Journal of Industrial Engineering and Applied Science
Journal Home: http://jieas.suaspress.org/ | CODEN: JIEAAE
Vol. 3, No. 2, 2025 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

Trojan Virus Detection and Classification Based on Graph Convolutional Neural Network Algorithm

REN, Wenkun [1*] XIAO, Xingpeng [2] XU, Jian [3] CHEN, Heyao [4] ZHANG, Yaomin [5] ZHANG, Junyi [6]

[1] Illinois Institute of Technology, USA
[2] Shandong University of Science and Technology, China
[3] University of Southern California, USA
[4] Beijing University of Posts and Telecommunications, China
[5] University of San Francisco, USA
[6] Lawrence Technological University, USA

**** REN, Wenkun is the corresponding author, E-mail: charlsiexno9@gmail.com***

Abstract: This article proposes a new method for Trojan virus detection and classification based on graph convolutional neural network (GCN) algorithm. By observing the performance evaluation indicators of the model during the training process, the accuracy of the model continued to improve from the initial 64.27% to 88.28% and gradually stabilized, proving that the model can effectively identify Trojan viruses during the training process. In addition, confusion matrix analysis based on the training set shows that the model performs quite well in classification tasks, with an overall accuracy of 91.06%, precision of 89.24, recall of 92.63, and F1 score of 90.91. These indicators indicate that the model can demonstrate good performance in detecting Trojan viruses from various perspectives. On the test set, the model also demonstrated excellent performance, with an accuracy rate of 90.96%, an accuracy rate of 90%, a recall rate of 91.43%, and an F1 score of 90.71. By analyzing the confusion matrix of the test dataset, it can be seen that the classification performance of the model in practical applications is similar to that on the training set, further verifying its good generalization ability. In summary, the experimental results in this article demonstrate that the Trojan virus detection method based on graph convolutional neural networks has high accuracy and stability, and demonstrates superior performance compared to traditional detection methods. This method provides new ideas and technical support for Trojan virus detection in the field of network security, which can effectively respond to increasingly complex network security threats and provide theoretical basis and practical guidance for related research and applications. Through further optimization and improvement, the method proposed in this article is expected to play a greater role in future Trojan virus detection, helping to enhance the level of network security protection.

Keywords: Trojan Virus, Graph Convolutional Neural Network, Virus Detection and Classification.

Disciplines: Computer Science. **Subjects:** Cybersecurity.

DOI: https://doi.org/10.70393/6a69656173.323735 **ARK:** https://n2t.net/ark:/40704/JIEAS.v3n2a01

1 INTRODUCTION

Against the backdrop of rapid development of information technology, network security has become a global issue. As an important form of cyber attack, Trojan viruses have attracted deep attention from scientific researchers and information security experts due to their concealment and harmfulness. Trojan viruses usually disguise themselves as legitimate software, and after users install them without their knowledge, hackers can remotely manipulate computers, steal sensitive information, or implement other malicious functions. Therefore, early and effective detection of Trojan programs is crucial.

With the increasingly complex network environment, the variants and transmission methods of Trojan viruses are also constantly evolving. Traditional feature-based defense mechanisms are gradually becoming inadequate in dealing with new and mutated Trojan programs. Many new Trojan programs can bypass these traditional detection techniques, leading to serious threats to the system. Therefore, researchers urgently need to find new detection methods to improve the accuracy and efficiency of Trojan detection. In recent years, the rise of intelligent machine learning technology has provided an opportunity to solve this problem.

Machine learning algorithms provide powerful tools for Trojan detection. These algorithms have the ability to analyze large datasets, discover potential patterns, and perform classification, making them a core component of modern Trojan detection systems. Compared with traditional methods, machine learning not only improves the accuracy of detection, but also significantly enhances the recognition ability of new

Trojan programs[1]. Machine learning algorithms can handle massive amounts of network traffic data. Trojan programs often transmit and spread through networks, and analyzing network traffic is an effective detection method. Machine learning models, especially deep learning models, can automatically extract features from network traffic and identify abnormal patterns. This ability enables the system to proactively identify potential Trojan attacks, rather than relying solely on post hoc analysis[2]. The adaptability of machine learning enables it to update and improve models in a timely manner. In environments where Trojan programs frequently mutate, the generalization ability of the model is particularly important[3]. By continuously learning historical data, machine learning models can automatically adjust their parameters to adapt to new attack methods[4]. For example, when a new Trojan virus appears, the model can be retrained to recognize this new type of threat. This simulation capability enables machine learning to keep pace with the times and always be at the forefront of network security[5]. This article is based on the graph convolutional neural network algorithm to detect and classify Trojan viruses, providing a new method for Trojan virus detection.

