International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-14 Issue-9, August 2025

Building an Application to Monitor Abnormal Network Node Traffic using Graph Learning Method

Phan Thi Ha, Trinh Thi Van Anh

Abstract: ***The authors have developed an advanced monitoring application based on state-of-the-art graph learning methods to enhance the management and operational safety of telecommunications and network systems in Vietnam. This system integrates Long Short-Term Memory (LSTM) [6], Graph Convolutional Networks (GCNs) [5], and Graph Attention Networks (GATs) [4] to model and analyse the dynamic behaviour of network nodes. By combining temporal and structural features of network data, the application is capable of detecting anomalies, forecasting potential failures, and recommending timely interventions with minimal human oversight. These predictive capabilities significantly enhance the reliability, efficiency, and safety of telecommunications infrastructure, while also reducing downtime and lowering operational and maintenance costs. Additionally, the authors have developed custom software tools to support the labelling of raw datasets collected from network nodes. These datasets originate from existing static alerting systems used by network operators. The labelling tool enables efficient and consistent annotation of data, which is crucial for training and validating machine learning models. By transforming unstructured raw logs into structured labelled data, the system ensures higher accuracy in learning algorithms. Overall, this solution offers a practical and intelligent approach to managing large-scale network systems, particularly in developing regions like Vietnam, where automation and cost efficiency are critical.***

Keywords: ***Long Short-Term Memory (LSTM), Graph Convolutional Networks (GCNs), Graph Attention Networks (GATs).***

Abbreviations:
SMOTE: Synthetic Minority Over-sampling Technique
LSTM: Long Short-Term Memory
GCNs: Graph Convolutional Networks
GATs: Graph Attention Networks
GRU: Gated Recurrent Unit

I. INTRODUCTION

Nowadays, network systems are increasingly complex and large-scale, requiring practical monitoring tools to ensure stable and continuous operation. Network monitoring not only helps detect and resolve problems promptly but also provides valuable information about the system's performance, status, and operating patterns. In telecommunications, monitoring network nodes [1] is immensely necessary. Network nodes are key components in transmitting data and providing telecommunication services. Issues with network nodes can cause severe disruptions, affecting the entire system's operation. Telecommunication networks are the foundation for not only communication connections but also many essential services in modern life, including e-commerce, banking and finance, as well as healthcare and education. Therefore, monitoring the traffic and maintaining the stability and safety of network nodes are extremely important. Traditional static alerting systems are designed to detect network failures through predetermined static rules (software such as Nagios, Zabbix, or SolarWinds helps automate the monitoring process) [8]. Traditional methods in network monitoring often follow static rules (rule-based) or use simple statistical models to detect anomalies. These methods work by setting fixed thresholds for parameters such as bandwidth, latency, and packet count. When a value exceeds a predetermined threshold, the system issues an alert about the potential for an anomaly.

The limitation of these systems is their inflexibility and difficulty in adapting to changes in the network, which can lead to false positives or missed new incidents (false negatives). To address this limitation, this paper proposes a method that combines the results from the static alerting system, graph learning, **and time series classification methods**. Researching and applying graph learning methods to detect abnormal traffic not only enhances detection ability but also opens up a more effective and intelligent approach.

II. GRAPH LEARNING METHOD

The method proposed in this paper combines a time series classification model and graph learning to detect abnormal traffic in network systems. In which:

Graph Learning analyses the structure of links between nodes to detect systematic anomalies. Nodes in telecommunications networks often represent devices or entities in the system (routers, servers, etc.). The links between nodes are determined by a number calculated based on the error classification from the static alerting system, the level of impact of the error, or the level of anomaly that the alerting system has previously labelled. Here, the authors use the Graph Convolutional Networks (GCNs) [4] and Graph Attention Networks (GATs) [5] learning models

