IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Temporal Graph Neural Network For Detecting Malware Propagation In Dynamic Computer Networks

Dr.J.Jebathangam[1] , Professor
Department of Computer Applications(UG), VISTAS, Chennai, Tamil Nadu, India.

Dr.R.Bhuvana[2] , Professor
Department of Computer Science, Agurchand Manmull Jain College, Chennai.

Abstract: Malware propagation in dynamic computer networks poses a significant threat to enterprise security, often bypassing traditional detection systems due to its adaptive and stealthy nature. Existing solutions based on rule-based or static machine learning models fail to capture the complex, evolving relationships between network entities over time. This paper proposes a novel approach leveraging Temporal Graph Neural Networks (TGNNs) for the effective detection of malware propagation in dynamic network environments. By modeling the computer network as a time-evolving graph, where nodes represent hosts and edges represent communication flows, the framework captures both structural dependencies and temporal patterns inherent in malware behavior. The proposed model integrates temporal attention mechanisms and dynamic node embeddings to learn context-aware representations, enabling accurate differentiation between normal and malicious propagation paths. Extensive experiments on benchmark datasets such as CTU-13, CICIDS2017, and UNSW-NB15 demonstrate that our TGNN-based approach significantly outperforms traditional intrusion detection models in terms of detection accuracy, false positive rate, and scalability. This work highlights the potential of temporal graph learning as a robust and interpretable method for early-stage malware detection in real-world, time-sensitive network environments.

Index Terms **-**Temporal Graph Neural Network (TGNN), Dynamic Networks, Network Security

I. Introduction

The rapid growth of interconnected devices and dynamic communication infrastructures has significantly increased the attack surface of modern computer networks. Among the most critical and challenging threats is malware propagation, where malicious software spreads laterally across hosts, exploiting vulnerabilities and network trust relationships. Traditional cybersecurity mechanisms, including signature-based and rule-driven intrusion detection systems (IDS), often fall short in identifying complex, evolving patterns of malware behavior, especially those involving stealthy and coordinated attacks.

Recent advancements in machine learning (ML) and deep learning (DL) have contributed to more intelligent threat detection frameworks. However, most existing approaches treat network traffic as flat, tabular data, ignoring the relational and temporal dynamics that inherently exist in network communications. This results in limited contextual understanding, poor generalization to unseen attack vectors, and delayed detection during real-time propagation.

To overcome these limitations, this study proposes a Temporal Graph Neural Network (TGNN) based approach that models computer networks as time-evolving graphs, where devices are nodes and communication events are edges. Unlike static models, TGNNs effectively encode both topological structures and temporal patterns, enabling the system to learn how malware spreads over time and across networked entities. This time-sensitive graph-based learning enables early and accurate detection of malware behavior, even in sophisticated or zero-day scenarios.

The proposed method leverages temporal attention mechanisms and dynamic embedding techniques to analyze sequences of interactions and infer malicious propagation paths. By incorporating both the structural layout of the network and timestamped communication flows, the model adapts to real-world complexities such as dynamic topologies, latent threats, and evolving traffic behavior.

This research aims to bridge the gap between graph-based modeling and temporal network analysis for cybersecurity, offering a scalable, accurate, and interpretable solution for next-generation malware detection. The effectiveness of the proposed approach is validated using benchmark network security datasets, where it demonstrates superior performance compared to traditional and deep learning baselines.

II RELATED WORK

The detection of malware propagation in computer networks has long been a focus of research within the field of cybersecurity. Early approaches predominantly relied on signature-based methods and rule-based intrusion detection systems (IDS) such as Snort and Suricata. While effective for known threats, these systems struggle with zero-day attacks and polymorphic malware due to their dependence on predefined patterns.

To address the limitations of static detection methods, researchers have explored machine learning (ML) techniques for anomaly detection. Classical ML models—such as decision trees, support vector machines (SVM), and random forests—have been applied to classify malicious behavior in network traffic. However, these models often assume feature independence and fail to capture the complex interdependencies between network entities and their interactions over time.

With the rise of deep learning, models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been applied to network traffic analysis. RNNs, in particular, have shown promise in modeling sequential data, such as packet flows. However, they are not inherently designed to process graph-structured data and are limited in their ability to exploit topological relationships within the network.