2 DATA SOURCES

In terms of data selection, we chose an open-source dataset that has been validated multiple times and can verify the classification performance of the model. The dataset is the open-source dataset of Kaggle website, which includes multiple network monitoring indicators and their final categories. There are two types in total, one is Trojan virus and the other is Benign[6]. Display the partial dataset as shown in Table 1.

TABLE 1.SELECTED PARTIAL DATASET.

Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Class
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Benign
0	0	0	0	0	0	0	0	Trojan
322594	0	322594	322594	60306983	0	60306983	60306983	Benign
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Benign
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Trojan
0	0	0	0	0	0	0	0	Benign

3 METHOD

Graph Convolutional Neural Network (GCN) is a deep learning model specifically designed to handle graph structured data, aiming to capture features of nodes and their neighborhoods in a graph through convolution operations. Unlike traditional Convolutional Neural Networks (CNNs), which are mainly applied to regular grid data such as images, graph data typically has irregular connection patterns, making standard convolution operations difficult to apply directly. GCN creatively achieves effective feature extraction of irregular graph data by introducing the adjacency matrix of the graph and combining it with the characteristics of the Laplacian matrix[7]. The two schematic diagrams of GCN are shown in Figure 1.

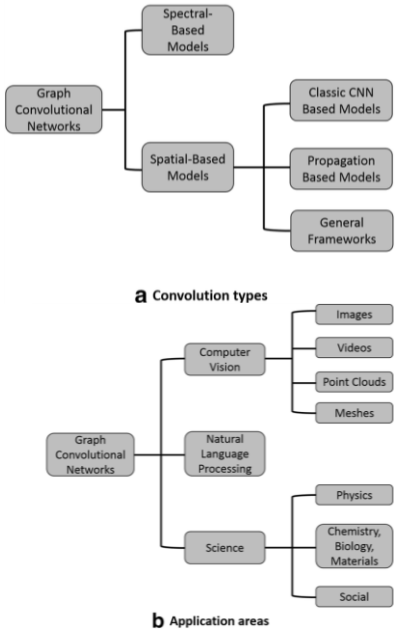

FIGURE 1. THE TWO SCHEMATIC DIAGRAMS OF GCN

Journal of Industrial Engineering and Applied Science
Journal Home: http://jieas.suaspress.org/ | CODEN: JIEAAE
Vol. 3, No. 2, 2025 | ISSN 3005-6071 (Print) | ISSN 3005-608X (Online)

The core idea of GCN is to update the representation of nodes by aggregating features of their neighbors[8]. Specifically, GCN utilizes the adjacency matrix and node feature matrix of the graph, and through multiple layers of convolution operations, enables the representation of each node at each layer to fuse the information of its neighboring nodes. In each layer, the feature vectors of nodes are aggregated through multiplication with the adjacency matrix, thus completing the transmission of information. This method not only improves the expressive power of nodes, but also enables the model to capture higher-order graph structural features. The basic structural form of GCN is shown in Figure 2.

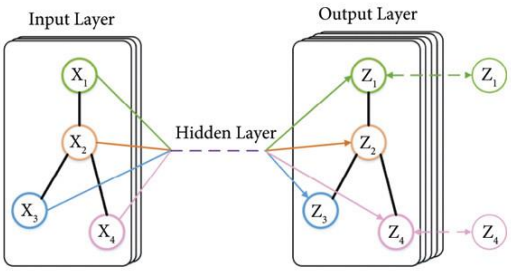

FIGURE 2. THE BASIC STRUCTURAL FORM OF GCN

During the model training process, GCN usually uses backpropagation algorithm to optimize the network. In order to improve the generalization ability of the model, GCN introduces graph regularization techniques to avoid overfitting of the model. In practical applications, GCN can handle various tasks such as node classification, link prediction, and graph classification, and with its efficient feature aggregation mechanism, it can run on large-scale graph datasets. In some specific application scenarios, such as social network analysis, recommendation systems, and bioinformatics, GCN has demonstrated excellent performance.

However, GCN also has some limitations. For example, due to the fact that the aggregation process of information is based on adjacency matrices, GCN has a higher computational complexity when dealing with sparse graphs with a large number of nodes. In addition, GCN may experience over smoothing when faced with a graph depth that is too large, resulting in feature representations of different nodes becoming too similar and affecting the model's discriminative ability. Therefore, researchers have proposed various improvement strategies, such as Graph Attention Mechanism (GAT) and Graph Convolution Extension, aimed at overcoming these challenges and enhancing the application effectiveness of GCN.

