The OSCAR PUBLISHING Services
American Journal of Applied Science and Technology

VOLUME Vol.06 Issue 08 2026
PAGE NO. 77-84
DOI-10.37547/ajast/Volume06Issue08-07

An Advanced Graph-Based Deep Learning Approach for Cyberattack Detection in Cloud Computing Networks

Nguyen Minh Anh

Faculty of Information Technology, Vietnam National University, Hanoi, Vietnam

Received: 01 July 2026; **Accepted:** 17 July 2026; **Published:** 13 August 2026

Abstract: The increasing structural complexity of cloud computing networks has created a need for cyberattack detection techniques capable of representing relationships among users, workloads, services, virtual machines, communication flows, and security events. Conventional detection approaches frequently treat network observations as independent records, limiting their ability to capture relational dependencies that may characterize coordinated or multi-stage attacks. This paper proposes an advanced graph-based deep learning approach in which cloud-network entities are represented as nodes, their interactions as edges, and security-relevant observations as graph attributes. The methodological foundation integrates graph representation, learned heuristic reasoning, structured state-space analysis, and neural learning principles derived from the supplied literature. The approach is conceptually positioned between classical graph-search methods and modern neural architectures, enabling contextual threat identification while preserving structural information. The study develops a graph construction mechanism, graph-based representation learning process, cyberattack classification layer, and adaptive threat-prioritization mechanism. Theoretical analysis indicates that graph-based modeling can improve contextual interpretation of attacks because suspicious behavior is evaluated not only from individual events but also from their surrounding relational structure. The proposed framework further incorporates heuristic concepts from planning research to support efficient exploration of large and complex attack states. Findings suggest that combining graph structure with deep representation learning provides a stronger foundation for cloud cyberattack detection than isolated event classification, although computational complexity, graph construction quality, adversarial manipulation, and model interpretability remain important limitations. The framework extends the graph-based cybersecurity direction established by Marri et al. (2025) by emphasizing an integrated architectural and analytical perspective.

Keywords: Research Productivity, Higher Education, Regression Analysis, Deep Learning, Predictive Modeling, Educational Analytics, Artificial Intelligence, Faculty Performance.

1. INTRODUCTION

Background

Cloud computing networks are inherently relational systems. Users interact with applications, applications communicate with services, services access virtual machines and storage resources, and network components continuously exchange data. Consequently, cyberattacks rarely occur as isolated events. A compromised credential may lead to unauthorized service access, lateral movement, abnormal resource consumption, and subsequent data-exfiltration activity. Detecting such behavior requires a representation capable of preserving dependencies among heterogeneous entities.

Graph-based learning provides an appropriate theoretical representation because a network can naturally be modeled as a graph (G=(V,E)), where (V) represents entities and (E) represents interactions. In a cloud environment, nodes may correspond to users, containers, virtual machines, APIs, services, IP addresses, or security events, while edges can represent communication, authentication, access,

dependency, or temporal relationships. The central premise of this research is that the security meaning of an event depends partly on its position within this relational structure.

Earlier graph-search research demonstrated the importance of representing states and transitions explicitly. Doran and Michie (1966), for example, established an early computational foundation for graph traversal, while Bonet and Geffner (2001) demonstrated how heuristic search could efficiently navigate large planning spaces. Although these studies were not designed for cybersecurity, their underlying concepts are relevant to cloud threat analysis because attack progression can similarly be viewed as movement through interconnected states.

Recent research has increasingly connected deep learning with structured reasoning and graph-oriented representations. Dong et al. (2019) demonstrated neural mechanisms for reasoning over logical structures, while Agostinelli et al. (2019) showed that deep learning combined with search can solve complex structured problems. Within cybersecurity, Marri et al. (2025) specifically investigated a graph-based deep learning model for identifying cyber threats in cloud platforms. The present study builds theoretically upon this direction by developing a more explicit framework connecting graph construction, deep representation learning, state-aware reasoning, and cyberattack classification.

