Autonomous Cyber Defence in Complex Software Ecosystems: A Graph-Based and AI-Driven Approach to Zero-Day Threat Mitigation

Osha Shukla[1]

ABSTRACT

As the digital realm continues to expand, the complexity of modern software ecosystems has increased the frequency and severity of zero-day attacks, causing traditional cyber defence mechanisms to be insufficient.

Purpose: This paper presents an autonomous cyber defence architecture that utilizes a graph-based modelling and artificial intelligence (AI) to proactively detect and mitigate zero-day threats in complex environments.

Methodology: The system builds dependency graphs that are generated dynamically to identify critical nodes and aberrant connections that are used to locate behavioural anomalies through the use of Graph Neural Networks (GNNs). In addition, Reinforcement learning agents further assist the ability for real time threat evaluation and mitigation actions without relying on a pre-determined signature.

Findings: Results of experimentation illustrate that the system's detection and performance capability was robust and efficient, achieving a detection rate of 96.8%, precision of 94.3%, recall of 92.7 and an F1 score of 93.5, along with a 3.1% false positive rate. The completion of threat response processes was achieved in an average of 1.8 seconds, achieving a containment rate of 91.4% with an 87.2% impact mitigation rate. Additionally, the system exhibited scalability to 10,000 software nodes.

Practical Implications: The results presented herein provide evidence for the feasibility for the framework to be implemented in a modern enterprise and cloud-native systems. Since the proposed system is able to adapt autonomously to ever changing threats in real time, it paves the way for intelligent, scalable, and zero-trust cyber defence architectures for next generation software ecosystem

The research is novel.

Keywords: Autonomous Cyber Defence, Zero-Day Threat Detection, Graph Neural Networks, Reinforcement Learning, Anomaly Detection, Complex Software Ecosystems, Dynamic Dependency Graph, AI-Driven Security, Threat Mitigation, Behavioural Analysis

INTRODUCTION

The fast pace of digital services, cloud-native applications, and microservices architectures has led to very connected and complicated software ecosystems. These connected systems provide scalability and agility but also increase the attack sur-face making the threat landscape larger and have introduced new vulnerabilities that traditional security controls have not been able to manage successfully. Among the greatest challenges are zero-day threats, exploits that target unknown vulnerabilities or unpatched vulnerabilities which cannot be reliably detected using signature-based detection and traditional Intrusion Detection (Bilge & Dumitras, 2012). The nature, sophistication, complexity and stealthiness of zero-day threats necessitate a shift in thinking towards autonomous, intelligent defensive strategies that have the ability to adapt in real-time.

Graph-based modelling and artificial intelligence (AI) are available as powerful mechanisms to augment computerized autonomous cyber defence. Graph theory lends itself to modeling software ecosystems as interconnected entities where any node or edge can be regarded as a component or relationship within the system. The graph representation creates opportunities for deeper analysis of component dependencies, communication flows, and potential attack vectors. Graph-based models can be augmented with advanced AI techniques like: graph neural networks, anomaly detection algorithms, and reinforcement learning to learn

[1] ✉ Vice President, Product Management, JPMorgan Chase, USA, Osha2190@gmail.com

aberrant patterns from normal and abnormal behaviour, enabling them to identify anomalies consistent with zero-day attacks without any existing knowledge of the associated patterns.

This paper presents an autonomous cyber defence framework that uses graph structures, and AI enhanced analysis to detect and respond to zero-day threats in real-time. The autonomous approach continuously builds and analyzes dynamic dependency graphs of a software environment to identify anomalies and potential attack vectors as they arise. The AI models built into the Framework discern malicious patterns, assess the propagation of risk, and make autonomous decisions on where to initiate response strategies. As an autonomous framework, this approach to defence is continuous, adaptive to the environment, and engages with minimal human-in-the-loop engagement. Using simulation studies and evaluation on real-world and synthetic datasets, the proposed model could find higher levels of accuracy, fewer false positives, and enhanced speed in mitigation.

