and Z. Zhu, "Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5892–5899, 2020.

[10] Rui and Yong, "From artificial intelligence to augmented intelligence," *IEEE Multimedia*, vol. 24, no. 1, pp. 4–5, 2017.

[11] R. Mishra, H. P. Gupta, R. Kumar, and T. Dutta, "Leveraging augmented intelligence of things to enhance lifetime of uav-enabled aerial networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 1, pp. 586–593, 2022.

[12] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.

[13] K. Hassani and A. H. Khasahmadi, "Contrastive multi-view representation learning on graphs," in *International Conference on Machine Learning (PMLR)*, pp. 4116–4126, 2020.

[14] Y. Liu, Z. Li, S. Pan, C. Gong, C. Zhou, and G. Karypis, "Anomaly detection on attributed networks via contrastive self-supervised learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 6, pp. 2378–2392, 2021.

[15] Y. Zheng, M. Jin, Y. Liu, L. Chi, K. T. Phan, and Y.-P. P. Chen, "Generative and contrastive self-supervised learning for graph anomaly detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 220–12 233, 2021.

[16] S. Ahmadi, "Chapter 7 - vehicle-to-everything (v2x) communications," in *5G NR*, S. Ahmadi, Ed. Academic Press, pp. 789-843, 2019.

[17] C. Oh, E. Jung, H. Rim, K. Kang, and Y. Kang, "Intervehicle safety warning information system for unsafe driving events: Methodology and prototypical implementation," *Transportation Research Record*, vol. 2324, no. 1, pp. 1–10, 2012.

[18] G. Pang, C. Shen, L. Cao, and V. D. H. Anton, "Deep learning for anomaly detection: A review," *ACM Computing Surveys*, no. 2, p. 54, 2022.

[19] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "Anomalous: A joint modeling approach for anomaly detection on attributed networks," in *Twenty-Seventh International Joint Conference on Artificial Intelligence IJCAI-18*, pp. 3513–3519, 2018.

[20] B. Perozzi and L. Akoglu, "Scalable anomaly ranking of attributed neighborhoods," in *Proceedings of the 2016 SIAM International Conference on Data Mining (SIAM)*, pp. 207–215, 2016.

[21] K. Ding, J. Li, R. Bhanushali, and H. Liu, "Deep anomaly detection on attributed networks," in *Proceedings of the 2019 SIAM International Conference on Data Mining (SIAM)*, pp. 594–602, 2019.

[22] Y. Li, X. Huang, J. Li, M. Du, and N. Zou, "Specae: Spectral autoencoder for anomaly detection in attributed networks," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (ICIKM)*, pp. 2233–2236, 2019.

[23] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 11, pp. 4037–4058, 2020.

[24] Y. Liu, M. Jin, S. Pan, C. Zhou, Y. Zheng, F. Xia, and S. Y. Philip, "Graph self-supervised learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 6, pp. 5879–5900, 2022.

[25] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Sixth International Conference on Data Mining (ICDM)*, pp. 613–622, 2006.

[26] G. Jeh and J. Widom, "Scaling personalized web search," in *Proceedings of the 12th International Conference on World Wide Web*, pp. 271–279, 2003.

[27] J. Zhang, H. Zhang, C. Xia, and L. Sun, "Graph-bert: Only attention is needed for learning graph representations," *arXiv preprint arXiv:2001.05140*, 2020.

[28] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 5812–5823, 2020.

[29] L. Tang and H. Liu, "Relational learning via latent social dimensions," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 817–826, 2020.

[30] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "Arnetminer: extraction and mining of academic social networks," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pp. 990–998, 2008.

[31] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-gcn: A temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation systems*, vol. 21, no. 9, pp. 3848–3858, 2019.

[32] J. Li, H. Dani, X. Hu, and H. Liu, "Radar: Residual analysis for anomaly detection in attributed networks." in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence(IJCAI)*, vol. 17, pp. 2152–2158, 2017.

[33] L. Ruff, R. A. Vandermeulen, N. Grnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, "Deep semi-supervised anomaly detection," in *International Conference on Learning Representations(ICLR)*, pp. 1-23, 2019.

[34] Z. Peng, M. Luo, J. Li, H. Liu, and Q. Zheng, "Anomalous: A joint modeling approach for anomaly detection on attributed networks." in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3513–3519, 2018.


Authorized licensed use limited to: Fujian Normal University. Downloaded on June 11,2024 at 06:07:18 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.