4 RESULT

This article uses graph convolutional neural networks as a model for machine learning classification of Trojan viruses. In terms of experimental parameter settings, the learning rate is set to 0.01, the weight decay is 5e-4, the optimizer uses Adam, the dropout ratio is set to 0.5, the number of hidden layers is 2, each layer has 32 dimensions, and the training times are 500.

For hardware configuration, this experiment uses NVIDIA RTX 3060 GPU, 8GB video memory, and Intel Core i7 CPU. The memory is 32GB, and the SSD storage is 512GB. In terms of software, Ubuntu 18.04 is chosen as the operating system, and Matlab R2022a is used as the programming language.

Import the dataset in a 6:4 ratio for training, and output the change curves of accuracy and loss during the training process, as shown in Figure 3.

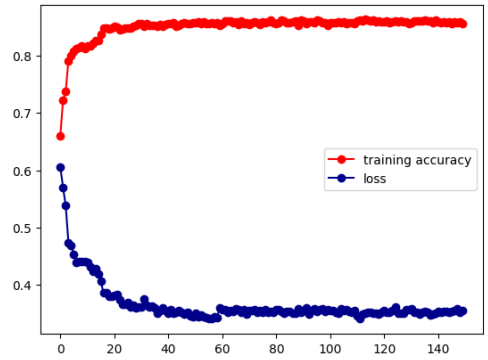

FIGURE 3. THE CHANGE CURVES OF ACCURACY AND LOSS DURING THE TRAINING PROCESS.

From the changes in accuracy and loss during the training process, it can be seen that the accuracy gradually increased from the initial 64.27% to 88.28% and tended to converge. The accuracy eventually stabilized at around 88%, proving that the model can detect Trojan viruses well during the training process.

Output the Trojan virus classification confusion matrix of the training set, as shown in Figure 4. According to the confusion matrix of the training set, the classification accuracy of the model is 91.06%, Precision is 89.24%, Recall is 92.63%, and F1 Score is 90.91%.

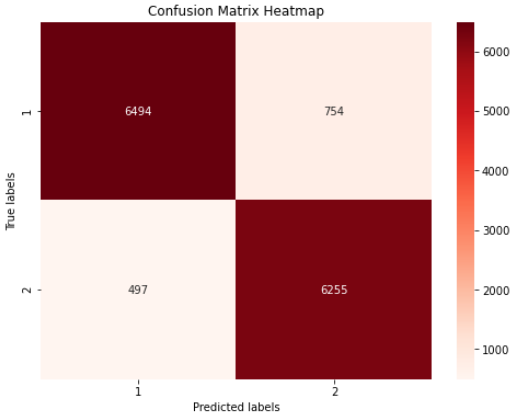

FIGURE 4. THE CONFUSION MATRIX.

Output the confusion matrix for Trojan virus detection

classification in the test set, as shown in Figure 5. According to the confusion matrix of the test set, the accuracy, precision, and recall of the model are 90.96%, 90%, and 91.43%, respectively, F1 Score: It is 90.71%. The graph neural network model in this article also showed good performance on the test set, with excellent generalization ability.

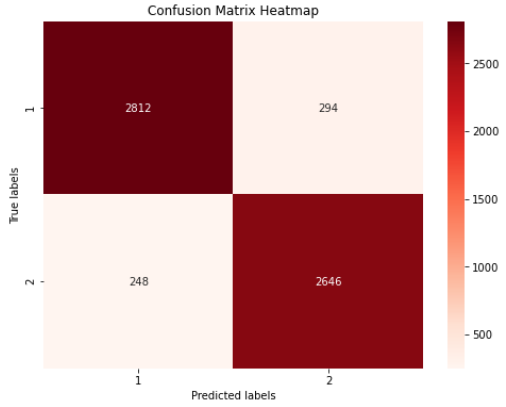

FIGURE 5. THE CONFUSION MATRIX.

5 CONCLUSION

This article explores a new method based on Graph Convolutional Neural Network (GCN) algorithm for detecting and classifying Trojan viruses. During the training process, we observed changes in the accuracy and loss of the model, and found that the accuracy gradually increased from 64.27% to 88.28%, and tended to converge, eventually stabilizing at around 88%[9]. This indicates that during the training period, the model gradually gained strong ability to effectively detect Trojan viruses, demonstrating good learning performance.