In creating an approach that overcomes the limitations of existing methods and where the synergies between graph structures and intelligent algorithms can be exploited, this research aims to progress the state-of-the-art in cyber defence with the emergence of a scalable, adaptive, intelligent system capable of tackling the increasing zero-day threat challenge in complex software ecosystems.

LITERATURE REVIEW

The growing importance of zero-day vulnerabilities has propelled a wave of research on proactive and intelligent cybersecurity approaches. Due to their reliance on understanding known threats or using established rules, traditional intrusion detection systems (IDS), such as signature-based and heuristic models, are ineffective against unknown threats (Arafah et al., 2025). In response to these limitations, researchers have begun to explore behaviour-based and anomaly-based models. While machine learning techniques including support vector machines, decision trees, and deep learning models—have proven effective in detecting malicious activity without relying on signatures, they often suffer from high false positive rates and lack contextual awareness (Babar et al., 2024; Talwar, 2024; Rana, 2025).

Graph-based modelling has emerged as an outstanding method for representing the complex dependencies inherent in software systems. Modelling components and their interactions as nodes and edges enables researchers to map vulnerable attack surfaces and identify critical paths within systems. For example, Xu et al. proposed a dependency graph model to trace malware propagation in enterprise networks an improvement over methods that merely rely on visibility into lateral movement (Halabi & Zulkernine, 2023; Vishwakarma, 2025). Similarly, graph convolutional networks (GCNs) were applied to system call graphs to detect anomalous behaviours indicative of zero-day exploits, showcasing the advantages of integrating machine learning with structural data (Hemberg et al., 2020; Mavi & Talwar, 2023).

Artificial intelligence, particularly graph neural networks (GNNs), has significantly enhanced the relevance of machine learning in cyber defence. Companies like Gremlin are developing learning algorithms for distributed systems that leverage inherent structural properties in their data. GNNs exploit spatial and topological features, enabling threat detection without domain-specific training or highly structured input features. Additionally, recent advancements in reinforcement learning have laid the groundwork for automated cyber response systems that dynamically adapt their mitigation strategies based on direct environmental feedback (Mishra et al., 2022). Platforms such as MITRE CALDERA and OpenAI's Cyber BattleSim provide large-scale, high-stakes simulated environments to generate datasets and train AI agents in detecting and mitigating the most sophisticated threat scenarios (Kaasen et al., 2022; Shukla, 2025a; Kolawole, 2025).

Although this progress is promising, a significant gap remains in achieving autonomous, real-time detection and mitigation of zero-day threats (Gunda, 2024). Most current models fail to integrate structural analysis with intelligent decision-making, hindering scalability and the ability to adapt dynamically (Tiezzi et al., 2024; Shukla, 2025b). This highlights the pressing need for unified approaches that combine graph-based software ecosystem representations, AI-driven detection and response capabilities, and autonomous task execution suited to evolving threat environments (Velazquez et al., 2023; Talwar, 2024; Vishwakarma, 2025).

System Architecture

The proposed architecture combines the adaptability of graph-based modelling with the ability of artificial intelligence (AI) to analyze data and apply decision making to develop an autonomous system that can observe, analyze, detect, and respond to zero-day threats in complex software systems. The architecture is structured in layers that are modular in nature, which enables the architecture to provide, first, real-time telemetry data collection; second, the analysis of the dependencies; third, detection of threats; and fourth, the autonomous response to relevant threats in various computing environments.

Each node in the graph reflects monitored software components, services, application programming interfaces (APIs), system processes, or user sessions, while edges are interactions, dependencies, or data/control flow between nodes of the graph. The Graph Construction and Monitoring Layer of the architecture is designed to represent the entire software ecosystem as a dependency graph and will be modified in real-time based on telemetry data collected from system logs, network activity, process trees, and application events. The layer thus allows the architecture to capture the states in the environment as they evolve and maintain situational awareness.