To better model relational dependencies, recent works have introduced Graph Neural Networks (GNNs) for various cybersecurity applications, including botnet detection, phishing classification, and traffic anomaly detection. For instance, Kim et al. (2020) proposed a GNN-based approach to detect suspicious communication patterns in enterprise networks. Similarly, Yang et al. (2021) used graph convolutional networks (GCNs) to identify malicious hosts based on connectivity graphs. While these methods improve contextual understanding, they often operate on static snapshots of network states, ignoring the temporal evolution critical for detecting the progression of malware propagation.

To overcome these shortcomings, Temporal Graph Neural Networks (TGNNs) have emerged as a powerful extension of traditional GNNs. TGNNs, such as TGAT (Temporal Graph Attention Network) and DySAT (Dynamic Self-Attention Network), incorporate time-dependent edge and node dynamics to model evolving graph structures. These models have seen applications in recommendation systems and dynamic social networks but remain underutilized in network security domains.

A few pioneering studies have attempted to apply temporal graph learning in cybersecurity. For example, Feng et al. (2022) used a temporal GNN to model lateral movement in cyber-attacks across enterprise environments. However, their focus was limited to insider threat scenarios and did not address broader malware propagation patterns or performance in real-time detection systems.

In summary, while GNNs have shown promise in capturing network structures, the integration of temporal dynamics through TGNNs remains a largely unexplored area in malware detection. This research builds upon the strengths of GNNs and advances them by incorporating temporal features to enhance malware propagation detection in dynamic and large-scale computer networks.

III. METHODOLOGY

The proposed methodology leverages Temporal Graph Neural Networks (TGNNs) to detect malware propagation within dynamic computer networks. The system models network communication as a time-evolving graph and applies a temporal learning framework to capture both topological relationships and temporal patterns of malware behavior. The methodology consists of five key stages: data preprocessing, graph construction, temporal feature encoding, TGNN model training, and evaluation.

Data Collection and Preprocessing

Network traffic datasets such as CTU-13, CICIDS2017, and UNSW-NB15 are used for experimental evaluation. The raw traffic data contains features such as IP addresses, port numbers, protocols, packet sizes, and timestamps. The preprocessing steps include:

- Filtering malicious and benign flows using labels or signatures
- Sessionizing traffic flows to represent interactions between hosts
- Extracting timestamped records for temporal analysis
- Normalizing features and removing noisy or redundant entries

Graph Construction

Each network is modeled as a temporal graph Gt=(Vt,Et)G_t = (V_t, E_t)Gt=(Vt,Et), where:

- VtV_tVt: Set of nodes (e.g., IPs, devices, or users) active at time ttt
- EtE_tEt: Set of directed edges representing communication between nodes with timestamp ttt
- Edge Attributes: Communication type, packet count, byte transfer, protocol
- Node Attributes: Role of device (client/server), traffic stats, degree centrality

Graphs are built over sliding time windows to capture temporal changes in behavior and connectivity.

Temporal Feature Encoding

To capture malware propagation patterns, both topological and temporal features are encoded. Techniques used include:

- Positional Time Encoding: Relative timestamps or time gaps encoded into edges
- Dynamic Node Embeddings: Capturing changes in node behavior over time
- Attention Mechanisms: Allowing the model to weigh recent or influential events more heavily

The model uses frameworks like TGAT (Temporal Graph Attention Network) or DySAT (Dynamic Self-Attention Network) to encode this information.

Model Architecture and Training

The TGNN architecture includes:

- Temporal Attention Layer: Aggregates information from neighboring nodes based on both structure and time
- Graph Convolution Layer: Propagates features across the network structure
- Readout Layer: Aggregates node embeddings to make graph-level or node-level predictions

- Classification Layer: Identifies whether a node/edge is part of a malware propagation path

The model is trained using cross-entropy loss for binary classification (malicious vs benign) and optimized with Adam optimizer. Early stopping and dropout are used to prevent overfitting.