1.2 Problem Statement

A fundamental weakness of conventional cloud intrusion detection is the tendency to analyze network events as independent observations. Such approaches may identify abnormal traffic volume or unusual authentication behavior but can overlook the relationship between multiple apparently benign activities. An attacker may distribute malicious behavior across different entities and time periods, making individual observations difficult to classify.

The research problem is therefore formulated as follows: How can graph-based deep learning be architected to exploit structural relationships among cloud-network entities for more context-aware cyberattack detection while maintaining computationally feasible analysis of large network states?

1.3 Research Relevance

The relevance of the proposed approach emerges from two complementary research traditions. First, graph-search and planning research establishes methods for navigating large structured state spaces through heuristic guidance (Bonet & Geffner, 2001; Arfaee et al., 2011). Second, neural learning research demonstrates that learned representations can approximate complex decision functions and reasoning processes (Ferber et al., 2020; Ferber et al., 2022). Integrating these perspectives offers a theoretical basis for cyberattack detection in which graph topology provides context and deep learning provides adaptive feature extraction.

1.4 Objectives

The primary objective is to develop a conceptual graph-based deep learning architecture for cyberattack detection in cloud computing networks. Specifically, the study aims to formulate a graph representation of cloud security events, design a multi-stage graph-learning mechanism, incorporate structural and heuristic reasoning into threat identification, and critically assess expected detection behavior, scalability, and limitations.

1.5 Scope and Significance

The study focuses on network-level and cloud-resource-level cyberattack identification. It considers heterogeneous nodes, relational edges, graph attributes, learned representations, and attack-state transitions. It does not claim a particular empirical performance value because no experimental dataset or measured benchmark results were supplied. Accordingly, the Results section reports analytically derived findings from the proposed architecture rather than fabricated experimental statistics.

2. LITERATURE REVIEW

Graph-based problem solving has a long theoretical history. Doran and Michie's (1966) Graph Traverser demonstrated that computational systems can explore

interconnected states to reach objectives. This principle is relevant to cybersecurity because an attack can be represented as a sequence of transitions between security states. However, simple traversal is insufficient for contemporary cloud environments because the number and heterogeneity of possible states can become extremely large.

Planning research subsequently addressed this scalability problem through heuristic search. Bonet and Geffner (2001) established planning as heuristic search, demonstrating the importance of estimating the distance between a current state and a desired state. Bonet (2013) further examined admissible heuristics for SAS+ planning, while Bäckström and Nebel (1995) analyzed complexity properties of SAS+ planning. Together, these works indicate that structured state representations provide analytical advantages but also introduce computational challenges.

Pattern databases offered another mechanism for handling large state spaces. Culberson and Schaeffer (1998) introduced pattern databases as a means of obtaining useful heuristic information, and Edelkamp (2001) applied pattern databases to planning. Their significance for cloud security lies in the possibility of constructing reduced representations of attack-relevant graph states rather than processing every possible state simultaneously.

Arfaee et al. (2011) shifted attention toward learning heuristic functions for large state spaces. Garrett et al. (2016) similarly investigated learning to rank for synthesizing planning heuristics. These studies are particularly relevant to cyberattack detection because the importance of graph paths or security states can potentially be learned rather than defined exclusively through manually engineered rules.

Neural approaches provide a complementary direction. Ferber et al. (2020) examined neural-network heuristics for classical planning and the effect of hyperparameters, while Ferber et al. (2022) investigated bootstrapping and comparisons among neural heuristic methods. Gehring et al. (2022) conceptualized heuristics as dense reward generators within reinforcement learning for planning. These studies support the proposition that learned models can transform complex structured states into useful decision signals.

Agostinelli et al. (2019) demonstrated the integration of deep reinforcement learning and search for solving a highly complex structured problem. Dong et al. (2019) developed Neural Logic Machines for relational reasoning, providing further evidence that neural architectures can operate over structured symbolic or relational information. Frances et al. (2017) and Francés et al. (2018) emphasized structured planning descriptions and modeling frameworks, reinforcing the value of explicit representations.

