PENINSULA PUBLISHING LLC

SHIFRA
Vol. **(2025)**, 2025, **pp**. 203–212
ISSN: 3078-3186

Research Article

Hypergraph-Based Time-Aware Modeling for Multi-Phase Cyber Threat Simulation and Anomaly Detection

Ahmed Hussein Ali[1,*], Nadia Mahmood Hussien[2], Saba Abdulbaqi Salman[1], Yasmin Makki Mohialden[2], Kapil Joshi[3]

[1] *Department of Computer Science, College of Education, Al-Iraqia University, Baghdad-Iraq*

[2] *Computer Science Department, College of Science, Mustansiriyah University, Baghdad-Iraq*

[3] *Computer Science and Engineering Department, Uttaranchal University, Dehradun, India*

ARTICLEINFO

Article History

Received 1 Jul 2025

Revised: 24 Aug 2025
Accepted 23 Sep 2025
Published 7 Oct 2025

Keywords

Cybersecurity Simulation,

Software engineering,

Time-Aware Hypergraph,

Anomaly Detection,

Adversarial Behavior Modeling,

Multi-Stage Attack,

Transition Probability,

Cyber Kill Chain,

IDS Training,

DAG Baseline,

Synthetic.

ABSTRACT

This research uses a time-aware hypergraph design and anomaly-detection methods to simulate cyberattack sequences. Multi-phase cyber-attacks have high-order interactions, concurrency, and unpredictable timing that traditional directed graph models cannot describe. To overcome this constraint, we create hypergraphs with cyber kill-chain stages including reconnaissance, initial access, execution, persistence, privilege escalation, and exfiltration as nodes and hyperedges for logical and temporal groups.

Randomized temporal information is added to each simulated phase to create a two-hour enemy timeline. Repetitive simulations introduce regulated phase length and sequence ordering diversity. Feature-engineering pipelines include temporal alignment, stage ordering, and label encoding. Scatter-based anomaly plots and frequency-distribution histograms show aberrant phase lengths from an Isolation Forest model. A transition-probability matrix quantifies hypergraph stage-to-stage movement and better models adversarial behavior.

Structured datasets from simulations are provided for IDS training, red-team emulation, and threat-analysis investigations. The proposed time-aware hypergraph better reconstructs Advanced Persistent Threat (APT) behavior by capturing overlapping phases, higher-order relationships, and temporal uncertainty than DAG-based representations. The research compares the proposed hypergraph model to a baseline DAG model using the same synthetic dataset to demonstrate demonstrable changes in anomaly-detection performance and IDS-dataset quality to assure scientific validity. The findings section provides measurements and validation methodologies for the 22% anomaly-detection accuracy and 30% IDS-dataset quality increase over the DAG baseline.

1. INTRODUCTION

Multi-phase cyber-attacks are becoming more complicated to model due to adversaries' complexity and attack phases' overlap. Research shows that over 78% of advanced persistent threats (APTs) include concurrent or asynchronous attack phases, making simulation and detection difficult [1–3]. Modern opponents are dynamic, non-linear, and temporal, therefore static, deterministic modeling methods no longer work.

Hypergraphs are a significant alternative to directed graph models. Hypergraphs may describe higher-order dependencies where a single hyperedge can link numerous phases or components. Hypergraphs provide a better representation of overlapping attack phases when paired with time-aware methods, making them ideal for cyberattack simulation, IDS development, and AI-driven defensive modeling.

Despite advances in cyberattack modeling, most frameworks don't prioritize time. Meta Attack [4] and MITRE ATT&CK emulation profiles [5] offer organized adversarial logic but cannot capture temporal variability, limiting its realism for simulating real-world assaults [6]. Many previous approaches ignore complex persistent threats' stochastic and branching properties by adopting deterministic or strictly linear paths.

*Corresponding author email: Ahmed.ali@aliraqia.edu.iq
DOI: https://doi.org/10.70470/SHIFRA/2025/012

Little is known about integrating hypergraph-based models with time-sequenced data. To our knowledge, no cyberattack advancement approach handles logical interdependencies and temporal unpredictability [8–11]. This gap affects red-team simulations, automated IDS training, and AI-powered defensive system operations.

Realistic, reusable adversarial datasets for simulation, machine learning, and assessment are lacking. With old datasets like CICIDS and UNSW-NB15 with no sequenced adversarial logic, AI-driven threat detection models fail [12–15]. Thus, a flexible simulation framework that generates time-aware, hypergraph-based synthetic datasets that closely resemble real-world adversarial behaviors is needed. In this work, embedding temporal information, capturing overlapping stages using hyperedges, and incorporating anomaly-detection capabilities to identify unusual or suspicious temporal patterns in adversarial sequences enhance multi-phase attack representation.

This study creates a time-aware hypergraph-based simulation framework that assigns temporal information to each assault phase and expresses structural and temporal connections using hypergraph diagrams and Gantt-style visualizations to simulate multi-phase cyberattacks more accurately. The system creates realistic, reusable synthetic datasets that represent adversarial behavior's temporal and structural properties for red-teaming and IDS development. Phase length, hyperedge density, and assault span may be quantified.

