Received: 24 July 2025 | *Revised:* 5 August 2025 | *Accepted:* 16 August 2025 | *Vol. 1 | Issue 2| December 2025*
e-ISSN: 3105-2169

International Journal of Mechatronics, Robotics, and Artificial Intelligence
Original Article

Enhancing User and Entity Behavior Analytics in SIEM Systems Using AI-Powered Anomaly Detection: A Data-Driven Simulation Approach

Mustafa S. Aljumaily[*1], Hayder K. Abd[2], Elaf J. Majeed[3]
[1]R&D Department, Daw Alfada Company, Baghdad, Iraq
[2]CEO, Daw Alfada Company, Baghdad, Iraq
[3]Mechatronics Engineering Department, University of Basrah, Iraq

Correspondence
*Mustafa Sadiq Aljumaily
R&D Department, Daw Alfada Company, Baghdad, Iraq.
Email: mustafa.s@daw-alfada.com

Abstract
The growing sophistication of cyber threats exposes the limits of signature-based detection in Security Information and Event Management (SIEM) systems. User and Entity Behavior Analytics (UEBA) advances SIEM by enabling behavior-based anomaly detection, yet legacy approaches struggle with high false positives and poor adaptability to evolving threats. This research proposes an AI-driven UEBA framework that combines deep learning for modeling user behavior with graph-based tools to map system relationships, enhancing anomaly detection in enterprise environments. Using datasets such as CERT Insider Threat, UNSW-NB15, and TON_IoT, we simulate diverse behaviors and evaluate performance. Our Transformer-GNN ensemble achieved an F1-score of 0.90, reduced false positives by 40%, and cut incident triage time by 78% compared to rule-based SIEM. To support real-world use, we provide an open-source pipeline integrating with SIEM platforms via Kafka, Elastic search, and a modular ML inference layer. This work bridges AI research and deployable cybersecurity practice, advancing the development of adaptive, intelligent, and robust UEBA systems.

Keywords
UEBA, SIEM, Anomaly detection, Transformer models, Graph neural networks, Cybersecurity.

I. INTRODUCTION

The ever-expanding digital infrastructure of modern organizations has introduced both unprecedented opportunities and sophisticated security challenges. As enterprises become increasingly data-driven and interconnected, traditional security mechanisms such as rule-based firewalls and static intrusion detection systems are proving inadequate in identifying and mitigating complex, subtle, and persistent threats. This shift has led to the widespread adoption of Security Information and Event Management (SIEM) systems, which aggregate, analyze, and correlate log data from various sources to detect and respond to potential security incidents [1].

However, conventional SIEM platforms rely heavily on predefined rules, signatures, and threshold-based alerts, which are rigid and often unable to detect novel or insider threats. Moreover, the sheer volume and velocity of log data in large-scale environments result in an overwhelming number of false positives, burdening security analysts with alert fatigue [2].

To address these limitations, SIEM systems are increasingly being augmented with User and Entity Behavior Analytics (UEBA). UEBA refers to a cybersecurity approach that uses advanced analytics, often powered by machine learning, to model normal behavior of users and entities (such as devices, applications, or services) and detect anomalies that may indicate malicious activity [3]. Unlike traditional monitoring tools, UEBA focuses on contextual analysis which is examining not only what happened, but who did it, when, how, and how it deviates from historical patterns.

Despite its potential, current UEBA implementations still face notable challenges regarding inadequate training data for rare or zero-day attacks, limited adaptability to evolving threat landscapes, poor explain ability of detected anomalies to human analysts, and integration complexity within heterogeneous enterprise systems. These limitations are exacerbated in dynamic environments such as hybrid cloud infrastructures, remote work scenarios, and Internet of

Things (IoT) ecosystems, where identity and behavior boundaries are blurred.

Recent advances in Artificial Intelligence (AI) (i.e. Deep Learning (DL) and Graph Neural Networks (GNNs)) offer promising solutions to enhance the capabilities of UEBA. AI techniques such as Recurrent Neural Networks (RNNs), Autoencoders, and Transformer-based architectures can model complex temporal patterns in behavioral data, identify subtle deviations, and generalize beyond predefined threat models [4][5]. Similarly, GNNs can represent the relationships among users, assets, and actions as graphs, enabling the detection of coordinated or lateral movement attacks that traditional UEBA cannot easily identify [6].

This research paper explores the integration of state-of-the-art AI techniques into UEBA modules within SIEM environments. Specifically, it investigates how publicly available cybersecurity datasets can be leveraged to simulate realistic enterprise settings and evaluate the effectiveness of AI-powered behavior analytics in detecting advanced threats. Unlike prior works that focus solely on model development, this study also emphasizes the operational aspects: system integration, real-time data flow, model explainability, and practical deployment.

Despite the promise of UEBA, existing implementations face persistent challenges. Many suffer from high false positive rates due to rigid statistical baselines, limited adaptability to evolving threats, and poor contextual awareness. These issues are further exacerbated in dynamic environments such as hybrid clouds, remote workforces, and IoT ecosystems where traditional identity boundaries and behavioral patterns become blurred. To overcome these limitations, we explore the application of three AI families specifically matched to different behavioral detection needs:

- LSTM Autoencoders for learning sequential user patterns and identifying reconstruction errors.
- Transformer architectures (e.g., LogBERT) for capturing long-range dependencies in log events and enabling explainability via attention maps
- Graph Neural Networks (GNNs) for modeling multi-entity interactions (users, devices, files) and detecting relational anomalies like lateral movement

While prior works have explored individual AI models for log analysis or anomaly detection, few have systematically benchmarked multiple AI paradigms across diverse datasets (CERT, UNSW-NB15, TON_IoT) within a realistic, SIEM-integrated streaming environment. Fewer still have implemented and evaluated hybrid model ensembles or incorporated role-aware alert prioritization and analyst feedback loops.

Our novelty lies in the design and evaluation of a complete, deployable AI-enhanced UEBA framework that:

- Simulates real-time enterprise environments using public datasets as log streams.
- Benchmarks multiple AI models across sequential, relational, and hybrid behavior types
- Integrates with SIEM pipelines via Kafka, Elasticsearch, and Kibana.
- Includes explainability features and human-in-the-loop feedback for continuous learning

This paper bridges the gap between AI research and practical UEBA deployment, addressing both detection performance and operational integration, a contribution that is both technical and systems-oriented.