Manuscript received on 09 July 2025 | First Revised Manuscript received on 24 July 2025 | Second Revised Manuscript received on 05 August 2025 | Manuscript Accepted on 15 August 2025 | Manuscript published on 30 August 2025.
*Correspondence Author(s)
Dr. Phan Thi Ha*, Lecturer, Faculty of Information Technology at Posts and Telecommunications Institute of Technology (PTIT) in Ha Noi, Vietnam, and Computing Fundamental Department, FPT University, Hanoi, Viet Nam. Email ID: hapt@ptit.edu.vn, hapt27@fe.edu.vn and hathiphan@yahhoo.com, ORCID ID: 0009-0003-2521-0717
Th S. Trinh Thi Van Anh, Lecturer, Faculty of Information Technology at Posts and Telecommunications Institute of Technology (PTIT) in Ha Noi, Vietnam, and Computing Fundamental Department, FPT University, Hanoi, Viet Nam. Email ID: vanh22@yahoo.com, anhttv@ptit.edu.vn and anhttv20@fe.edu.vn, ORCID ID: 0009-0005-8062-2014

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an open access article under the CC-BY-NC-ND license http://creativecommons.org/licenses/by-nc-nd/4.0/

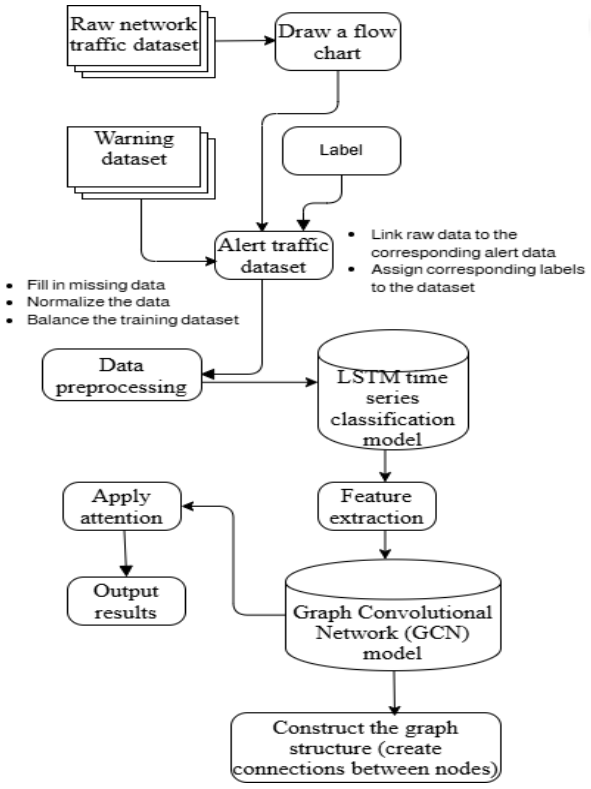
to explore complex relationships between nodes, thereby detecting anomalous patterns that the static alerting system cannot recognize.

Time series classification [2] is a crucial machine learning technique used to analyse time-varying data, enabling the detection of early anomalies in the network or factors affecting system performance. Each node in the network usually has a time series for its performance indicators over time, such as the amount of traffic transmitted, delay, or error rate. Fluctuations in the time series can reflect abnormal node states, such as sudden increases in traffic or error rates. Analyzing these series not only helps to monitor the current state of the network but also helps to predict potential anomalies in the future. Time series data in the network system has distinct characteristics that need to be considered when building a classification model: Periodicity, Heterogeneity, Interdependence, and Gated Recurrent Unit (GRU). Currently, various methods are available for classifying time series. In this paper, we employ the Long Short-Term Memory (LSTM) method [6].

Anomalies in a node's time series can propagate through the links in the graph, affecting neighbouring nodes. Therefore, combining time series analysis and graph learning methods enables machine learning models to recognize more complex anomaly patterns, including those that are interconnected between nodes. If only one of the two models detects an anomaly or neither model detects it, the system may need to re-examine the information to avoid generating false alerts.

III. BUILDING AND EVALUATING A MODEL FOR THE NETWORK NODE ABNORMALITY MONITORING SÝTEM

Figure 1 illustrates the general model for monitoring network node anomalies.