The closest supplied cybersecurity contribution is Marri et al. (2025), which investigated a graph-based deep learning model for identifying cyber threats in cloud platforms. The present framework extends that research direction conceptually by emphasizing a unified pipeline in which cloud events are transformed into graph states, graph embeddings capture relational dependencies, learned heuristics prioritize suspicious regions, and a classification mechanism assigns threat categories. The principal research gap is therefore not the absence of graph-based cybersecurity modeling, but the need for a more explicit theoretical integration of graph representation, learned state prioritization, and deep threat classification.

3. METHODOLOGY

3.1 Proposed Research Architecture

The proposed architecture consists of five interconnected layers: cloud-event acquisition, heterogeneous graph construction, graph representation learning, structural threat reasoning, and cyberattack classification.

The general architecture can be expressed as:

[

Cloud\ Events \rightarrow Graph\ Construction \rightarrow Graph\ Embedding \rightarrow Threat\ Reasoning \rightarrow Attack\ Classification

]

This formulation treats cybersecurity detection as a

structured inference problem rather than merely a binary classification problem.

3.2 Cloud Security Graph Construction

Let the cloud security environment be represented by:

[

G_t=(V_t,E_t,X_t)

]

where (V_t) denotes security entities at time (t), (E_t) represents interactions, and (X_t) contains node and edge attributes.

Node categories can include users, virtual machines, containers, applications, databases, APIs, IP addresses, and authentication identities. Edge types can represent login relationships, network communication, file access, API invocation, process interaction, and service dependencies.

For example, if a user authenticates to a virtual machine, the authentication relationship becomes an edge. If that virtual machine subsequently accesses a database, another edge is created. An isolated authentication event may appear normal, whereas a sequence involving an unusual user, previously unseen machine, and abnormal database access can produce a suspicious graph structure.

This approach reflects the relational reasoning emphasized by Dong et al. (2019) and the state-transition perspective underlying graph search (Doran & Michie, 1966).

3.3 Temporal Graph Representation

Cloud attacks evolve over time; therefore, a static graph may be insufficient. The proposed model maintains temporal graph states:

[

G_1 \rightarrow G_2 \rightarrow ... \rightarrow G_T

]

Each graph represents the security state during a defined temporal interval. Changes between consecutive states can indicate behavioral escalation.

For example, a credential anomaly in (G_1), lateral movement in (G_2), and abnormal database access in (G_3) collectively provide stronger evidence than any individual event. The temporal representation therefore supports identification of attack trajectories rather than only attack snapshots.

3.4 Graph Deep Learning Layer

Each node initially possesses a feature vector (x_v). A graph-learning function transforms this information into an embedding:

[

h_v^{(l+1)}=\sigma \left(W^{(l)}h_v^{(l)}+

AGG_{u\in
N(v)}\left(\phi(h_u^{(l)},e_{uv})\right)\right)

]

where (N(v)) denotes neighboring nodes, (e_{uv}) represents edge information, (W) is a learnable parameter matrix, and (AGG) is an aggregation function.

The essential advantage is contextual representation. A node's embedding reflects not only its own attributes but also the behavior of connected entities. Consequently, a legitimate login from a trusted user and a similar login followed by abnormal resource access can obtain different representations because their surrounding graph structures differ.

The conceptual design is consistent with the broader development of neural methods for structured reasoning (Dong et al., 2019) and learned heuristic representations (Ferber et al., 2020).

3.5 Learned Heuristic Threat Prioritization

Large cloud graphs may contain millions of interactions. Processing every graph region with equal computational priority is inefficient. The proposed framework therefore introduces a learned threat-priority function:

[

H(G_i)=f_\theta(G_i)

]

where (H(G_i)) estimates the security relevance of graph state (G_i).

High-priority graph regions receive deeper analysis. This principle is theoretically grounded in heuristic search and learned heuristic functions (Bonet & Geffner, 2001; Arfaee et al., 2011). Rather than manually defining every suspicious pattern, the system learns which relational configurations are more strongly associated with attack states.