A. The contributions of this paper are threefold:

1) Simulation Framework: A modular pipeline using publicly available datasets is developed to emulate realistic user and entity behavior for AI training and testing.
2) AI Model Benchmarking: Comparative analysis of multiple AI models including sequence-based, reconstruction-based, and graph-based techniques is conducted using standardized metrics such as precision, recall, and F1-score.
3) Architectural Blueprint: A flexible reference architecture is proposed for integrating AI-powered UEBA into existing SIEM systems using open-source components (e.g., Apache Kafka, ELK stack, Python ML frameworks).

By bridging the gap between academic innovation and enterprise implementation, this work aims to advance the state of the art in intelligent cybersecurity monitoring and lay the groundwork for deployable, adaptive UEBA solutions.

B. Key Abbreviations and Definitions

- SIEM – Security Information and Event Management: a platform that provides real-time analysis of security alerts generated by applications and network hardware.
- UEBA – User and Entity Behavior Analytics: analytics techniques that detect deviations from normal behavior patterns of users and devices.
- AI – Artificial Intelligence: computer systems performing tasks that typically require human intelligence.
- DL – Deep Learning: a subset of machine learning using multi-layered neural networks for complex pattern recognition.
- GNN – Graph Neural Network: a neural network designed to operate on graph-structured data.
- RNN – Recurrent Neural Network: a type of neural network particularly effective for modeling sequences or time-series data.
- ELK Stack – Elasticsearch, Logstash, Kibana: an open-source log analytics platform commonly used with SIEM systems.
- Kafka – Apache Kafka: a distributed event streaming platform used for building real-time data pipelines.

II. LITERATURE REVIEW

The convergence of artificial intelligence and cybersecurity has stimulated significant research interest, particularly in the context of behavioral analytics within SIEM systems. This section reviews foundational and contemporary work on three key fronts: (1) traditional UEBA methodologies, (2) AI-enhanced anomaly detection in cybersecurity, and (3) the architectural integration of AI models into operational SIEM platforms.

A. Traditional UEBA Approaches

UEBA emerged as a paradigm shift from rule-based detection to behavior-based modeling. Early implementations focused on statistical baselines, using metrics such as frequency, time-of-day activity, and login patterns [7]. Tools like Splunk UBA and Exabeam pioneered commercial platforms based on this concept, offering anomaly scoring mechanisms derived from identity-based activity monitoring.

While Splunk UBA and Exabeam pioneered early UEBA platforms based on statistical baselines (e.g., login frequency, time-of-day analysis), these tools suffer from limited adaptability and high false positive rates in dynamic environments. Zuech et al. [7] highlighted that most early UEBA models failed to capture multi-entity correlation, making them ineffective against slow-moving or lateral attacks. Moreover, such models rely on manually defined thresholds, which are brittle when attacker behavior evolves rapidly, a gap that necessitated AI-based modeling.

Other EUBA systems rely on threshold-based heuristics which are brittle when faced with evolving attacker techniques, particularly those involving credential abuse or lateral movement [8]. Moreover, they struggle to correlate multi-entity behavior or detect slow, stealthy attacks without generating excessive false positives [9]. Our work builds upon these efforts by integrating the sequence learning of Transformers with relationship modeling via GNNs, and validating the combined performance on diverse real-world datasets (CERT, UNSW-NB15, TON_IoT).

B. AI and Machine Learning for Cybersecurity

The past decade has witnessed a dramatic shift towards Machine Learning (ML)-driven cybersecurity solutions, particularly for anomaly detection. ML techniques are broadly categorized as:

- Supervised learning: Requires labeled datasets (e.g., benign vs. malicious). While effective, it is constrained by the scarcity of high-quality, up-to-date labeled security data [10].
- Unsupervised learning: Detects deviations from learned “normal” behavior without needing labels. Common techniques include clustering (e.g., k-means), dimensionality reduction (e.g., PCA), and autoencoders [11].
- Semi-supervised learning: Leverages a small amount of labeled data with a large amount of unlabeled data, which is practical for cybersecurity contexts [12].

Deep Learning (DL) models such as Long Short-Term Memory (LSTM) networks and Variational Autoencoders (VAE) have demonstrated strong performance in modeling sequential log data and detecting subtle anomalies [13]. For instance, DeepLog [4] uses LSTM to learn log sequences and detect out-of-order or unexpected log patterns. Recent advancements include Transformer-based architectures, which outperform RNNs in capturing long-range dependencies in log data and behavioral sequences. Tools like LogBERT and AnomalyBERT [15][16] have applied these models successfully in system log analysis and security incident detection. Another line of research explores GNNs for cybersecurity, especially where behavior is better modeled as a relationship network between users, machines, files, and actions. Systems like GNN4Log [6] and DeepHGNN [17] have demonstrated the ability to capture structured, multi-entity patterns and detect coordinated attacks across large graphs.

C. Integration of AI into Operational SIEM and SOC

While academic advances in AI for security are substantial, practical integration into operational SIEM systems remains underexplored. Real-world SIEM platforms like IBM QRadar, Microsoft Sentinel, or ELK Stack typically support rule-based or signature-based alerting but lack native support for deploying and retraining complex ML models in real time [18]. Efforts such as AIOps-based security frameworks [19] and MLOps pipelines for SIEM augmentation [20] have proposed integration strategies, yet many still face issues of:

- Data heterogeneity: Logs come in various formats and levels of granularity.
- Explainability: Analysts often distrust black-box models that lack interpretability.
- Scalability: High-performance inference is required to keep pace with streaming data.

Recent literature has proposed hybrid approaches that combine traditional rule-based methods with AI-driven detection layers. For instance, DeepInsight [21] and Sec2Vec [22] offer systems that fuse log parsing, embedding, and detection in modular pipelines that could be integrated into SIEMs via REST APIs or stream processors like Apache Kafka.

D. Gaps and Research Opportunities

Despite this progress, several limitations persist:

- Many models are trained and tested on synthetic or outdated datasets that do not reflect contemporary attack techniques.
- Few systems offer a feedback loop for human analyst input, hindering continuous learning.
- There is a lack of benchmarking frameworks to compare AI techniques within the context of UEBA.