[Fig.1: Overview Model of Network Node Abnormality Monitoring]

- *Data usage alert* is collected from network nodes every time the static warning system emits a signal. The network operator's static warning system continuously monitors the operation of network devices and issues warnings when detecting unusual events. Each alert is recorded with an alert error code and the time the alert occurred. This is a critical time frame for obtaining data related to the specific network node for the labelling process. In particular, the network operator's static warning system operates based on the static warning data set, which includes three main components:

i. *Device Information:* The alert system collects data from network devices, including end devices and intermediate devices such as routers and switches. Device information typically includes the device type (router, switch, firewall, or server, etc.), the IP address of each device in the network, the Device ID, Physical location, and Operating status (including parameters such as uptime, last reboot time, and average response time).

ii. *Traffic Information:* One of the most critical parameters for monitoring the network system's operation is data on the traffic transmitted between network nodes. This data includes indicators such as Total traffic, Number of packets (the Number of packets sent and received by each device within a specified period), Packet loss rate, Bandwidth used, and Latency.

iii. *Alert/Error Codes:* The carrier's static alert system also provides alert error codes, which help detect problems and abnormalities in network operations. These error codes often include Connection error codes, Performance error codes and Security error codes.

iv. *Alert Occurrence Time:* time recorded by the system at the time the system traffic reaches the configured warning threshold. The network operator's static alert system continuously monitors the operation of network devices and issues warnings when detecting unusual events. Each alert is recorded with an alert error code and the time it occurred. This is a critical time frame for obtaining data related to the specific network node for the labelling process.

A. Raw Traffic Data Set

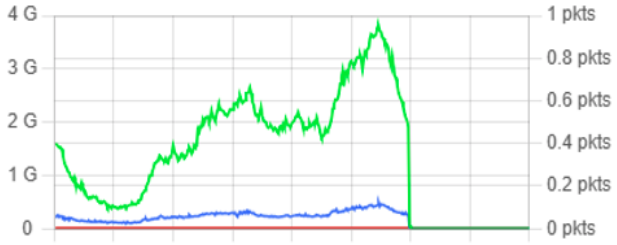
The network operator's static alert system continuously monitors the operation of network devices and issues alerts when anomalous events are detected. Each alert is recorded with an alert error code and the time it occurred. This is a critical time frame for obtaining data related to the specific network node for the labelling process. The collected raw data includes Traffic information (Traffic transmitted through the network node during this period), device operating parameters (Parameters such as CPU, RAM, latency, and the number of packets processed), and connection information (the connection status between network nodes, including bandwidth, latency, and packet loss).

To evaluate and label the data, the authors developed a tool to create a graph that explicitly represents the performance indicators of network nodes, thereby classifying the collected data as normal or abnormal accordingly. Figures 2 and 3 represent the traffic of a node labelled abnormal; Figure 3

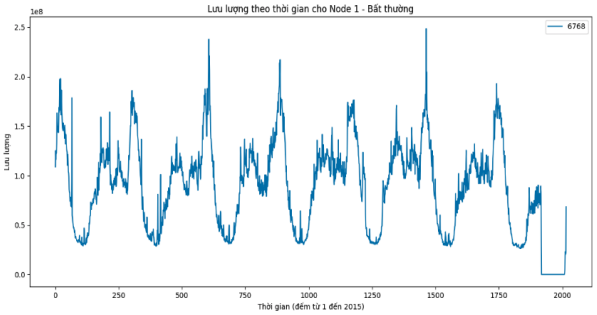
Retrieval Number: 100.1/ijitee.I112214090825
DOI: 10.35940/ijitee.I1122.14090825
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

represents the traffic of a node labelled normal.



[Fig.2: The Traffic of a Node in a State of Complete Bandwidth Reduction]



[Fig.3: Flow Chart of an Abnormal Node]

B. Data Preprocessing

The raw traffic dataset comprises all network node performance parameters, including missing, noisy, or inconsistent values, which can negatively impact the results of the machine learning model if not handled promptly. To qualify for the classification model (LSTM - Long Short-Term Memory), the data needs to be preprocessed through the following steps:

- **Filling in Missing Data:** While collecting raw data of network nodes, due to factors such as system errors, temporary disconnections, or technical problems, some values may be missing or not fully recorded. The data is collected every 5 minutes for 7 days, resulting in a large dataset. Filling in missing values in these cycles plays a vital role in ensuring the quality of input data for machine learning models.
- **Data Normalisation**: After filling in missing values, the next step in preprocessing is data normalisation. Normalization helps to make all features related to nodes and links consistent and accurate, making it easier for the machine learning model to process the data and avoiding features with large values dominating features with smaller values. In this study, the authors applied the min-max normalization method to bring all values to the range [0,1]. This method calculates the normalized value for each feature based on the formula:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

x' is the normalized value; x is the original value of the feature; x_{max} và x_{min} are the minimum and maximum values of the feature, respectively. Min-Max normalization helps preserve the linear relationships between values in the original data, while ensuring that the traffic features are on the same scope, facilitating machine learning model training.