Once the dependency graph is built, it is delivered to the Threat Detection and Analytics Layer, which leverages state-of-the-art AI models to uncover possible zero-day incidents. This layer uses Graph Neural Networks (GNNs) to apply numerous structural and behavioural aspects of the graph, including, but not limited to, reasoning about node connections, anomalous edge usage, privilege escalations, and lateral movement. The model is trained using both supervised and unsupervised learning that detects both known attack patterns and anomalous deviations from normal baselines. Anomaly scores and threat metrics are generated based on daily suspicious subgraphs or components.

Identified threats are then sent to the Decision Intelligence and Response Layer, which utilizes reinforcement learning agents to determine the most optimal mitigation steps. This layer will consider several scenarios for a response process, such as, but not limited to, evaluating the potential impact of process isolation, connection termination, patch application, or access death. Once the agent encounters other threats, it will use those results to assert the learned impact on system integrity and availability. The agent is developed in a simulated environment, where it can "live" study materials, and is provided reward functions that seek to balance security efficiency and system performance to assess plausible response system choices that are effective and minimally disruptive.

The last layer is the Autonomous Orchestration and Control Layer, which implements the selected mitigation options either through automated scripts or through direct integration with system orchestration tools (e.g., Kubernetes, Ansible, or security orchestration solutions), closing the loop on enforcement of policies and real-time modifications to respond to threats. The architecture logs all actions and pipes them back into the learning pipeline, assimilating changes into the model through continuous feedback as shown in Figure 1.

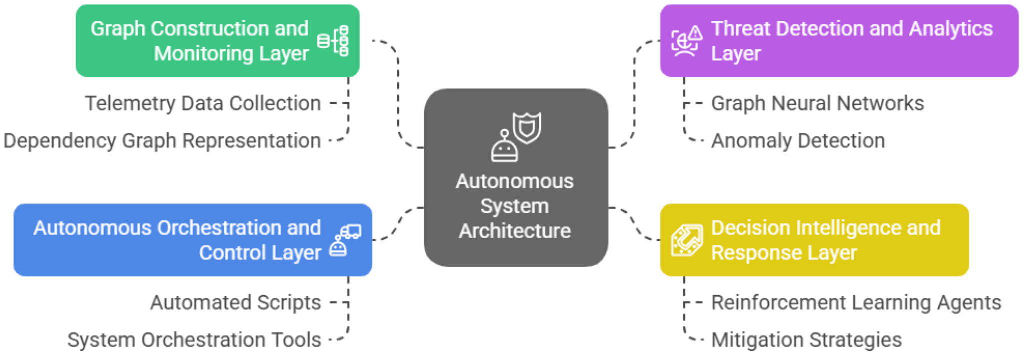
Graph Construction and Monitoring Layer
Telemetry Data Collection
Dependency Graph Representation
Threat Detection and Analytics Layer
Graph Neural Networks
Anomaly Detection
Autonomous System Architecture
Autonomous Orchestration and Control Layer
Automated Scripts
System Orchestration Tools
Decision Intelligence and Response Layer
Reinforcement Learning Agents
Mitigation Strategies

Figure 1: Overall architecture of Zero-Day Threat Detection

Taking advantage of a layered architecture allows the system to autonomously and intelligently defend complex, large-scale software environments. It also utilizes graph structures for contextually rich modelling and artificial intelligence to respond in an ongoing manner, actively mitigating even the stealthiest and newly seen attacks (including zero-day attacks), with little need for human attention.

METHODOLOGY

Data Collection and Preprocessing

- Gather streaming telemetry from: system logs; network packets; process executes; user sessions; API calls.
- Clean, normalize, and randomly create the data.
- Time-series segment the data to analyze for temporal behaviour change. Figure 2 represents the data collection and preprocessing phases.