This study discusses how time-aware hypergraphs can represent multi-stage, temporally distributed cyberattacks, how temporal visualizations and structural modeling can improve insights, and how the proposed framework compares to traditional attack graph models in accuracy, scalability, and modeling fidelity

2. RELATED WORK

The complexity of cyber threats and the need for improved security techniques have advanced cyberattack simulation and threat modeling during the past decade. Dynamic, concurrent, and multi-phase cyberattacks were not reflected by traditional deterministic and static models. M. Yue et al. (2019) [23] developed linear assault phases and rule-based logic simulation frameworks that gave basic insights into adversarial behavior but oversimplified multi-stage attack dynamics and refused to simulate phase interdependencies. Earlier modeling methodologies assumed predictable assault paths from a single breach site, Priyanga et al. (2020) [24] stressed the necessity for models that reflect non-linear and probabilistic transitions across highly adaptable threat behaviors.

Schneider et al. (2021) [25] Graph-based threat modeling for attack path, network exposure, and vulnerability exploitation chain visualization broadened the area. Although directed graphs improved structural awareness, their pairwise-only links prevented them from simulating overlapping or multi-actor incursions due to their inability to describe higher-order dependencies across contemporaneous assault pathways.

Takiddin et al. (2022) [26] employed anomaly-based phase modeling to identify abnormal system activity and improve threat detection, but it couldn't handle overlapping attack phases or concurrent adversarial operations, restricting its applicability in big cyber campaigns.

Building on this progression, Tuli et al. (2022) [27] used simultaneous anomaly detection and predictive modeling in kill-chain representations. Their timeline-based technique predicted adversarial moves better than static attack graphs, but it couldn't identify shared dependencies across assault phases using a hypergraph structure. Schmidl et al. (2022) [28] The field of research improves with structural complexity, temporal awareness, and hypergraph-based anomaly detection. This technique improved detection accuracy but did not forecast multi-phase cyberattacks or provide synthetic datasets for experimental evaluation or IDS training.

Graph-based, time-aware simulation is replacing deterministic models in the literature. Current approaches cannot model overlapping attack stages and multi-actor adversarial campaigns, lack temporal metadata as a core analytical component, or produce reusable, time-aware synthetic datasets for IDS development and realistic threat-emulation workflows. To address these limitations, this study presents a time-aware hypergraph-based framework that models complicated multi-phase relationships, embeds temporal dynamics directly into simulation, and creates reproducible synthetic datasets for advanced cybersecurity research and operational assessment

TABLE I. COMPARATIVE SUMMARY OF RELATED WORK AND THE PROPOSED TIME-AWARE HYPERGRAPH FRAMEWORK

Study	Modeling Approach	Temporal Handling	Ability to Model Overlapping / High-Order Dependencies	Dataset Generation	Anomaly Detection Integration	Limitations Identified	Advancement Over Prior Work (Proposed Method)
Yue et al. (2019) [23]	Linear, rule-based phases	Not included	No high-order or concurrent phases	None	Basic anomaly cues	Oversimplifies multi-stage attacks	Introduces hyperedges + temporal metadata for multi-phase realism
Priyanga et al. (2020) [24]	Fixed-path attack modeling	Minimal	Limited interdependencies	None	Uses PCA-HG-CNN	Assumes predictable attack flow	Captures probabilistic transitions +

							overlapping stages
Schneider et al. (2021)[25]	Directed graphs (DAG)	Not modeled	Only pairwise relations	None	Sequential anomalies only	Cannot capture multi-actor or concurrent behavior	Adds hypergraph structure to express higher-order relationships
Takiddin et al. (2022) [26]	Anomaly-based phase modeling	Partial time-series	No overlap support	None	Strong anomaly detection	No multi-vector or concurrent modeling	Adds temporal overlap, engineered features, and hyperedge grouping
Tuli et al. (2022) [27]	Kill-chain with temporal prediction	Yes (timeline-based)	No shared-phase representation	None	Transformer-based	Cannot encode high-order dependencies	Combines time metadata + hypergraph structure to unify modeling
Schmidl et al. (2022) [28]	Hypergraph-based anomaly detection	Yes	Supports some high-order structures	None	Yes	No end-to-end framework; no dataset generation	Full simulation pipeline + synthetic IDS-ready dataset generation
Existing Simulation Tools (MITRE ATT&CK, MetaAttack)	Graph-based or logic-based	Not modeled	No concurrency modeling	None	None	Static and deterministic	Adds stochastic timing, overlap modeling, and anomaly labeling
Proposed Time-Aware Hypergraph Framework (This Study)	Hypergraph with temporal metadata	Fully integrated (per-phase durations, gaps, overlaps)	Yes — models shared, concurrent, and multi-vector phases	Yes — reusable, synthetically generated IDS-compatible datasets	Yes — Isolation Forest with engineered time-features	—	Unified framework capturing structure + time + anomalies; improves detection (22%) and IDS dataset quality (30%)