- **Balancing the Training Dataset:** The number of samples of anomalies is often much lower than that of the non-anomalies. This can lead to bias in the machine learning model, which in turn affects the ability to detect anomalies. To address this issue, the authors employed the Synthetic Minority Over-sampling Technique (SMOTE) [3] to augment the minority class samples by generating synthetic samples from existing ones.
- **Ensuring Data Consistency**: After completing the above steps, the team re-examines the entire dataset to ensure consistency, including verifying that all network node and link features have the same structure and that there are no discrepancies in sample counts or collection times. Any inconsistencies found are adjusted or removed to prevent them from affecting the machine learning model.

An LSTM (Long Short-Term Memory) time series classification model will be applied to the time series data of each node to detect abnormalities in the node's operation. Each model will learn to predict the normal state of the nodes based on their historical data. When there is a significant difference between the expected value and the actual value, the model will detect abnormalities. After detecting abnormalities from the time series data, this information will be incorporated into the graph learning model. Models such as GCN (Graph Convolutional Network) [7] utilise information from neighbouring nodes to determine whether an abnormality at one node is spreading to others. Then, the Attention mechanism will be applied to calculate the output of the GCN (Graph Convolutional Network), helping the model focus on more critical features from the nodes, thereby improving its ability to detect abnormalities.

The model training process is performed on a preprocessed dataset, where the data is divided into two parts: 70% of the data is used for training, and the remaining 30% is used for testing. Table 1 below shows the results of evaluating the models on the test set:

Table-I. Results of Evaluating the Models on the Test Set

Model	Accuracy	Recall	Precision	F1-score
GCN	0.72	0.85	0.71	0.83
GATs	0.82	0.96	0.80	0.89

The results show that the GATs (Graph Attention Networks) model has better performance than GCN (Graph Convolutional Network) in most metrics, especially in Recall, which shows that GATs (Graph Attention Networks) have better ability to detect abnormal patterns thanks to the Attention mechanism.

IV. BUILDING A TRAFFIC MONITORING APPLICATION

The authors built an application with features as follows:

- **Real-Time Monitoring**: Users can monitor the status and traffic of network nodes in real-time, with alerts sent immediately when abnormalities are detected.
- **Search on Demand**: The application provides filters based on network node information, such as network node IP address, region, device name, and network node status
- **Viewing Network Node Traffic Details**: Mining operations

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)
© Copyright: All rights reserved.

engineers can view and monitor the traffic graph of each network node over a selected period.

- **History Storage**: The application stores past alerts, allowing users to revisit and analyze past network node alerts.
- **Tracking and Labelling Features**: Allowing users to mark alerts that need to be tracked, and relabel alerts that the model misclassifies
- **Detailed Reporting**: The application provides detailed reports on unusual events, including relevant parameters such as time of occurrence, predicted cause, and severity.

The application was tested on an internal test network. The test results in Table 1 show that the application is capable of detecting and alerting to abnormal events with an accuracy of 82%. The system is capable of processing and analysing data in real-time, with an average delay of 0.25 seconds. Figures 4, 5, and 6 are examples of the tested application.



[Fig.4: Main Application Interface]

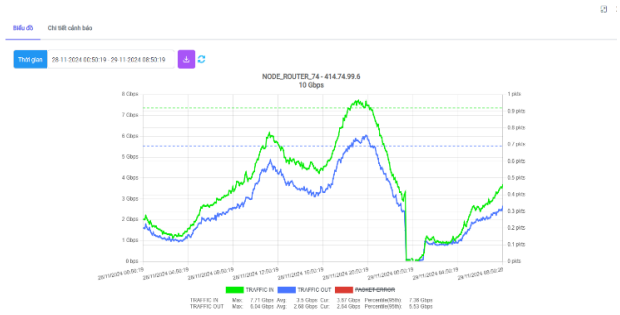


Figure 5. Network Node Traffic Viewing Interface



[Fig.6. Alert List Filter Interface]

V. CONCLUSION

The authors study the method of combining time series classification and graph learning methods in monitoring network node traffic, conducting experiments and evaluating the effectiveness of graph learning methods. Subsequently, they developed an application for monitoring abnormal network node traffic using graph learning methods, which helped improve the management and operation of telecommunications network systems in Vietnam. The system can detect abnormal patterns, predict upcoming incidents, and automatically make timely intervention recommendations. This helps improve the reliability, performance and safety of the network system, while minimizing operating and maintenance costs.