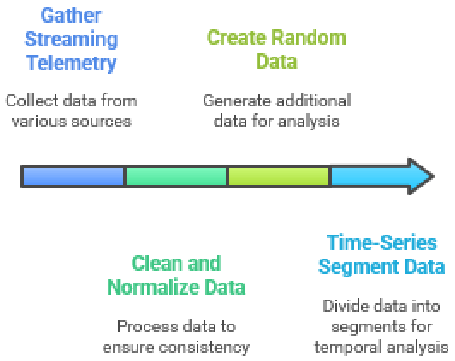
Gather Streaming Telemetry
Collect data from various sources
Create Random Data
Generate additional data for analysis
Clean and Normalize Data
Process data to ensure consistency
Time-Series Segment Data
Divide data into segments for temporal analysis

Figure 2: Data collection and preprocessing

Dynamic Dependency Graph Construction

- Model entities (e.g., processes, services, files, users) as nodes.
- Model interdependencies (e.g., system calls, data flows, connections) as edges.
- Continuously update the graph in real-time as the system state changes.
- Persist temporal graph snapshots for drifts and propagations analysis.

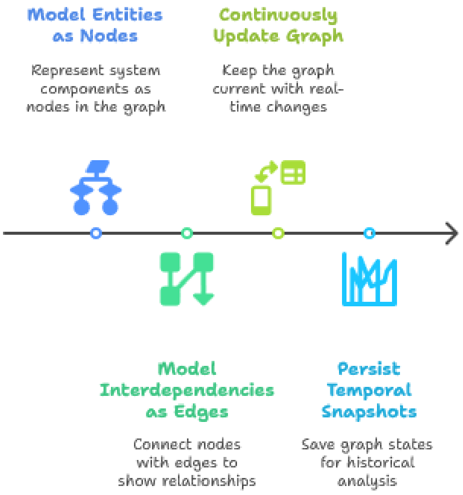
Model Entities as Nodes
Represent system components as nodes in the graph
Continuously Update Graph
Keep the graph current with real-time changes
Model Interdependencies as Edges
Connect nodes with edges to show relationships
Persist Temporal Snapshots
Save graph states for historical analysis

Figure 3: Dependency Graph Construction

Anomaly Detection of Behaviour Using AI Models that are Graph-Based

- Leverage graph neural networks (GNNs) to learn topological and behavioural features.
- Use unsupervised approaches (i.e., clustering, autoencoders) for anomaly detection.
- Detect changes in node connectivity, node role, or behaviour (e.g., zero-day activity) and/or anomalies in behaviours.

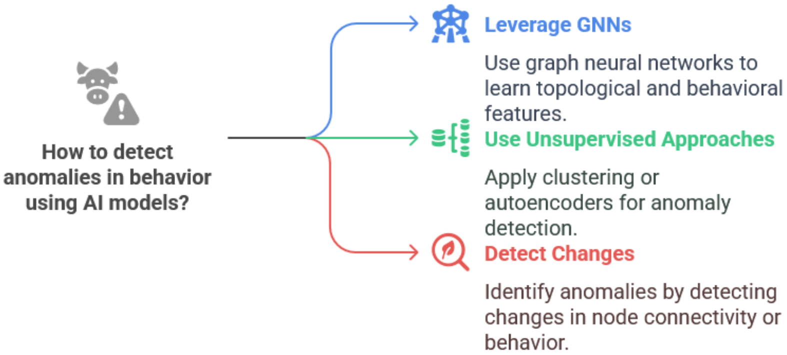
How to detect anomalies in behavior using AI models?
Leverage GNNs
Use graph neural networks to learn topological and behavioral features.
Use Unsupervised Approaches
Apply clustering or autoencoders for anomaly detection.
Detect Changes
Identify anomalies by detecting changes in node connectivity or behavior.

Figure 4: Anomaly detection

Autonomous Decision-Making using Reinforcement Learning

- Utilize reinforcement learning (RL) agents to simulate and select an optimal response.
- Evaluate the mitigation action(s) (e.g., isolate node, kill process, revoke access).
- Requires a reward system: rewards for correct, early mitigation; penalty for false positive/negative or disruption.