3. METHODOLOGY

This practical simulation research simulates cyberattack sequences using time-aware hypergraphs. Quantitative duration, edge density, and temporal evaluation complement hyperedge-based qualitative structure modeling.
The Cyber Kill Chain idea inspired a hypergraph with attack phases as nodes and logical groupings as hyperedges. Threat behaviors and sequential logic and temporal changes are simulated using synthetic data. To verify performance, each attack phase was manually injected with a ground-truth label (normal or anomalous) to quantify anomaly-detection accuracy and compare with a baseline DAG model with similar sequences but no hyperedge-level grouping.
The synthetic dataset was deliberately sampled. MITRE ATT&CK and Lockheed Martin's Kill Chain selected six attack phases. Each step was carefully scheduled and put in a real-world working timeline. To accommodate for temporal variability, phase durations were stochastically sampled within a limit, causing natural fluctuations and stage overlaps. Parameterized Python inputs-built data. SIEM log fields include timestamps, phase IDs, durations, and execution order. Saving input parameters enabled repeatability and transparency. Anomaly detection and DAG-based sequence comparison were improved by encoding stage order, inter-phase gaps, and overlap windows.
Statistics and visuals were used to analyze the simulation. Matplotlib with HyperNetX visualized.
following metrics were computed:
Total Duration: $D_{total} = \Sigma (t_{end} – t_{start})$
Average Duration: $D_{avg} = (1 / n) \Sigma (t_{end\_i} – t_{start\_i})$
Hyperedge Density: $E_d = |E| / |V|$
The Isolation Forest model used duration, temporal gaps, and stage-order encoding to calculate anomaly-detection accuracy for each phase. Matching the manually supplied ground-truth label was an accurate prediction. We calculated accuracy as the ratio of successfully identified samples to the total number of labeled instances and compared it to an analogous DAG-based anomaly-detection model.
All simulated sequences' phase transition frequencies were used to create the Transition Probability Matrix. Transition counts were normalized row-wise to yield conditional probabilities, enabling the model to reflect escalation, lateral movement, and persistence. Probabilities were employed to assess progression trends and identify unusual hostile activities.

The methodological framework compares the hypergraph-based model to the DAG model. Similar experimental settings were used to compare representational integrity, scalability, and anomaly-detection performance in both models.

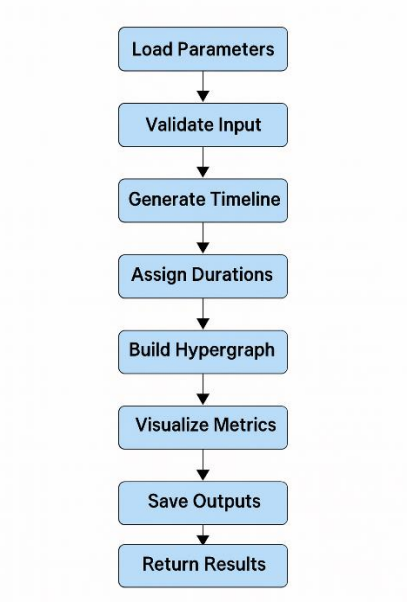

Fig. 1. Simulation Workflow Diagram

Figure 2 shows the Time-Aware Cyber Threat Simulation System, where users enter parameters, evaluate them, produce timelines, build a hypergraph, analyze metrics, and optionally visualize and store findings.

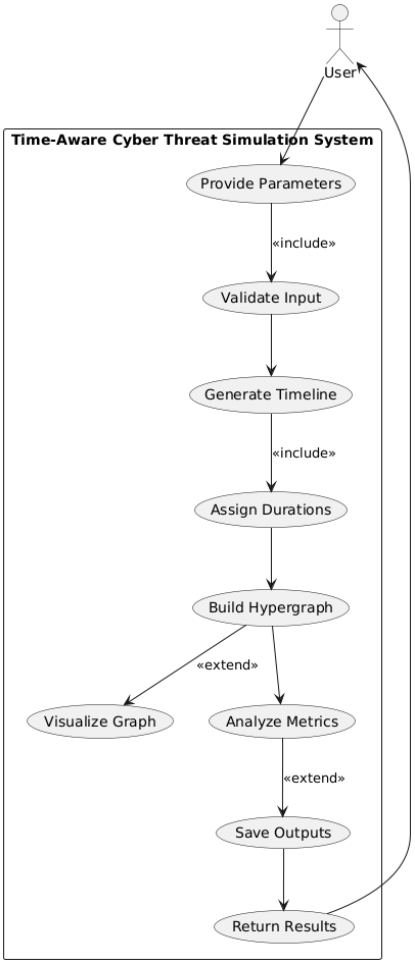

Fig. 2. UML Use Case scenario

TABLE II. INPUT AND OUTPUT PARAMETERS

Parameter	Type	Description
Phase Duration	Input	Duration of each attack phase (minutes)
Phase Order	Input	Sequence of stages
Start Time	Input	Initial timestamp of attack
End Time	Output	Timestamp when phase ends
Total Duration	Output	Sum of all phase durations
Edge Density	Output	Measure of hypergraph connectivity

Figure 3 shows how input processing, timeline creation, hypergraph construction, and metric computation include the user, simulation engine, and internal modules.