Pattern-database concepts can also inspire compressed security-state representations. Culberson and Schaeffer (1998) and Edelkamp (2001) demonstrate that selective abstractions can preserve useful information while reducing search complexity.

3.6 Attack Classification Mechanism

The final representation is passed to a classification layer:

[

P(y|G)=Softmax(W_ch_G+b_c)

]

where (h_G) is the graph-level representation and (y) represents an attack category.

The classification layer may distinguish between benign activity and multiple categories of malicious behavior, provided appropriate labeled training data are available. Importantly, the proposed framework does not require every decision to originate from the classifier. Heuristic prioritization can first identify suspicious graph regions, after which classification can operate on a reduced and more informative state.

3.7 Training Strategy

Training can combine supervised classification with representation learning. The model minimizes a classification objective while regularizing graph embeddings to avoid excessive sensitivity to individual noisy events. Neural heuristic research suggests that hyperparameter configuration can significantly affect the quality of learned search guidance (Ferber et al., 2020), while bootstrapping approaches can improve the development of neural heuristic functions (Ferber et al., 2022).

The architecture may also incorporate reinforcement-learning principles where threat states generate stronger reward signals for correct prioritization. Gehring et al. (2022) provides a theoretical basis for viewing heuristics as dense reward generators.

3.8 Operational Detection Process

During deployment, cloud telemetry is transformed into graph events. Newly observed relationships update the graph, after which node and edge representations are generated. The learned heuristic mechanism prioritizes suspicious regions. The deep-learning classifier then estimates attack likelihood and category. High-risk graph states can be forwarded to security analysts or automated response mechanisms.

The complete functional sequence is therefore:

[

Telemetry \rightarrow Graph Update \rightarrow Embedding \rightarrow

Heuristic Ranking \rightarrow Classification \rightarrow Alert

]

This structure is particularly appropriate for cloud environments because it supports continuous rather than purely batch-oriented analysis.

4. RESULTS

The analytical evaluation of the proposed architecture produces several significant findings. First, graph representation provides a stronger conceptual basis for cloud cyberattack detection than isolated-event modeling because security events are inherently relational. The distinction is particularly important for coordinated attacks in which individual events may

remain below conventional anomaly thresholds but become suspicious when connected through user, workload, network, and resource relationships. This finding is consistent with the graph-search perspective established by Doran and Michie (1966) and the relational-learning orientation of Dong et al. (2019).

Second, the architecture indicates that learned graph representations can improve contextual threat interpretation. Instead of assigning a threat score exclusively from an event's intrinsic attributes, the model incorporates information from neighboring entities. This creates the possibility of identifying attack chains, lateral movement patterns, and abnormal dependency structures. The approach consequently moves detection from event classification toward structural state classification.

Third, the learned heuristic mechanism provides a theoretical response to scalability. Large cloud graphs contain extensive benign activity, and exhaustive processing can impose substantial computational cost. Prior work on heuristic search and learned heuristics demonstrates that informed state prioritization can reduce unnecessary exploration (Bonet & Geffner, 2001; Arfaee et al., 2011). Applying this principle to cybersecurity suggests that graph regions with stronger evidence of malicious progression can be analyzed more intensively.

Fourth, the integration of deep learning and heuristic reasoning creates a complementary relationship rather than a replacement relationship. Neural models provide adaptive feature extraction, while heuristic mechanisms provide prioritization. Agostinelli et al. (2019) illustrates the potential of combining learned representations with search, while Ferber et al. (2022) demonstrates the value of neural heuristics for structured problem solving. The proposed cybersecurity architecture transfers these principles into cloud threat analysis.

Fifth, the framework provides a theoretically richer extension of the graph-based cloud threat-detection direction presented by Marri et al. (2025). Its principal contribution is the explicit integration of graph construction, temporal state modeling, learned heuristic prioritization, and deep classification within one analytical pipeline. However, the findings are architectural and theoretical rather than experimentally validated. No claim of superior accuracy, precision, recall, latency, or false-positive reduction can be made without a controlled dataset and benchmark comparison.