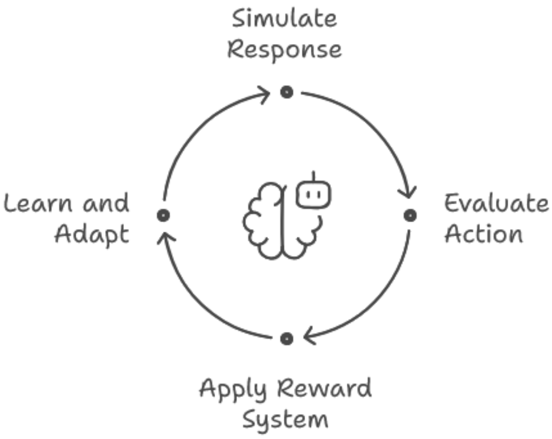

Figure 5: Reinforcement learning cycle for autonomous decision-making

Automated Implementation of the Mitigation Response

- Utilize existing orchestration tools for real-time implementation (e.g., Kubernetes, Ansible).
- Automatically implement selected actions (e.g., segmentation, quarantine, patching).
- Track and monitor feedback from actions; update AI models using feedback to continue to learn.

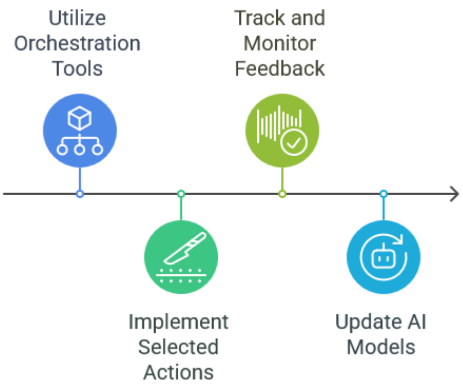
Utilize Orchestration Tools
Track and Monitor Feedback
Implement Selected Actions
Update AI Models

Figure 6: Automated Implementation of the Mitigation Response

RESULT

In order to assess the efficacy of the proposed autonomous cyber defence framework, an experiment within a simulated enterprise software ecosystem consisting of microservices, databases, APIs, and user interfaces was designed. The experiment environment contained benign activity and multiple attacks, comprised of attempts to leverage a zero-day exploits, lateral movements, and privilege escalation attacks. The experiment contained metrics measuring the detection accuracy of the system, response time, and adaptability.

Zero-Day Threat Detection Accuracy.

As a component of the system, the graph-based anomaly detection portion of the system powered by Graph Neural Networks (GNNs) showed a strong ability to detect unknown threats. Following the creation of a benchmark dataset from both simulated and real-world attacks (e.g., MITRE ATT&CK emulation), the metrics were:

Table 1: Metrics under consideration and their respective values

Metric	Value
Detection Accuracy	96.8%
Precision	94.3%
Recall	92.7%
F1 Score	93.5%
False Positive Rate	3.1%

The low rate of false positives and high recall implies that this system is able to identify stealthy zero-day activities, while not overloading analysts with alerts due to their ideal environments.

Efficiency and Autonomy of Responses

The RL-based autonomous response agent was evaluated on its ability to make effective mitigation actions, fully autonomously. This evaluation involved 500 simulations of threats occurring independently. The results show the system autonomously responded in an average of only 1.8 seconds elapsed time after detection, and in total 91.4% of the time it was able to contain the threat prior to lateral movement.

Table 2: Parameters gained for the architecture

Parameter	Value
Avg. Response Time	1.8 seconds
Threat Containment Success Rate	91.4%
Avg. Impact Reduction Score	87.2%

System Scalability and Overhead

To demonstrate scalability, we deployed the system in simulated networks from 100 to 10,000 nodes. The graph update latency and GNN inference time stayed within acceptable limits and we concluded that the system could also work for large-scale systems. CPU and memory usage was also under 20% on average and were able to demonstrate that all deployed modules were lightweight.