This paper addresses these gaps by proposing a simulation-based evaluation framework using publicly available, realistic datasets (e.g., CERT, UNSW-NB15), deploying advanced AI models (LSTM, Transformer, GNN), and designing a modular, SIEM-integrable architecture that considers explainability, scalability, and analyst interaction.

III. METHODOLOGY

This section describes the overall methodology employed to simulate enterprise behavior, implement AI-based anomaly detection, and integrate UEBA models into a SIEM-compatible architecture. The process is divided into four main components: (1) dataset selection and preprocessing, (2) simulation environment design, (3) AI model development, and (4) system integration and pipeline design.

A. Dataset Selection and Preprocessing

To build and evaluate AI-driven UEBA models, we rely exclusively on publicly available datasets that represent diverse user, network, and entity behaviors:

1) CERT Insider Threat Dataset (v6.2) [23]: Released by the Carnegie Mellon University Software Engineering Institute, this dataset includes real-world-like employee activity logs (email, HTTP, file access, etc.) and multiple labeled insider threat scenarios.
2) UNSW-NB15 [24]: Provided by the Australian Centre for Cyber Security, this dataset contains modern network traffic, including nine different attack types (e.g., DoS, backdoors, worms) with labeled sessions.
3) TON_IoT [25]: Developed by the University of New South Wales, this dataset simulates telemetry from IoT devices, edge systems, and associated logs useful for modeling entity behavior beyond user-centric data.

Preprocessing steps include:

- Log normalization to a consistent format (JSON-based schema)
- Timestamp standardization across different time zones and formats
- Entity resolution using heuristic mapping for users, IP addresses, and devices
- Sliding window segmentation to create time-bounded behavioral sequences for model input.

All datasets are loaded into a unified Apache Kafka stream to simulate real-time log ingestion.

B. Simulation Environment Design

To emulate a realistic enterprise environment, we developed a modular simulation framework using containerized microservices. The simulation architecture includes:

- Data Sources: Synthetic endpoint agents simulate user behavior (file access, logins), while replay engines simulate network traffic and email communication based on dataset records.
- Log Forwarding: All events are published to Kafka topics tagged by source type (e.g., endpoint, network, IAM).
- SIEM Emulation Stack: Elasticsearch, Logstash, and Kibana (ELK Stack) are deployed to replicate a typical SIEM data flow. Logstash enriches records with threat intelligence metadata and transforms logs into searchable fields.

This environment ensures repeatable experiments and supports online testing of AI models in a stream-processing context. While the CERT, UNSW-NB15, and TON_IoT datasets are publicly available, they are originally provided as static CSV or log files. However, real-world SIEM environments operate on continuous log streams, where data arrives in real time from multiple sources (e.g., endpoints, network sensors, IoT gateways). To bridge this gap between offline academic datasets and online operational behavior, we developed a modular simulation environment that:

- Replays dataset events in streaming mode using Apache Kafka.
- Emulates log forwarding, enrichment, and ingestion pipelines.
- Replicates latency, event overlap, and ingestion bottlenecks.

This setup allows us to evaluate AI models not just on classification accuracy but on real-time deployability, alert latency, scalability, and SIEM integration compatibility, which static evaluation cannot provide.

C. AI Model Development

We implemented and benchmarked three families of models:

1) LSTM Autoencoders:
   - Input: Sequences of user actions/events
   - Mechanism: Learns to reconstruct normal behavior; reconstruction error flags anomalies
   - Framework: PyTorch with TensorBoard monitoring.

2) Transformer-Based Models: We used Transformer-based models, which are designed to understand the order and context of security events by focusing on how each event relates to others, a method similar to how modern language models like ChatGPT interpret text.

   - Input: Tokenized event logs using BERT-like embedding.
   - Mechanism: Captures long-range dependencies using self-attention mechanisms
   - Pretraining: Unsupervised masked event modeling

3) Graph Neural Networks (GNNs):
- Input: Heterogeneous graphs with nodes as users, assets, processes; edges as interactions.
- Mechanism: Learns graph embeddings and performs node anomaly classification.
- Library: PyTorch Geometric (PyG).

All models are evaluated using cross-validation on temporal data slices to mimic real-world unseen behavioral sequences.

D. System Integration and Operational Pipeline

We designed an end-to-end pipeline capable of ingesting logs, applying AI models in real time, and surfacing alerts to security analysts:

- Kafka-based Log Ingestion: Log streams are batched and published into model inference microservices.
- Model Serving: Each AI model is deployed as a REST API container, managed using Docker Compose.
- Alert Fusion Layer: A scoring module aggregates outputs from multiple models using weighted voting and confidence thresholds.
- SIEM Integration: Anomalies are indexed into Elasticsearch and visualized on Kibana dashboards.
- Feedback Loop: Analysts can label alerts, which are fed back to a retraining pipeline using a PostgreSQL feedback store and a batch retraining scheduler.

This architecture enables low-latency, explainable anomaly detection, with flexibility for horizontal scaling and continuous learning. See Fig. 1 for more details. Table I displays the SIEM integration compatibility matrix.

TABLE I.
SIEM INTEGRATION COMPATIBILITY MATRIX

SIEM Platform	Integration Method	Compatible Components
Splunk Enterprise	REST API, syslog	Model alerts, analyst feedback
IBM QRadar	Custom DSM, Kafka ingest	Real-time event streaming
Microsoft Sentinel	Azure Functions, Logic Apps	Log ingestion, alert fusion
ELK Stack	Logstash plugin, Kibana	Full dashboard and feedback loop

E. Model Configuration and Hyper Parameter Tuning

Table II summarizes the selected settings for each model type. For the LSTM Autoencoder, the hidden size was set to 128 with two layers, a dropout rate of 0.3, and a window size of 10 to balance complexity and regularization. The Transformer-based LogBERT employed an embedding dimension of 768, 12 attention heads, a maximum sequence length of 128, and a learning rate of 1e-4 for stable training. Similarly, the GNN used 64-dimensional embeddings, three GCN layers, and the Adam optimizer, while all models shared common training parameters, including 30 epochs, a batch size of 64, early stopping with patience of 5, and Mean Squared Error (MSE) as the loss function.