DECLARATION STATEMENT

After aggregating input from all authors, I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/Competing Interests:** Based on my understanding, this article does not have any conflicts of interest.
- **Funding Support:** This article has not been sponsored or funded by any organization or agency. The independence of this research is a crucial factor in affirming its impartiality, as it was conducted without any external influence.
- **Ethical Approval and Consent to Participate:** The data provided in this article is exempt from the requirement for ethical approval or participant consent.
- **Data Access Statement and Material Availability:** The data and materials supporting the findings of this article are publicly accessible.
- **Author's Contributions:** The authorship of this article is contributed equally to all participating individuals.

REFERENCES

1. Wu, L., Zhao, Q., Wei, W., Liu, J., & Li, X. (2025). The role of nodes in controlling and observing complex networks. PLOS ONE, 20(6), e0325824. DOI: https://doi.org/10.1371/journal.pone.0325824
2. Hassan Ismail Fawaz1, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller1, "Deep learning for time series classification: a review", DOI: https://arxiv.org/pdf/1809.04356,14May2019.
3. Mohanty, N., Behera, B. K., & Ferrie, C. (2025, January 31). Quantum SMOTE with Angular Outliers: Redefining Minority Class Handling. arXiv preprint. DOI: https://doi.org/10.48550/arXiv.2501.19001
4. Shaked Brody, Uri Alon, Eran Yahav, "How Attentive are Graph Attention Networks?", DOI: https://arxiv.org/abs/2105.14491,31Jan2022.
5. T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907v4, Feb. 2017. [Online]. Available: DOI: https://arxiv.org/abs/1609.02907v4
6. Y. Li, Y. Xu, Y. Cao, J. Hou, C. Wang, W. Guo, X. Li, Y. Xin, Z. Liu, and L. Cui, "One-class LSTM network for anomalous network traffic detection," Applied Sciences, vol. 12, no. 10, p. 5051, 2022. [Online]. Available: DOI: https://doi.org/10.3390/app12105051
7. Abu-El-Haija, S., Kapoor, A., Perozzi, B., & Lee, J. (2018, February 24). N-GCN: Multi-scale Graph Convolution for Semi-supervised Node Classification (arXiv:1802.08888). arXiv. DOI: https://doi.org/10.48550/arXiv.1802.08888.
8. E. Galstad, "Nagios Core: Open-source monitoring tool for hosts and services," Nagios Enterprises, 2025. [Online]. Available: https://www.nagios.org/projects/nagios-core/

AUTHOR'S PROFILE



Dr. Phan Thi Ha is currently a lecturer at the Faculty of Information Technology at the Posts and Telecommunications Institute of Technology (PTIT) in Vietnam, as well as at the Computing Fundamentals Department, FPT University, Hanoi, Vietnam. She received a B.Sc.In Math & Informatics, a M.Sc. in Mathematics, a Guarantee for Computer Systems and a PhD. in Information Systems in 1994, 2000 and 2013, respectively. Her research interests include machine learning, natural

© Copyright: All rights reserved.

language processing and mathematics applications.



This. Trinh Thi Van Anh is currently a lecturer and researcher at the Faculty of Information Technology, Posts and Telecommunications Institute of Technology (PTIT), Vietnam, as well as the Computing Fundamentals Department, FPT University, Hanoi, Vietnam. She received a B.Sc. in Electronics and Telecommunications in 1993 (HUST) and an M.Sc. in Computer Science in 1998 (HUST). Her research interests include machine learning, NLP and opinion mining.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of the Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)/ journal and/or the editor(s). The Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP) and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

© Copyright: All rights reserved.