Further analyzing the confusion matrix of the training set, we obtained multiple indicators of the model in classification tasks: classification accuracy of 91.06%, precision of 89.24%, recall of 92.63%, and F1 score of 90.91%. These results fully demonstrate the excellent performance of the model in identifying Trojan viruses, accurately distinguishing Trojan viruses from normal samples in various situations, and improving the overall performance of the detection system. Meanwhile, these indicators also indicate that the model maintains a certain balance across samples of different categories, ensuring effective detection of various types of Trojan viruses.

For the performance of the test set, we output the corresponding confusion matrix, which shows that the accuracy of the model on the test set is 90.96%, the precision is 90%, the recall is 91.43%, and the F1 Score is 90.71%[10]. These test results further validate the effectiveness of the model, indicating that it not only performs well on training data, but also has strong generalization ability on unseen data. This phenomenon demonstrates that our designed graph convolutional network can successfully capture the potential features of Trojan viruses, enabling accurate identification in different samples[11].

Through the analysis of the training and testing sets, we can conclude that the graph convolutional neural network model proposed in this study has achieved satisfactory results in the task of Trojan virus detection and classification[12]. In summary, this article provides an effective new method for detecting Trojan viruses, demonstrating the enormous potential of graph convolutional neural networks in the field of security applications.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

REN, Wenkun

Information Technology and Management, Illinois Institute of Technology, Chicago, USA.

XIAO, Xingpeng

Computer Application Technology, Shandong University of Science and Technology, Qingdao, China.

XU, Jian

Electrical and Electronics Engineering, University of Southern California, Angeles, USA.

CHEN, Heyao

Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing, China.

ZHANG, Yaomin

Computer Science, University of San Francisco, San Francisco, USA.

ZHANG, Junyi

Electrical and Computer Engineering, Lawrence Technological University, Houston, USA.

REFERENCES

[1] Chen, H., Shen, Z., Wang, Y. and Xu, J., 2024. Threat Detection Driven by Artificial Intelligence: Enhancing Cybersecurity with Machine Learning Algorithms.

[2] Ullah, S., Ahmad, T., Buriro, A., Zara, N., & Saha, S. (2022). TrojanDetector: a multi-layer hybrid approach for trojan detection in android applications. Applied Sciences, 12(21), 10755.

[3] Liang, X., & Chen, H. (2019, July). A SDN-Based Hierarchical Authentication Mechanism for IPv6 Address. In 2019 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 225-225). IEEE.

[4] Nenov, L., Kassev, K., & Chanev, D. (2021, September). Investigation of algorithms for virus detection using neural networks and machine learning. In 2021 Sixth Junior Conference on Lighting (Lighting) (pp. 1-4). IEEE.

[5] Liang, X., & Chen, H. (2019, August). HDSO: A High-Performance Dynamic Service Orchestration Algorithm in Hybrid NFV Networks. In 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 782-787). IEEE.

[6] Riadi, I., & Aprilliansyah, D. (2023). Analysis of Anubis Trojan Attack on Android Banking Application Using Mobile Security Labware. International Journal of Safety & Security Engineering, 13(1).

[7] Chen, H., & Bian, J. (2019, February). Streaming media live broadcast system based on MSE. In Journal of Physics: Conference Series (Vol. 1168, No. 3, p. 032071). IOP Publishing.

[8] Kanaker, H., Karim, N. A., Awwad, S. A., Ismail, N. H., & Zraqou, J. (2022). Trojan Horse Infection Detection in Cloud Based Environment Using Machine Learning. International Journal of Interactive Mobile Technologies, 16(24).

[9] Zanluqui, N. G., Oliveira, L. G., Polonio, C. M., França, T. T., De Souza, G. P., Muraro, S. P., ... & Peron, J. P. S. (2021). Zika Virus Infection of Murine and Human Neutrophils and Their Function as Trojan Horses to the Placenta. BioRxiv, 2021-09.

[10] Hendrawan, A. H., Kurniawan, R., Aprian, A. J., Primasari, D., & Subchan, M. (2024). Enhancing Cybersecurity Through Live Forensic Investigation of Remote Access Trojan Attacks using FTK Imager Software. International Journal of Safety & Security Engineering, 14(1).

[11] Xu, J., Chen, H., Xiao, X., Zhao, M., Liu, B. (2025). Gesture Object Detection and Recognition Based on YOLOv11.Applied and Computational Engineering,133,81-89.

[12] Chiwariro, R., & Pullagura, L. (2023). Malware detection and classification using machine learning algorithms. Int J Res Appl Sci Eng Technol, 11(8), 1727-1738.