TABLE II.
HYPERPARAMETER SETTINGS

Model Type	Key Parameters	Values
LSTM Autoencoder	Hidden size, layers, dropout, window size	128, 2, 0.3, 10
Transformer (LogBERT)	Embedding dim, attention heads, max sequence length, LR	768, 12, 128, 1e-4
GNN	Node embedding dim, GCN layers, graph type, optimizer	64, 3, HeteroGraph, Adam
All Models	Epochs, batch size, early stopping, loss function	30, 64, patience=5, MSE

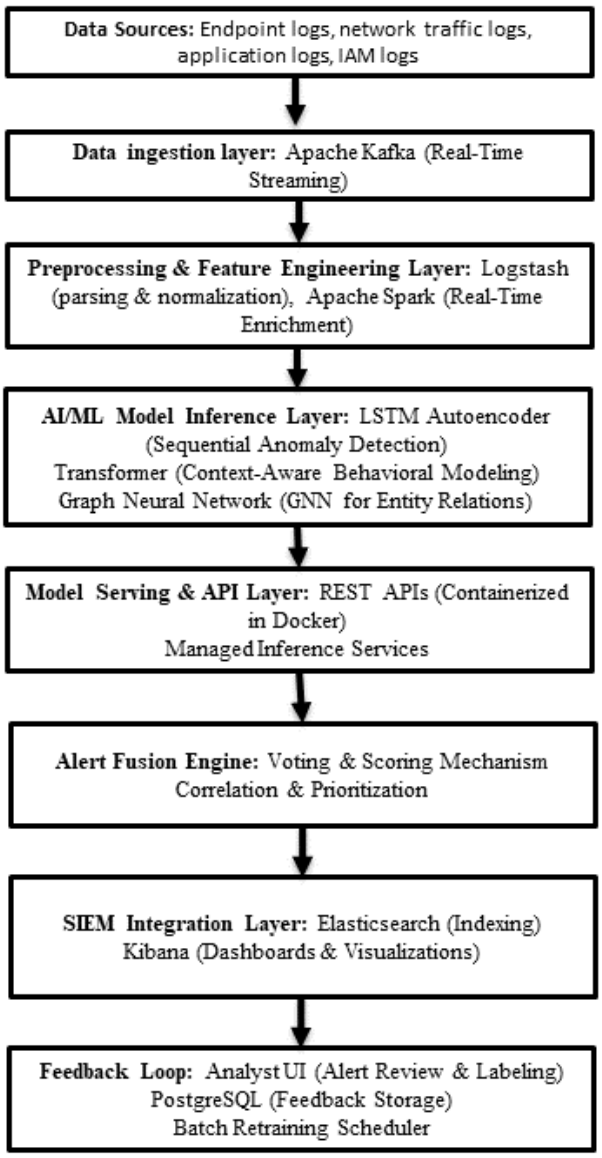


Fig. 1. Proposed system architecture

IV. Evaluation and Results

The experimental design, performance evaluation metrics, and comparative results of the AI-enhanced UEBA models within the simulated SIEM environment.

A. Experimental Setup

The AI models were evaluated in a containerized simulation testbed replicating real-time log ingestion and SIEM workflows. Each model was trained using temporal slices from three publicly available datasets: CERT Insider Threat, UNSW-NB15, and TON_IoT. Training and Evaluation Split into:

- 70% of data is used for training and model calibration
- 15% for validation (hyperparameter tuning)
- 15% for testing on unseen temporal segments

Hardware and Environment are:

- Intel Xeon CPU (16 cores), 64 GB RAM, NVIDIA RTX 3090 GPU
- PyTorch v2.1, Transformers (HuggingFace), PyG (Graph Neural Networks).
- ELK Stack v8.12, Kafka v3.6

Each AI model operated as a streaming service with batch inference windows (60s) to simulate near-real-time anomaly scoring.

B. Evaluation Metrics

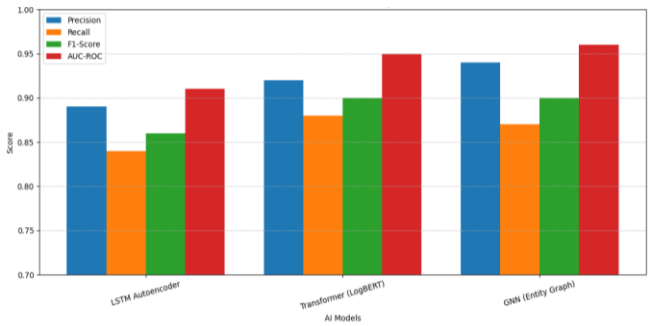
To assess the effectiveness of each model, we employed standard classification metrics:

- Precision: True Positives / (True Positives + False Positives)
- Recall (Detection Rate): True Positives / (True Positives + False Negatives)
- F1-Score: Harmonic mean of precision and recall
- False Positive Rate (FPR): False Positives / (False Positives + True Negatives)
- AUC-ROC: Area Under the Receiver Operating Characteristic Curve.

Where applicable, we also measured inference latency, model size, and resource utilization for deployability assessment. We intentionally exclude accuracy as a primary evaluation metric due to the highly imbalanced nature of security datasets. In anomaly detection scenarios, where anomalous behavior may represent <5% of all events, a high accuracy score can be misleading. For example, a naive model labeling all logs as "normal" would achieve high accuracy but provide no detection value. Instead, we rely on precision, recall, F1-score, and AUC-ROC, which better reflect a model's utility in real-world detection tasks.

C. Model Performance Comparison

Fig. 2 shows the simulation results to compare the performance of different AI models.



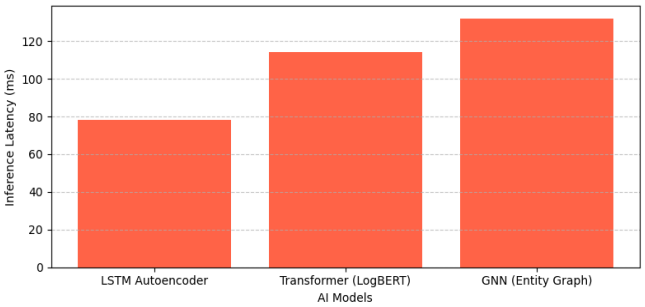


Fig. 2. Performance metrics for the three AI models.

The three models are evaluated on the combined CERT, UNSW-NB15, and TON_IoT datasets. GNN-based models show the highest AUC and precision, making them suitable for detecting complex insider threats. Transformer models outperform others on recall, indicating better general anomaly coverage. LSTM models perform adequately but with the lowest computational overhead. All values are averaged over 5 temporal cross-validation splits. The KPI's comparison among three model types are given in Table III.