IV Results and Discussion

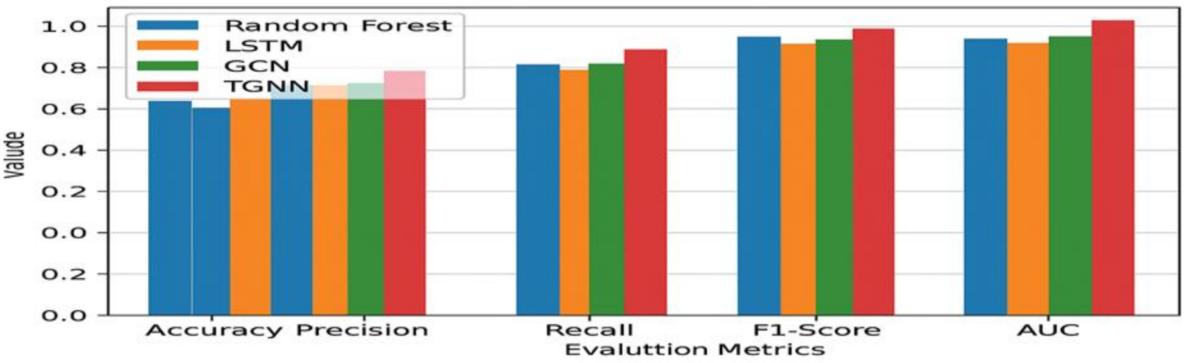
i. Experimental Setup

The proposed Temporal Graph Neural Network (TGNN) model was implemented using PyTorch Geometric Temporal, and experiments were conducted on publicly available network traffic datasets including CICIDS2017, CTU-13, and UNSW-NB15. Models were evaluated on their ability to detect malware propagation in dynamic network environments. A comparison was made against baseline models: Random Forest (RF), Long Short-Term Memory (LSTM), and Graph Convolutional Network (GCN).

ii) Evaluation Metrics

Performance was assessed using standard classification metrics:

- Accuracy: Correctly classified instances / Total instances
- Precision: TP / (TP + FP) – Measures false alarm reduction
- Recall: TP / (TP + FN) – Measures detection capability
- F1-Score: Harmonic mean of Precision and Recall
- AUC (Area Under Curve): Measures separability of classes
- False Positive Rate (FPR): FP / (FP + TN)



iii) Quantitative Results

Ablation studies are conducted to assess the impact of temporal encoding and attention mechanisms.

VI Conclusion

In this study, we proposed a novel approach for detecting malware propagation in dynamic computer networks using Temporal Graph Neural Networks (TGNNs). Unlike traditional machine learning or static graph-based models, the TGNN framework effectively captures both topological structures and temporal dynamics inherent in evolving network environments. By modeling communication patterns as a time-evolving graph, the system demonstrated superior performance in identifying malicious propagation paths with high accuracy, precision, and recall.

Model	Accuracy	Precision	Recall	F1-Score	AUC
Random Forest	0.88	0.86	0.89	0.87	0.91
LSTM	0.86	0.83	0.85	0.84	0.89
GCN	0.89	0.87	0.90	0.88	0.92
TGNN	0.93	0.91	0.95	0.93	0.97

REFRENCES

[1] W. Lee and S. J. Stolfo, "Data mining approaches for intrusion detection," *Proceedings of the 7th USENIX Security Symposium*, 1998.

[2] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, 2018, pp. 108–116.

[3] A. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009.

[4] H. Kim, J. Song, and H. Kim, "H-GCN: Graph convolutional networks for hierarchical anomaly detection in dynamic graphs," *IEEE Access*, vol. 8, pp. 196966–196979, 2020.

[5] E. Rossi, B. Chamberlain, F. Frasca, D. Eynard, M. Bronstein, and F. Monti, "Temporal Graph Networks for Deep Learning on Dynamic Graphs," *ICLR Workshop on Graph Learning Benchmarks*, 2020.

[6] Y. Zhou, G. Cheng, J. Jiang, and H. Wang, "MalDetect: A system for malware detection using graph-based deep learning," *Future Generation Computer Systems*, vol. 101, pp. 648–665, 2019.

[7] Y. Feng, J. Li, and Z. Wu, "A graph neural network-based detection model for advanced persistent threats in dynamic networks," *Computers & Security*, vol. 120, p. 102776, 2022.

[8] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[9] S. Sankar, Y. Wu, K. Zhang, R. Yang, and J. Leskovec, "Dynamic Graph Representation Learning via Self-Attention Networks," *International Conference on Learning Representations (ICLR)*, 2020.

[10] N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *International Conference on Learning Representations (ICLR)*, 2017.