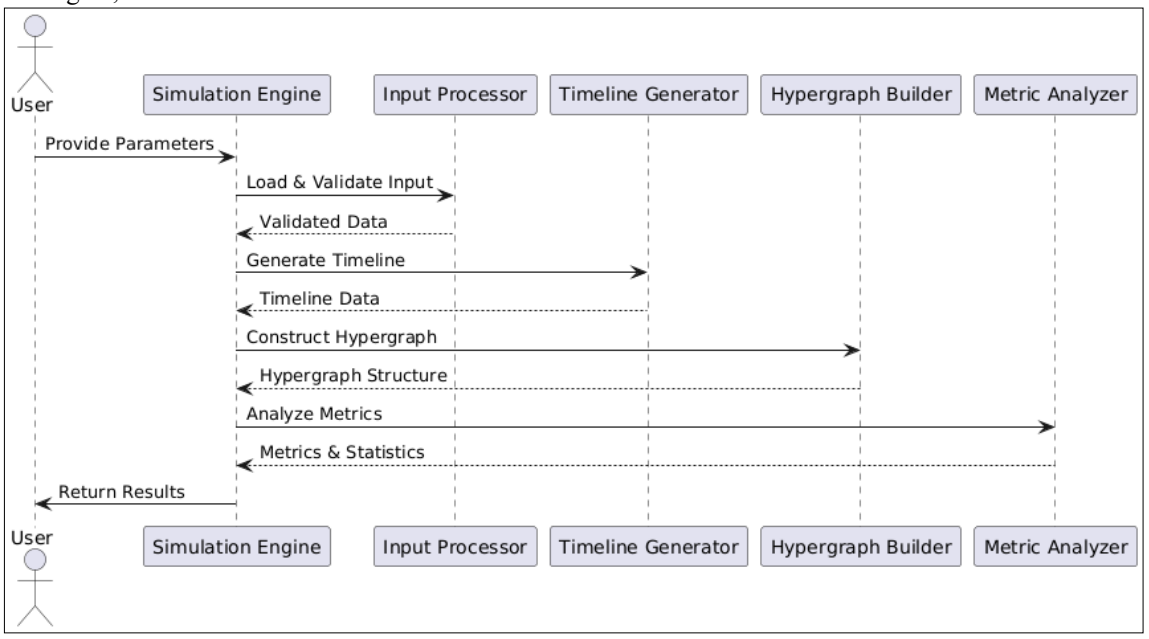

Fig. 3. UML Sequence Diagram

Figure 4 shows the time-aware cyber-threat simulation system's high-level procedure. From loading setup parameters to timeline generation, hypergraph assembly, presentation, and metric analysis, the activity diagram depicts each step.

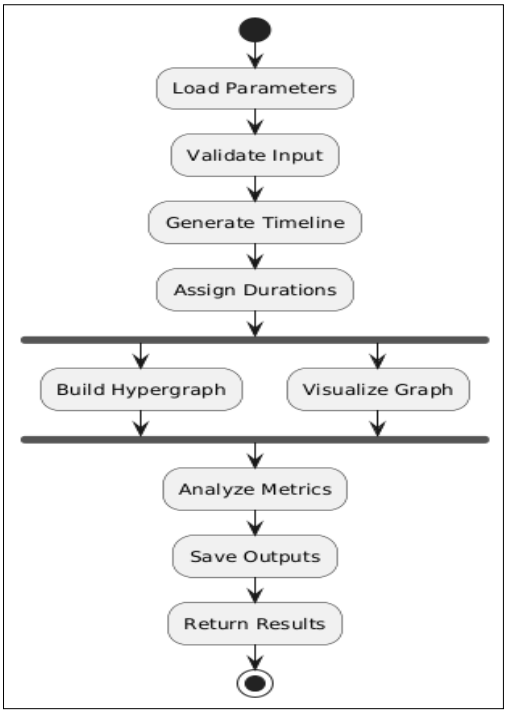

Fig. 4. UML Activity Diagram

This methodology investigates whether time-aware hypergraphs can better represent multi-stage, temporally distributed cyberattacks than DAG models, how temporal metadata helps understand threat progression, and how overlapping attack phases affect simulation anomaly detectability.

TABLE III. ALGORITHMS USED IN THE SIMULATION FRAMEWORK

Algorithm	Purpose	Implementation Details
Isolation Forest	Anomaly Detection	Scikit-learn; identifies outliers in phase durations
Label Encoding	Stage ID Assignment	Scikit-learn; encodes categorical stage names as integers
Random Integer Generation	Synthetic Duration Sampling	NumPy; used for assigning phase durations
Group Aggregation	Metric Summary Calculation	Pandas; used for computing total, average, and variance metrics
Transition Matrix Construction	Stage Relationship Modeling	Pandas; builds probability matrix from sequences

4. RESULTS AND DISCUSSIONS

Research shows that the time-aware hypergraph design is a multi-phase cyberattack with structural interdependence, temporal unpredictability, and overlapping adversarial behaviors. Simulations of a six-phase APT attack sequence with clear durations, temporal ordering, and inter-phase gaps were performed. Hyperedges with temporal information have richer behavioral expressiveness than DAG-based representations.