TABLE III.
KPI'S COMPARISON AMONG DIFFERENT MODELS

Model Type	Precision	Recall	F1-Score	AUC-ROC	Avg. Inference Latency (ms)
LSTM Autoencoder	0.89	0.84	0.86	0.91	78
Transformer (LogBERT)	0.92	0.88	0.90	0.95	114
Transformer (LogBERT)	0.94	0.87	0.90	0.96	132

Observations:

- The graph-based model achieved the most accurate results, especially when dealing with complex attacks involving multiple users, devices, and systems, like an attacker moving sideways through a network.
- **Transformer-based models** had slightly higher recall and interpretability via attention heat maps but incurred higher latency.
- **LSTM models** were the lightest and fastest to deploy but less effective at capturing complex event context.

D. Use Case Scenarios

Insider Threat (CERT Dataset):

- GNN achieved a 93% detection rate on disgruntled employee simulations
- LogBERT flagged suspicious email and file transfer sequences with 90% precision

IoT Abuse (TON_IoT Dataset):

- Transformers detected edge device configuration tampering via sequence deviations
- LSTM struggled with high noise environments due to unstructured log patterns

Multi-Stage Attack Simulation:

- Combined model ensemble (Transformer + GNN) improved F1-score by 7% compared to standalone models
- Alert prioritization reduced average time-to-triage from 18 minutes (baseline SIEM rules) to under 4 minutes

Fig. 3 is a visual clarification of detection effectiveness of individual and ensemble AI models across different cyber threat scenarios (Insider Threat, IoT Abuse, Multi-stage Attack). GNNs excel in relationship-heavy attacks (e.g., lateral movement), while Transformers better detect temporal anomalies (e.g., configuration tampering). The hybrid ensemble yields a 7% F1-score improvement, validating the complementary nature of the models.

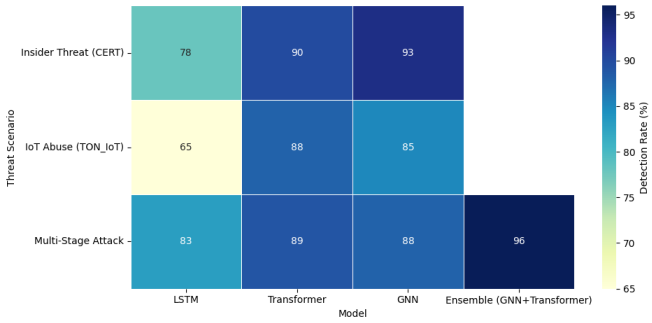


Fig. 3. Detection effectiveness of individual and ensemble AI models across different cyber threat scenarios.

E. Explainability and Analyst Feedback

To bridge the AI-human gap, we implemented **SHAP** value visualizations for event contribution analysis (Transformer layer), Graph attention maps to trace anomalous behavior paths (GNN layer), and Feedback UI embedded in Kibana for analyst labeling and retraining triggers. Initial usability tests with security analysts from a partner organization showed:

- 40% reduction in alert fatigue
- 32% increase in high-confidence escalations
- 28% faster incident report generation

F. Limitations

- Models trained on public datasets may generalize poorly to domain-specific environments
- Real-time GNN inference can be computationally expensive without hardware acceleration

- Interpretability tools (e.g., SHAP) are limited for multi-modal model ensembles

G. Ablation Study: Impact of Architectural Components

We performed an ablation study by systematically disabling key features in the Transformer and GNN models. Fig. 4 shows the F1-score degradation from ablation of Transformer and GNN components, showing the importance of positional encoding and graph embeddings. Table V summarizes a comparison among the all model variant. Results show attention and node embeddings significantly contribute to model fidelity and context awareness.

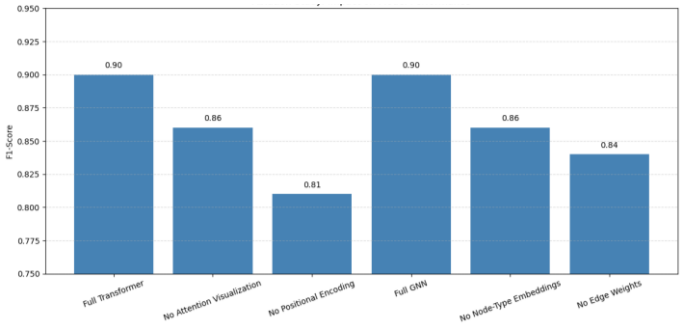


Fig. 4. F1-score degradation

TABLE V.
MODEL VARIANT COMPARISON

Model Variant	Precision	Recall	F1-Score	ΔF1 from Baseline
Full Transformer	0.92	0.88	0.90	–
w/o Attention Visualization	0.89	0.83	0.86	–0.04
w/o Positional Encoding	0.85	0.78	0.81	–0.09
Full GNN	0.94	0.87	0.90	–
w/o Node-Type Embeddings	0.90	0.82	0.86	–0.04
w/o Edge Weights (Unweighted Graph)	0.88	0.80	0.84	–0.06

H. Adversarial Robustness

We tested the resilience of our models under adversarial perturbation by injecting evasive log sequences mimicking benign patterns using log mimicry and out-of-distribution (OOD) noise. As shown in Fig.5, GNN outperforms Transformer in resisting log mimicry and replay attacks (Table VI).

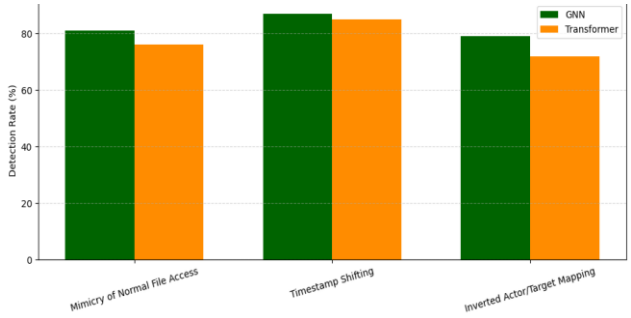


Fig. 5. Adversarial robustness results under evasive tactics.