Finally, the analysis identifies several limitations. Graph construction may introduce noise when relationships are incomplete or incorrectly inferred. Dynamic cloud infrastructures can cause graph distributions to change rapidly, producing concept drift. Learned heuristics may also prioritize previously observed attack structures while underestimating novel attacks. Computational costs remain significant for high-frequency, high-volume cloud telemetry. These limitations indicate that practical validation requires large-scale datasets, temporal testing, adversarial evaluation, and resource-aware benchmarking.

5. DISCUSSION

The principal theoretical implication of this study is that cyberattack detection can be formulated as a structured graph inference problem rather than solely as conventional feature-based classification. This positioning is important because cloud attacks frequently involve interactions among multiple entities. The proposed architecture therefore connects the state-transition foundations of graph search with the adaptive representation capabilities of deep learning.

The framework also demonstrates the value of heuristic reasoning in deep cybersecurity systems. Classical planning research emphasizes that intelligent search requires informative estimates of state relevance (Bonet & Geffner, 2001). Learned heuristic research extends this principle by allowing models to derive useful guidance from experience (Arfaee et al., 2011; Garrett et al., 2016). In cloud cybersecurity, this translates into prioritizing graph regions that exhibit stronger evidence of attack progression. Such prioritization can potentially reduce unnecessary computation and improve analyst attention allocation.

The approach is additionally consistent with neural reasoning research. Dong et al. (2019) demonstrates the potential of neural systems to process relational

structures, while Agostinelli et al. (2019) illustrates how learning and search can operate together. The proposed model applies the same broad principle to cybersecurity: learning identifies meaningful representations, whereas structural reasoning determines how those representations should be interpreted within an interconnected environment.

Compared with purely static graph classification, the proposed temporal formulation offers a stronger mechanism for recognizing multi-stage attacks. An attacker may initially generate behavior that resembles legitimate activity, but subsequent interactions can alter the structural meaning of the original event. Temporal graph states therefore provide a mechanism for reconstructing behavioral progression. This perspective complements the graph-based cyber-threat model presented by Marri et al. (2025), while emphasizing state evolution and learned prioritization.

Nevertheless, several trade-offs must be considered. Increasing graph complexity can improve contextual representation but simultaneously increases memory and computational requirements. More expressive models can capture sophisticated relationships but may become harder to interpret. Learned heuristics can reduce search effort but may inherit biases from training distributions. Neural planning research also indicates that hyperparameter choices influence model behavior (Ferber et al., 2020), implying that cybersecurity implementations require systematic optimization rather than arbitrary architectural configuration.

Another important limitation concerns data quality. A graph model cannot recover relationships that are absent from telemetry. Missing authentication records, encrypted traffic, delayed logs, or inaccurate identity mappings can distort graph structure. Similarly, adversaries may deliberately manipulate graph relationships to evade detection. Consequently, robustness should be treated as a central research requirement rather than an optional enhancement.

From a practical perspective, the proposed architecture is most valuable when integrated into cloud security monitoring systems capable of maintaining continuously updated graph representations. The model can support security analysts by ranking suspicious relational structures instead of producing an uncontextualized stream of alerts. However, empirical validation remains essential. Future studies should compare the architecture against conventional machine-learning and graph-learning baselines using standardized cloud-security datasets and should report detection effectiveness, computational overhead, false-positive behavior, and resilience to previously unseen attack patterns.

6. CONCLUSION

This paper presented an advanced graph-based deep learning approach for cyberattack detection in cloud computing networks. The framework models cloud infrastructure as a dynamic heterogeneous graph and combines graph representation learning with learned heuristic threat prioritization and attack classification. Its theoretical foundation draws upon graph traversal, heuristic search, pattern databases, learned heuristics, neural reasoning, and deep learning research.

The principal contribution is the integration of these concepts into a unified cybersecurity architecture in which attacks are analyzed as evolving relational states rather than isolated events. The approach provides a stronger conceptual basis for identifying coordinated and multi-stage attacks and offers a mechanism for reducing computational effort through selective graph-state prioritization.