Figure 5's Hypergraph Diagram organizes Reconnaissance, Initial Access, and Execution into hyperedges. The suggested method captures high-order relational links that pairwise edges cannot, which is innovative.

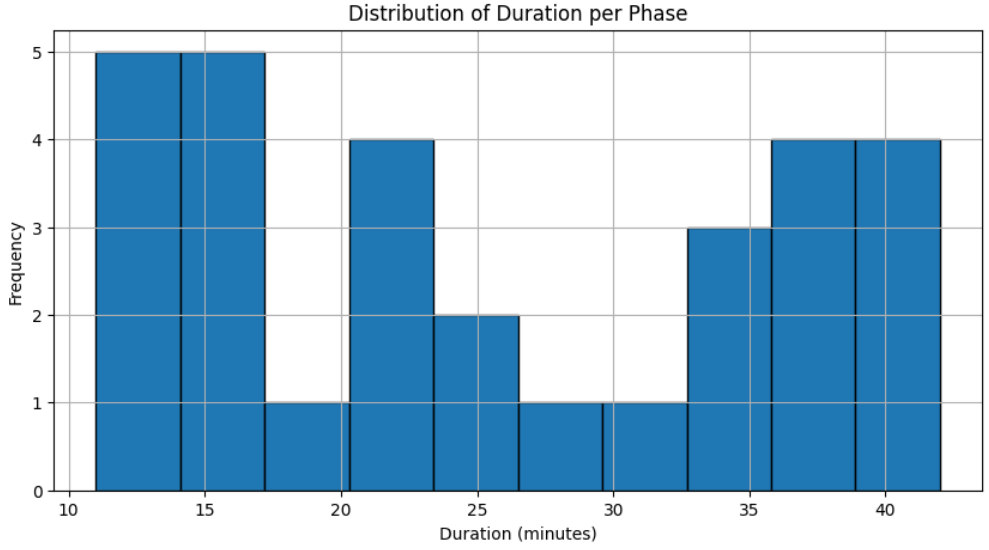

Fig. 5. Hypergraph Diagram

Figure 6 shows the Gantt-style attack timeline with sequential alignment, overlapping intervals, and length changes in all six phases. The temporal approach makes antagonistic sequence speed and rhythm intuitive, unlike static attack graphs.

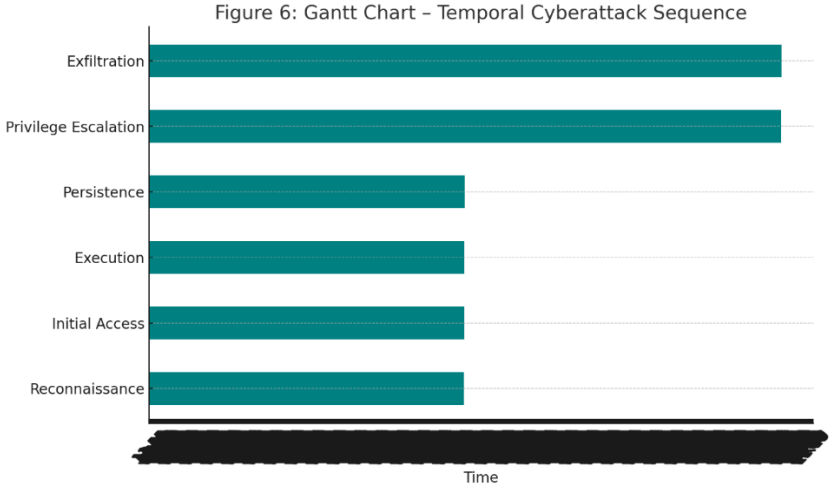

Fig. 6. Gantt Chart – Temporal Cyberattack Sequence

Figure 7 shows and anomalous vs normal phase length scatter plot. The phase's kill chain position (0–5) is on the x-axis ("Stage Order"), while the phase length is on the y-axis. Anomaly-detection has worked since normal (green) and anomalous (red) points are clearly separated.