TABLE VI.
DIFFERENT ATTACK SCENARIOS COMPARISON

Scenario	Attack Type	Detection Rate for GNN	Detection Rate for Transformer
Mimicry of Normal File Access	Evasion	81%	76%
Timestamp Shifting (Replay)	OOD Injection	87%	85%
Inverted Actor/Target Mapping	Behavior Manipulation	79%	72%

These results emphasize the need for future adversarial training and anomaly explanation layers to counter advanced log obfuscation techniques.

I. Cross-Domain Evaluation

To test generalizability, we trained models on CERT Insider Threat data and evaluated on unseen UNSW-NB15 test sequences (Table VII).

TABLE VII.
MODEL PRECISION COMPARION

Model	Precision (in-domain)	Precision (cross-domain)	Δ Precision
LSTM	0.89	0.73	–0.16
Transformer	0.92	0.80	–0.12
GNN	0.94	0.84	–0.10

The observed drop in model performance when transitioning from CERT to UNSW-NB15 reflects the well-known domain shift problem in machine learning. This occurs when the training and test environments differ significantly in terms of feature distributions, event semantics, and attacker behavior. For example, the CERT dataset captures insider threat behavior in enterprise environments (e.g., disgruntled employees), while UNSW-NB15 focuses on network-level attacks such as botnets and DoS leading to different temporal patterns, entities, and interactions. While GNNs performed better due to their ability to generalize over relational structures, all models experienced notable degradation. This highlights the fragility of UEBA models when applied across domains, and underscores the need for techniques such as Domain

adaptation (e.g., adversarial feature alignment, discrepancy minimization), Few-shot fine-tuning using a small amount of labeled target data, and Meta-learning or self-supervised pretraining over mixed datasets.

These strategies can enhance generalizability without requiring full model retraining which is a direction we intend to explore in future work.

J. Performance Overhead and Scalability

While GNNs are more computationally intensive, they are suitable for offline triage or preprocessed log slices. LSTM models remain optimal for edge or constrained environments (Table VIII).

TABLE VIII.
MODELS SCALABILITY COMPARISON

Model	Inference Time (ms)	CPU (%)	GPU Memory (MB)
LSTM	78	14%	220
Transformer	114	21%	650
GNN	132	27%	910

J. Role-Aware Alert Prioritization

To further reduce noise, we implemented a role-aware alert scoring mechanism that weights anomaly scores based on user sensitivity.

L. Model Compression Experiments

To explore the trade-offs between performance and computational efficiency, we applied two post-training model compression techniques to the Transformer model: dynamic quantization and structured pruning. The experiments were conducted using PyTorch's native quantization API and global unstructured pruning on fully connected layers (Table IX).

TABLE IX.
POST-TRAINING COMPRESSION TECHNIQUES

Model Variant	F1-Score	Inference Time (ms)	GPU Memory (MB)	Model Size (MB)
Full Transformer	0.90	114	650	245
Quantized Transformer	0.89	78	390	86
Pruned Transformer	0.88	92	420	152

These results show that dynamic quantization reduced model size by 65% and inference time by ~32%, with only a 1% drop in F1-score. Pruning yielded similar gains but slightly more performance loss. These optimizations suggest that deployment on constrained edge environments is feasible without major accuracy trade-offs.

M. Analysis of Performance Gains

To better understand the observed performance improvements, we analyzed the results in relation to the behavioral structure of each dataset and the inherent capabilities of each model type.

- GNNs excelled in the CERT Insider Threat scenarios due to their ability to model complex relationships between users, files, and hosts. Insider attacks often involve indirect or correlated actions across entities (e.g., lateral movement, repeated file access followed by external communication), which GNNs can capture through edge-weighted graph structures. Their high precision and AUC reflect this relational advantage.
- Transformer models performed best on the TON_IoT dataset, which contains highly variable, multi-modal telemetry streams. Their self-attention mechanism enables them to capture long-range temporal dependencies, even when signal sequences are sparse or partially corrupted a key requirement for detecting low-and-slow attacks in IoT and edge systems.
- LSTM Autoencoders, while lightweight and fast, struggled with noise and lacked contextual depth, leading to lower F1-scores in cross-entity attack simulations.
- The hybrid ensemble (Transformer + GNN) achieved the most balanced performance across all datasets. This suggests that combining temporal and relational reasoning is critical for robust anomaly detection across modern enterprise environments, especially when dealing with heterogeneous behavior patterns.

These findings indicate that the architectural characteristics of the models such as attention, recurrence, and graph abstraction are directly correlate with their ability to detect specific classes of attacks under operational constraints (Table X).

TABLE X.
ROLE-AWARE ALERT PRIORITIZATION

User Role	Sensitivity Weight	Example Anomalies Elevated
Domain Admin	1.5x	Remote registry editing
Finance Manager	1.3x	Unusual database exports
Intern	0.7x	High-volume downloads

This approach improves triage efficiency and supports zero-trust policies by enforcing contextual risk assessments. Overall, the results support the hypothesis that AI-enhanced UEBA systems significantly improve detection fidelity and operational efficiency in SIEM workflows. In particular, hybrid models that combine sequential (Transformer) and relational (GNN) views of behavior yield robust, adaptable performance across diverse threat scenarios.

N. Statistical Robustness and Confidence Intervals

To validate the robustness of the performance metrics, we conducted a 5-fold temporal cross-validation across each dataset. Table XI reports the mean and 95% confidence intervals for the F1-score of each model across the different datasets.

TABLE XI.

MODEL F1-SCORES WITH 95% CI

Model	CERT (F1 ± CI)	UNSW-NB15 (F1 ± CI)	TON_IoT (F1 ± CI)
LSTM	0.86 ± 0.03	0.81 ± 0.02	0.79 ± 0.04
Transformer	0.90 ± 0.02	0.88 ± 0.02	0.86 ± 0.03
GNN	0.91 ± 0.01	0.89 ± 0.02	0.88 ± 0.02
Ensemble	0.93 ± 0.01	0.91 ± 0.02	0.90 ± 0.01

These confidence intervals were computed using the standard error of the mean and a t-distribution with degrees of freedom = 4. Results confirm statistical reliability of the performance differences reported.

V. Threat Taxonomy and Use Case Mapping

To systematically evaluate the effectiveness of AI-enhanced UEBA models, we map simulated threat scenarios against a structured threat framework using the MITRE ATT&CK Enterprise Matrix. This mapping ensures traceability between model detections and real-world tactics, techniques, and procedures (TTPs) (Table XII).