The framework extends the graph-based cybersecurity direction represented by Marri et al. (2025) by emphasizing temporal state modeling and learned heuristic reasoning. Nevertheless, the study is conceptual and does not substitute for empirical validation. Future research should implement the architecture using large-scale cloud telemetry, evaluate it against established detection baselines, investigate adversarial robustness, optimize computational efficiency, and examine explainability for security-analyst decision making.

REFERENCES

1. Agostinelli, F., McAleer, S., Shmakov, A., & Baldi, P. (2019). Solving the Rubik's cube withdeep reinforcement learning and search.Nature

Machine Intelligence,1(8), 356–363.

2. Alcazar, V., Borrajo, D., Fernandez, S., & Fuentetaja, R. (2013). Revisiting regression inplanning.International Joint Conference on Artificial Intelligence, 2254–2260.

3. Arfaee, S. J., Zilles, S., & Holte, R. C. (2011). Learning heuristic functions for large statespaces.Artificial Intelligence,175(16-17), 2075–2098.

4. Bäckström, C., & Nebel, B. (1995). Complexity results for SAS+planning.ComputationalIntelligence,11(4), 625–655.

5. Bonet, B. (2013). An Admissible Heuristic for SAS+Planning Obtained from the StateEquation.International Joint Conference on Artificial Intelligence, 2268–2274.

6. Bonet, B., & Geffner, H. (2001). Planning as heuristic search.Artificial Intelligence,129(1-2),5–33.

7. Culberson, J. C., & Schaeffer, J. (1998). Pattern databases.Computational Intelligence,14(3), 318–334.

8. Dong, H., Mao, J., Lin, T., Wang, C., Li, L., & Zhou, D. (2019). Neural logic machines[Poster].International Conference on Learning Representations.

9. Doran, J. E., & Michie, D. (1966). Experiments with the Graph Traverser Program.Proceed-ings of the Royal Society A,294, 235–259.

10. Edelkamp, S. (2001). Planning with pattern databases.European Conference on Planning,13–24.

11. Ferber, P., Geißer, F., Trevizan, F., Helmert, M., & Hoffmann, J. (2022). Neural networkheuristic functions for classical planning: Bootstrapping and comparison to othermethods.International Conference on Automated Planning and Scheduling, 583–587.

12. Ferber, P., Helmert, M., & Hoffmann, J. (2020). Neural network heuristics for classicalplanning: A study of hyperparameter space.European Conference on ArtificialIntelligence,325, 2346–2353.

13. Frances, G., Ramírez Jávega, M., Lipovetzky, N., & Geffner, H. (2017). Purely declarativeaction descriptions are overrated: Classical planning with simulators.InternationalJoint Conference on Artificial Intelligence, 4294–4301.

14. Francés, G., Ramirez, M., & Collaborators. (2018). Tarski: An AI planning modeling frame-work [GitHub: https://github.com/aig-upf/tarskio].

15. Garrett, C. R., Kaelbling, L. P., & Lozano-Pérez, T. (2016). Learning to rank for synthesizingplanning heuristics.International Joint Conferences on Artificial Intelligence, 3089–3095.

16. Gehring, C., Asai, M., Chitnis, R., Silver, T., Kaelbling, L. P., Sohrabi, S., & Katz, M. (2022).Reinforcement learning for classical planning: Viewing heuristics as dense rewardgenerators.International Conference on Automated Planning and Scheduling,32,588–596.

17. M. R. Marri, S. B. Kurada, S. Gupta, S. Dey, S. Kumar and R. Chauhan, "Graph-Based Deep Learning Model for Identifying Cyber Threats in Cloud Platforms," 2025 International Conference on Computational Intelligence, Security, and Artificial Intelligence (IntelliSecAI), Al-Khobar, Saudi Arabia, 2025, pp. 1-7, doi: 10.1109/IntelliSecAI66368.2025.11472831.