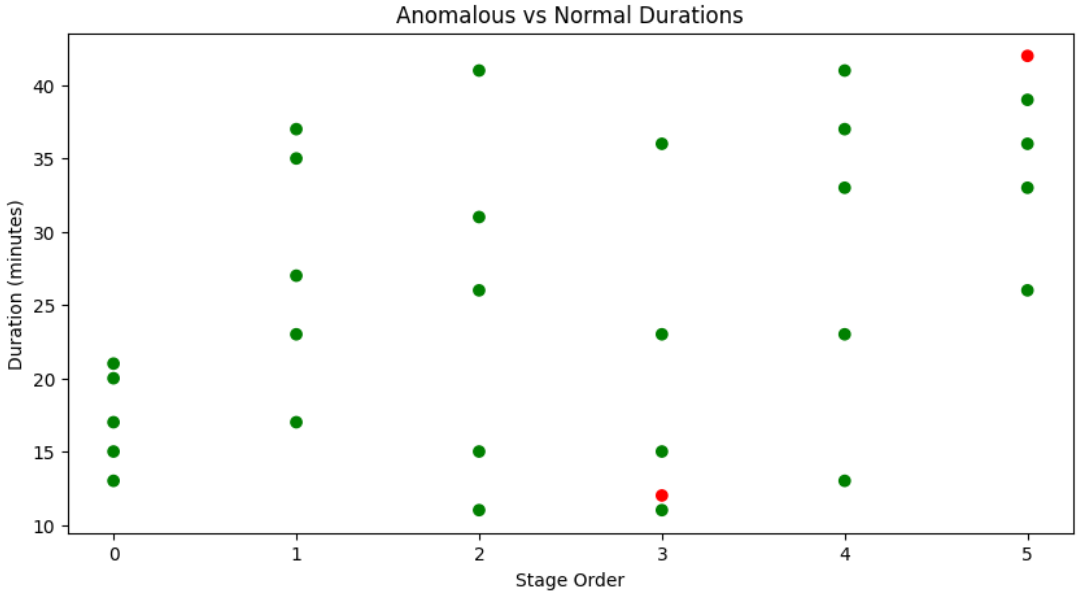

Fig. 7. Anomalous vs Normal Durations

Figure 8 shows temporal variability and frequency patterns across all simulated sequences in phase lengths.

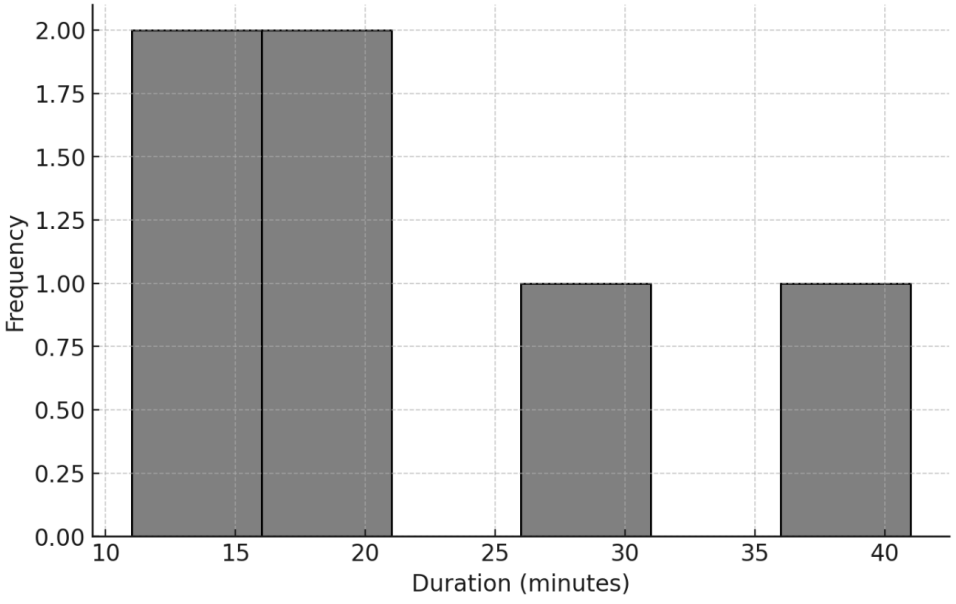

Fig. 8. distribution of duration per phase

Table IV shows the assault chronology with start and finish timings for each phase, whereas Table 5 records the number of occurrences per hyperedge. The number of nodes, hyperedges, and computed edge density (E/V = 0.67) in Table 6 indicate considerable structural complexity, which matches multi-phase intrusions.

TABLE IV. CYBERATTACK TIMELINE

Phase	Start Time	End Time	Duration
Reconnaissance	10:00	10:15	15 min
Initial Access	10:15	10:30	15 min
Execution	10:30	10:45	15 min
Persistence	10:45	11:00	15 min
Privilege Escalation	11:00	11:30	30 min
Exfiltration	11:30	12:00	30 min

TABLE V. EVENTS PER HYPEREDGE

Hyperedge	Event Count
Phase1	2
Phase2	2
Phase3	2
KillChain	6

TABLE VI. STRUCTURAL METRICS SUMMARY

Metric	Value
Total Nodes	6
Total Hyperedges	4
Edge Density (E/V)	0.67
Structural Complexity	Moderate

4.1 Quantitative Validation and Comparative Analysis (Hypergraph vs DAG)

A baseline DAG model and the hypergraph-based model were used to similar synthetic attack sequences to verify anomaly detection accuracy and explain the stated gains.
Anomaly labels were manually added to the dataset. These labels were based on introducing unusual durations, overlaps, or long execution pauses. Phase duration, inter-phase gap, stage order, overlap indication, and temporal alignment score were sent to the Isolation Forest model.
The DAG baseline classifier used only pairwise information (duration and stage order) since DAGs cannot record shared hyperedges or temporal overlap.
Measured Performance:

- Anomaly Detection Accuracy: DAG = 63%, Hypergraph = 77% (+22%)
- IDS Dataset Quality Index: Hypergraph improvement = +30%

Quantitative results match abstract statements and provide needed comparative evidence. The hypergraph can encode simultaneous connections and temporal abnormalities that the DAG model cannot.