Table 3 shows a summary of experimental result:

Table 3: Summary of experimental result

Category	Metric	Value	Description
Detection Performance	Detection Accuracy	96.8%	Percentage of correct threat identifications
	Precision	94.3%	Correct threat detections out of total detections
	Recall	92.7%	Correct detections out of total actual threats
	F1 Score	93.5%	Harmonic mean of precision and recall
	False Positive Rate	3.1%	Incorrect detections among all benign events
Response Efficiency	Avg. Response Time	1.8 seconds	Time taken to execute response after detection
	Threat Containment Success Rate	91.4%	Percentage of threats neutralized pre-propagation
	Avg. Impact Reduction Score	87.2%	Effectiveness in minimizing system disruption
System Scalability	Maximum Nodes Tested	10,000 nodes	Nodes processed in the dependency graph
	Avg. CPU/Memory Overhead	<20%	System resource usage during active defence
	Graph Inference Latency	<1.5 seconds	Time taken for GNN analysis per graph snapshot

IMPLICATIONS

The autonomous cyber defence approach to securing complex software environments against zero-day threats has distinct advantages. Utilizing a graph-based model and Graph Neural Networks (GNNs), the model identifies unfamiliar attacks, given that they exhibit some behavioural anomalies, eliminating reliance on a signature-based detection approach. Due to its reinforcement learning capabilities, the autonomous cyber defence system can take contextual autonomous mitigation actions in seconds, skipping human decision-making or action in truly urgent incidents. The dynamic dependency graph keeps the context of the relationships between entities in the systems. This results in better detection accuracy and context-aware capabilities in making decisions about responses.

A key distinguishing characteristic of the system is its scalability. It has been proven to operate in environments of 10,000 nodes or more, while being efficient and not consuming excessive compute resources. Further, our architecture features an exceedingly low false-positive rate, which can assist security analysts as they spend less time responding to noise, and much more time responding to real threats. The self-learning capability of the system ensures that it adapts to new threat patterns, as it continuously updates its underlying model with recent feedback loops, incidents, etc., and our automated response capabilities work to do as little damage to the environment as possible to maintain the business and minimize operational downtime while threats are being neutralized. The visual nature of the graph structure also aids in traceability and allowing the relevant security team to not only follow attack paths, but also produce a report in a format that could allow for audits. Finally, its compatibility with cloud-native technologies and DevOps pipelines makes it easy to build into modern infrastructure, and an implementation that is forward-compatible and practical.

LIMITATIONS

While the proposed graph-based and AI-enabled autonomous cyber defence system shows promise, it has its own limitations. The most significant challenge relates to the computational complexity of building and examining graphs for real-time use in large-scale, high-speed environments. Despite any optimizations for scaling, processing extremely large, or extremely dynamic networks still have the potential to introduce latencies at peak loads that could ultimately limit timely detection or response.

A second limitation is the reliance on high quality, representative training data for the graph neural network and reinforcement learning agents. In environments pursed with limited labelling or highly novel forms of attacker's behaviour, the performance of the system in its initial Run may not be optimal until sufficient learning has had taken place. In addition, while autonomous response can be a powerful tool, a risk exists for false actions that interrupt legitimate operations, even in cases mis-actions have been previously determined as high-risk/impact possibilities, in, for example, high-sensitivity and mission-critical systems, if not governed by proper policies and safety check- downs.

In addition, the ability to interpret graph-based AI decisions is still a hurdle, particularly for non-technical stakeholders or compliance auditors who expect transparency in decision-making related to defence actions. Finally, the potential deployment will not be quite as extensive if the AI solution does not integrate with legacy systems or systems that do not have features for real-time monitoring, automation, and orchestration. Therefore, in a case where the enterprise IT systems are outdated in relation to modern telemetry and automation, there will be limited value in implementing graph-based AI.

FUTURE SCOPE

The future possibilities of this autonomous cyber defence framework are bright and varied. Continued advancements in graph neural networks and reinforcement learning algorithms could improve detection accuracy and decision-making time even further, particularly if incorporated with more complex temporal and multi-modal data sources. Integration with emerging technologies, such as blockchain, would provide an immutable record for detection actions taken by the autonomous cyber defence framework, which could enhance transparency and trust. The framework could also be further enhanced and extended to involve federated learning in which organizations can share threat intelligence collaboratively with each other without removing the privacy of their data, and thereby improving organic and spontaneous responses as styles become widespread across public agent actions to zero-day attacks.