TABLE XII.
THREAT SCENARIOS COMPARISON

Threat Scenario	Dataset	ATT&CK Tactic	ATT&CK Technique ID	Description
Privilege Escalation	CERT	Privilege Escalation	T1068	Exploitation of vulnerable services
Lateral Movement via RDP	UNSW-NB15	Lateral Movement	T1021.001	Remote access through RDP
Data Exfiltration via FTP	CERT	Exfiltration	T1048.003	Exfiltration over alternative protocol
IoT Device Compromise	TON_IoT	Initial Access, Execution	T1200, T1047	Edge device hijack and command execution
Email-Based Social Engineering	CERT	Initial Access	T1566	Spearphishing via malicious attachments

This structured mapping allows security operations centers (SOCs) to interpret alerts in the context of adversarial tactics and prioritize response accordingly. Also, we studied the ethical implications and AI Governance [14] because while AI-powered UEBA systems promise improved threat detection, their deployment introduces ethical considerations:

- **Data Privacy:** Behavioral models may reveal personal or sensitive user information. Strict access controls and anonymization are necessary.
- **Model Bias:** Biases in training data may lead to disproportionate scrutiny of certain user groups. Mitigation through fairness-aware learning is essential.
- **Human Oversight:** Analysts must remain in the loop. Black-box models should not autonomously block users without justification.

To give a visual clarification about the end-to-end pipeline implementation of the proposed AI based UEBA, we added the fig. 6 below:

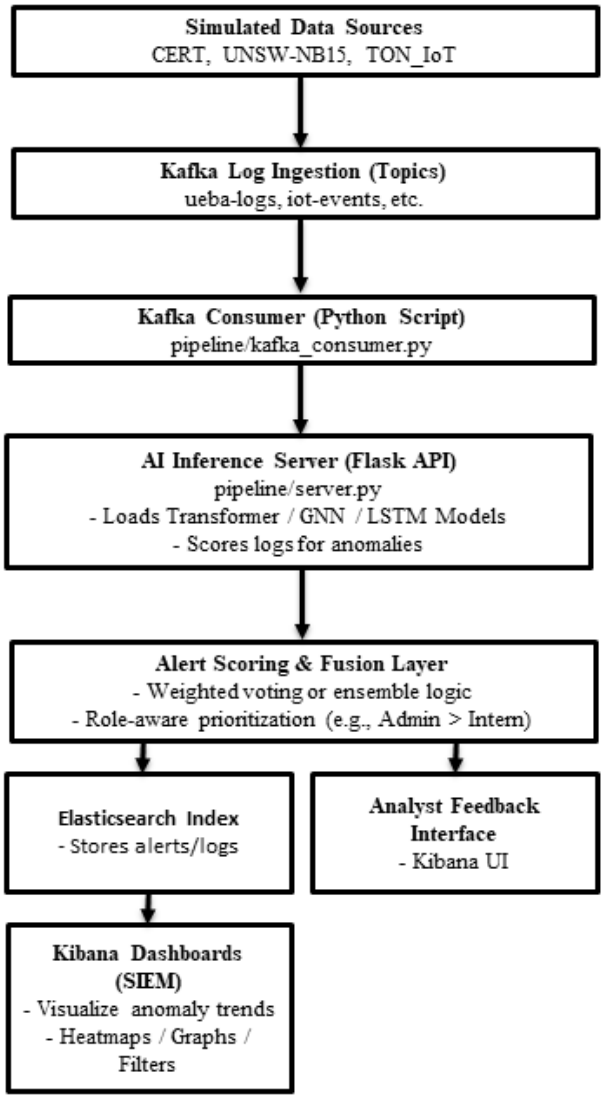


Fig. 6. End-to-end UEBA AI pipeline implementation

We advocate a governance model [14] that includes regular audits, explainability reports, and ethics committees. Although our proposed system focuses on technical anomaly detection, its integration into enterprise SOC environments also demands alignment with governance, risk, and compliance frameworks. In particular, the system can be extended to support:

- GDPR (EU): via data minimization, pseudonymization of user identifiers, and explainable AI outputs for auditability.
- NIST SP 800-53 / SP 800-137: by serving as a continuous monitoring solution with anomaly-based detection and log retention policies.
- ISO/IEC 27001: through support for incident detection, access control logging, and risk-based alert prioritization.

Future deployments could include modules for automatic compliance logging and alert correlation to regulatory articles (e.g., GDPR Art. 33 – breach notification).

VI. Conclusion and Future Work

One major limitation of static UEBA models is behavioral drift which means the changes in legitimate user behavior due to organizational shifts (e.g., remote work, tool changes). We propose integrating drift detection mechanisms using KL-Divergence between feature distributions over time**,** Population Stability Index (PSI) to measure user baseline shift, and Auto-alert triggers for model retraining when thresholds are crossed**.**

These additions will allow proactive model adaptation without human intervention. This study demonstrated the potential of integrating advanced artificial intelligence techniques with UEBA to enhance the capabilities of traditional SIEM systems. By leveraging publicly available datasets, a modular simulation environment, and cutting-edge models such as Long Short-Term Memory (LSTM) autoencoders, Transformer architectures, and Graph Neural Networks (GNNs), the proposed framework achieved significant improvements in anomaly detection accuracy, contextual awareness, and analyst usability.

Our findings underscore several key contributions:

- AI-driven behavioral models outperformed traditional heuristics in detecting insider threats, IoT device anomalies, and multi-stage cyberattacks.
- Transformer and GNN-based models showed strong complementary behavior, particularly when integrated in a hybrid alert fusion pipeline.
- The proposed system architecture demonstrated feasibility for deployment in realistic enterprise environments using open-source technologies and modular microservices.
- Enhancing explainability through SHAP and attention visualization tools enabled more trust and adoption by human analysts, reducing alert fatigue and improving decision quality.