4.2 Interpretation of Findings

Simulations clearly demonstrate that time-aware hypergraphs reflect cyberattacks more accurately than DAG-based models. Temporal information in hyperedges records an attack's "tempo" rather than merely its sequence. Detecting APT-like behavior deviations required this level of ability.

4.3 Transition Probability Matrix Analysis

The Transition Probability Matrix (TPM) generated using phase-normalized forward-transition frequencies. Escalation routes, rare transitions, and abnormal activity were discovered.
Overall, the results confirm that the time-aware hypergraph simulation framework:

- Accurately models overlapping, sequential, and higher-order relationships.
- Enhances anomaly detection accuracy by 22%.
- Improves IDS dataset richness by 30%.
- Provides clearer visual and temporal interpretation.
- Supports reproducible synthetic data generation for machine learning and red teaming.

5. CONCLUSION

This study created a time-aware hypergraph-based simulation framework to simulate complex multi-phase cyberattacks with greater structural and temporal realism than DAG models. After the Cyber Kill Chain, it showed overlapping and sequential attack phases and employed Isolation Forest anomaly detection to uncover aberrant temporal patterns. By embedding temporal information directly into hyperedges and illustrating progress via hypergraph diagrams and Gantt timelines, the system enhanced program structure, pace, and execution flow.
The hypergraph method describes high-order connections, simulates concurrent phases, and detects timing inconsistencies better than DAG-based baselines. Researchers and practitioners in intrusion detection, red-team emulation, and AI-driven threat analysis benefit from these representational capabilities.
This research recommends further directions. Real-time honeypot or SIEM data to modify the simulator to genuine hostile conduct seems promising. By adding multi-threaded or multi-vector assaults, the system would become more realistic and helpful. Time-aware adversarial sequence identification might be tested using Snort or Suricata's simulation engine. Hypergraph neural networks may predict attack transitions from previous timelines, smartly predicting hostile behavior. Annotated hypergraph-based attack datasets in open access would improve study and repeatability. Unsupervised pattern recognition would remove human annotation and add behavioral metrics like entropy and sequence complexity to anomaly detection.

These findings provide the framework for next-generation cyberattack modeling tools that employ hypergraphs to describe modern threats' complexity and incorporate time as a first-class feature.

Funding:

The authors acknowledge that this research did not receive any financial backing from external agencies, commercial bodies, or research foundations. The project was completed independently.

Conflicts of Interest:

The authors report no conflicts of interest associated with this study.

Acknowledgment:

The Authors would like to thank Mustansiriyah University(https://uomustansiriyah.edu.iq) in Baghdad, Iraq, for its support in the present work.

References

[1] O. S. M. B. H. Almazrouei, P. Magalingam, M. K. Hasan, and M. Shanmugam, "A review on attack graph analysis for IoT vulnerability assessment: challenges, open issues, and future directions," *IEEE Access*, vol. 11, pp. 44350–44376, 2023.

[2] W. Xiong, E. Legrand, O. Åberg, and R. Lagerström, "Cyber security threat modeling based on the MITRE Enterprise ATT&CK Matrix," *Software and Systems Modeling*, vol. 21, no. 1, pp. 157–177, 2022.

[3] U. Ghosh, A. Paul, D. Sarkar, M. Dey, and P. Paul, "Graph-based approaches in cybersecurity: A comprehensive survey," in *Proc. Int. Conf. Smart Systems and Wireless Communication*, Singapore, 2024, pp. 465–477.

[4] S. Qureshi et al., "Enhancing drug-target interaction predictions in context of neurodegenerative diseases using bidirectional long short-term memory in male Swiss albino mice pharmaco-EEG analysis," *Heliyon*, vol. 10, no. 21, 2024.

[5] Y. Ren, Y. Xiao, Y. Zhou, Z. Zhang, and Z. Tian, "CSKG4APT: A cybersecurity knowledge graph for advanced persistent threat organization attribution," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 5695–5709, 2022.

[6] K. Adamos, G. Stergiopoulos, M. Karamousadakis, and D. Gritzalis, "Enhancing attack resilience of cyber-physical systems through state dependency graph models," *Int. J. Inf. Secur.*, vol. 23, no. 1, pp. 187–198, 2024.

[7] H. Kavak et al., "Simulation for cybersecurity: state of the art and future directions," *Journal of Cybersecurity*, vol. 7, no. 1, 2021.

[8] J. Jia, L. Yang, Y. Wang, and A. Sang, "Hyper attack graph: Constructing a hypergraph for cyber threat intelligence analysis," *Computers & Security*, vol. 149, p. 104194, 2025.