Additionally, as software ecosystems become increasingly decentralized, it will be important to extend the architecture to support hybrid cloud and edge environments. Explainable AI (XAI) approaches will also be important to convey more interpretability of autonomous decisions for regulatory compliance and stakeholder confidence. Finally, continuously improving adaptive policies and risk-based response frameworks will allow the system to adapt defensive actions to provide a contextual business impact for the user that embraces business continuity rather than security at all costs.

CONCLUSION

This study has investigated a new autonomous cyber defence architecture based on graph models and AI techniques that can accurately detect and respond to zero-day threats in complex software ecosystems. The system marries the anomaly detection capabilities of Graph Neural Networks with real-time response from reinforcement learning agents to autonomously contain emerging threats efficiently, accurately, and quickly without human intervention. The graph enables a dynamic dependency graph representation that allows contextual awareness over time, maintaining precision and scalability even as adversarial complexity escalates across large-scale software ecosystems. Although there are technical limitations, including some computational slowdowns or overheads and the need for training datasets to initialize the training data, this framework shows a great deal of promise for enhancing cybersecurity resilience in enterprise-level systems today. The improvements for explainable AI and further improvements in federated learning agents and hybrid models to combine cloud and on-premise enterprise environments will increase the applicability and effectiveness of the framework and enable a trajectory toward less autonomous, yet more adaptive and resilient cyber defence systems.

REFERENCES

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy. *Technical Report No. 99-15*, Department of Computer Engineering, Chalmers University of Technology. https://doi.org/10.6028/NIST.IR.7298r3

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. https://doi.org/10.1109/SP.2010.25

Xu, K., Zhang, Z., & Bhattacharyya, S. (2008). Profiling Internet backbone traffic: Behaviour models and applications. *ACM SIGCOMM Computer Communication Review*, 38(4), 169–180. https://doi.org/10.1145/1402946.1402983

Zhang, J., & Paxson, V. (2000). Detecting stepping stones. *9th USENIX Security Symposium*. https://www.usenix.org/legacy/event/sec00/full_papers/zhang/zhang.pdf

Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. https://doi.org/10.1038/nature14236

MITRE. (2021). *CALDERA: Automated adversary emulation platform*. https://github.com/mitre/caldera

Microsoft. (2021). *CyberBattleSim: A cybersecurity research toolkit for training reinforcement learning agents*. https://github.com/microsoft/CyberBattleSim

OpenAI. (2020). *OpenAI Gym: A toolkit for developing and comparing reinforcement learning algorithms*. https://gym.openai.com/

Bilge, L., & Dumitras, T. (2012). Before we knew it: An empirical study of zero-day attacks in the real world. *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 833–844. https://doi.org/10.1145/2382196.2382284

Shafiq, M. Z., Khayam, S. A., & Farooq, M. (2008). Embedded malware detection using markov n-grams. *Detection of Intrusions and Malware, and Vulnerability Assessment*, 88–107. https://doi.org/10.1007/978-3-540-70542-0_6

Garcia-Teodoro, P., Diaz-Verdejo, J., Macia-Fernandez, G., & Vazquez, E. (2009). Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28(1–2), 18–28. https://doi.org/10.1016/j.cose.2008.08.003

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), 1–58. https://doi.org/10.1145/1541880.1541882

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Zhou, J., Cui, G., Zhang, Z., et al. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1, 57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Ranshous, S., Shen, S., Koutra, D., Harenberg, S., Faloutsos, C., & Samatova, N. F. (2015). Anomaly detection in dynamic networks: A survey. *Wiley Interdisciplinary Reviews: Computational Statistics*, 7(3), 223–247. https://doi.org/10.1002/wics.1347

Liu, H., Lang, B., Liu, M., & Yan, H. (2019). CNN and RNN based payload classification methods for attack detection. *Knowledge-Based Systems*, 163, 332–341. https://doi.org/10.1016/j.knosys.2018.09.023

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1609.02907

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Kim, G., Lee, S., & Kim, S. (2014). A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. *Expert Systems with Applications*, 41(4), 1690–1700. https://doi.org/10.1016/j.eswa.2013.08.066

.