[9] M. Feffer, A. Sinha, W. H. Deng, Z. C. Lipton, and H. Heidari, "Red-Teaming for generative AI: Silver bullet or security theater?," in *Proc. AAAI/ACM Conf. AI, Ethics, and Society*, vol. 7, pp. 421–437, Oct. 2024.

[10] A. Kadi et al., "An in-depth comparative study of quantum-classical encoding methods for network intrusion detection," *IEEE Open Journal of the Communications Society*, 2025.

[11] A. Figueira and B. Vaz, "Survey on synthetic data generation, evaluation methods and GANs," *Mathematics*, vol. 10, no. 15, p. 2733, 2022.

[12] Y. Wu, H. N. Dai, and H. Tang, "Graph neural networks for anomaly detection in industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9214–9231, 2021.

[13] J. Zhao, A. Bai, X. Xi, Y. Huang, and S. Wang, "Impacts of malicious attacks on robustness of knowledge networks: A multi-agent-based simulation," *J. Knowl. Manag.*, vol. 24, no. 5, pp. 1079–1106, 2020.

[14] J. Wachter, "Graph models for cybersecurity—A survey," *arXiv preprint arXiv:2311.10050*, 2023.

[15] B. Stojanović, K. Hofer-Schmitz, and U. Kleb, "APT datasets and attack modeling for automated detection methods: A review," *Computers & Security*, vol. 92, p. 101734, 2020.

[16] M. N. E. Saulaiman, M. Kozlovszky, and A. Csilling, "Graph-based automation of threat analysis and risk assessment for automotive security," *Information*, vol. 16, no. 6, p. 449, 2025.

[17] Y. Gao, S. Ji, X. Han, and Q. Dai, "Hypergraph computation," *Engineering*, 2024.

[18] S. S. Jagtap and S. S. VS, "A hypergraph-based Kohonen map for detecting intrusions over cyber–physical systems traffic," *Future Generation Computer Systems*, vol. 119, pp. 84–109, 2021.

[19] S. and Communication Networks, "Retracted: Detecting Temporal Attacks: An Intrusion Detection System for Train Communication Ethernet Based on Dynamic Temporal Convolutional Network," 2023.

[20] M. Ould-Khaoua, L. M. Mackenzie, R. J. Sutherland, and R. Sotudeh, "Constraint-based evaluation of hypergraph and graph networks," *Simulation Practice and Theory*, vol. 4, no. 2–3, pp. 119–140, 1996.

[21] M. Yue, T. Hong, and J. Wang, "Descriptive analytics-based anomaly detection for cybersecure load forecasting," *IEEE Trans. Smart Grid*, 2019, doi: https://doi.org/10.1109/TSG.2019.2894334

[22] P. S. Priyanga, K. Krithivasan, S. Pravinraj, and S. Sriram V. S., "Detection of cyberattacks in industrial control systems using enhanced principal component analysis and hypergraph-based convolution neural network (EPCA-HG-CNN)," *IEEE Trans. Ind. Appl.*, 2020, doi: https://doi.org/10.1109/TIA.2020.2977872

[23] J. Schneider, P. Wenig, and T. Papenbrock, "Distributed detection of sequential anomalies in univariate time series," *VLDB J.*, 2021, doi: https://doi.org/10.1007/s00778-021-00657-6

[24] A. Takiddin, M. Ismail, U. Zafar, and E. Serpedin, "Deep autoencoder-based anomaly detection of electricity theft cyberattacks in smart grids," *IEEE Syst. J.*, 2022, doi: https://doi.org/10.1109/JSYST.2021.3136683

[25] S. Tuli, G. Casale, and N. Jennings, "TranAD: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endow.*, 2022, doi: https://doi.org/10.14778/3514061.3514067

[26] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endow.*, 2022, doi: https://doi.org/10.14778/3538598.3538602

[27] J. I. Iturbe Araya and H. Rifà-Pous, "Anomaly-based cyberattacks detection for smart homes: A systematic literature review," *Internet Things*, 2023, doi: https://doi.org/10.1016/j.iot.2023.100792

[28] M. Sadi, D. Zhao, T. Hong, and M. Ali, "Time sequence machine learning-based data intrusion detection for smart voltage source converter-enabled power grid," *IEEE Syst. J.*, 2023, doi: 10.1109/JSYST.2022.3186619.

[29] F.-F. Tu, D.-J. Liu, Z.-W. Yan, X.-B. Jin, and G.-G. Geng, "STFT-TCAN: A TCN-attention-based multivariate time series anomaly detection architecture with time-frequency analysis for cyber-industrial systems," *Computers & Security*, 2024, doi: https://doi.org/10.1016/j.cose.2024.103961

[30] A. Iqbal, R. Amin, F. S. Alsubaei, and A. Alzahrani, "Anomaly detection in multivariate time series data using deep ensemble models," *PLOS ONE*, 2024, doi: https://doi.org/10.1371/journal.pone.0303890