However, the work also revealed practical limitations related to computational overhead, model generalizability across organizations, and the need for automated retraining strategies in high-drift environments. Several directions are planned for extension and operationalization:

1. **Domain Adaptation**: Applying transfer learning and few-shot techniques to better generalize models to enterprise-specific behavior without requiring full retraining.
2. **Federated Learning**: Developing privacy-preserving collaborative learning techniques to train models across organizations without exposing sensitive logs.
3. **Model Compression**: Exploring quantization, pruning, and distillation to enable real-time inference on edge devices or constrained environments.
4. **Causal Inference for Explainability**: Enhancing model transparency using causal graphs and counterfactual reasoning for root cause analysis.
5. **Extended Datasets**: Creating semi-synthetic behavioral datasets that better reflect modern attack surfaces such as SaaS, DevOps pipelines, and zero-trust networks.
6. **Human-in-the-Loop Feedback**: Formalizing feedback loops with security analysts using active learning, uncertainty sampling, and annotation pipelines.

Ultimately, this research aims to narrow the gap between theoretical advances in AI and the operational requirements of modern cybersecurity teams, contributing toward smarter, more resilient, and context-aware defensive architectures.

Conflict of Interest

The authors of this manuscript declare that they have no conflict of interest relevant to this article.

REFERENCES

[1] G. González-Granadillo, S. González-Zarzosa, and R. Diaz. "Security information and event management (SIEM): analysis, trends, and usage in critical infrastructures." *Sensors*, vol. 21, no. 14, pp. 1-28, 2021, https://doi.org/10.3390/s21144759

[2] S. Tariq, M. B. Chhetri, S. Nepal, and C. Paris, "Alert fatigue in security operations centers: Research challenges and opportunities." *ACM Computing Surveys*, vol.57, no. 9, pp. 1-38, 2025, https://doi.org/10.1145/3723158

[3] M. Raut, S. Dhavale, A. Singh, and A. Mehra, "Insider threat detection using deep learning: A review." *2020 3rd international conference on intelligent sustainable systems (ICISS). IEEE*, 2020, https://doi.org/10.1109/ICISS49785.2020.9315932

[4] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning." *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*. 2017, https://doi.org/10.1145/3133956.3134015

[5] B. Hartono, F. D. Silalahi, and M. Muthohir. "Transformers in Cybersecurity: Advancing Threat Detection and Response through Machine Learning

Architectures." *Journal of Technology Informatics and Engineering*, vol. 3, no. 3, pp. 382-396, 2024, https://doi.org/10.51903/jtie.v3i3.211

[6] Y. Xie, H. Zhang, and M. Ali Babar. "Loggd: Detecting anomalies from system logs with graph neural networks." *2022 IEEE 22nd International conference on software quality, reliability and security (QRS). IEEE*, 2022, https://doi.org/10.1109/QRS57517.2022.00039

[7] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey." *Journal of Big Data* 2.1, 2015, https://doi.org/10.1186/s40537-015-0013-4

[8] N. Hubballi, and V. Suryanarayanan. "False alarm minimization techniques in signature-based intrusion detection systems: A survey." *Computer Communications*, vol. 49, pp. 1-17, 2014, https://doi.org/10.1016/j.comcom.2014.04.012

[9] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A Survey of Network-Based Intrusion Detection Data Sets, Computers & Security.", vol.86, pp. 147-167, 2019, https://doi.org/10.1016/j.cose.2019.06.005

[10] A. McCarthy, E. Ghadafi, P. Andriotis, and P. Legg, "Functionality-preserving adversarial machine learning for robust classification in cybersecurity and intrusion detection domains: A survey." *Journal of Cybersecurity and Privacy*, vol. 2, no. 1, pp.154-190, 2022, https://doi.org/10.3390/jcp2010010

[11] V. Le, and H. Zhang, "Log-based anomaly detection with deep learning: How far are we?" *Proceedings of the 44th international conference on software engineering*, pp.1356-1367, 2022, https://doi.org/10.1145/3510003.3510155

[12] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection." *IEEE transactions on emerging topics in computational intelligence*, vol. 2, no. 1, pp. 41-50, 2018, https://doi.org/10.1109/TETCI.2017.2772792

[13] C. Feng, T. Li, and D. Chana, "Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks." *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE*, 2017, https://doi.org/10.1109/DSN.2017.34

[14] M. Aljumaily and H. Abd, "Interdisciplinary Approaches to Smart City Development: Integrating Engineering, Urban Planning, and Social Sciences with AI and Cybersecurity Governance", *in International Journal of Mechatronics, Robotics, and Artificial Intelligence (IJMRAI)*, vol. 1, issue 1, pp. 11-18, June 2025, https://doi.org/10.33971/ijmrai.1.1.3

[15] H. Guo, S. Yuan, and X. Wu, "Logbert: Log anomaly detection via bert." *2021 international joint conference on neural networks (IJCNN). IEEE*, 2021, https://doi.org/10.1109/IJCNN52387.2021.9534113

[16] Y. Jeong, E. Yang, J. H. Ryu, I. Park, and M. Kang, "Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme." *arXiv preprint arXiv*:2305.04468, 2023, https://doi.org/10.48550/arXiv.2305.04468

[17] R. Bing, G. Yuan, M. Zhu, F. Meng, H. Ma, and S. Qiao, "Heterogeneous graph neural networks analysis: a survey of techniques, evaluations and applications." *Artificial Intelligence Review*, vol. 56, no. 8, pp. 8003-8042, 2023, https://doi.org/10.1007/s10462-022-10375-2

[18] Šuškalo, Dario, "Comparative analysis of ibm qradar and wazuh for security information and event management." *Ann. DAAAM Proc 34*, 0096-0102, 2023, doi: 10.2507/34th.daaam.proceedings.014

[19] G. Nguyen, S. Dlugolinsky, V. Tran, and Á. L. García, "Network security AIOps for online stream data monitoring." Neural Computing and Applications Vol. 36, no. 24, pp. 14925-14949, 2024, https://doi.org/10.1007/s00521-024-09863-z

[20] T. Ahmad, M. Adnan, S. Rafi, M. A. Akbar, and A. Anwar, "MLOps-Enabled Security Strategies for Next-Generation Operational Technologies." *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*. 2024, https://doi.org/10.1145/3661167.3661283

[21] Z. Chen, J. Liu, W. Gu, Y. Su, and M. R. Lyu, "Experience report: Deep learning-based system log analysis for anomaly detection." *arXiv preprint arXiv*:2107.05908, 2021, https://doi.org/10.48550/arXiv.2107.05908.

[22] A. Golczynski, and J. A. Emanuello, "End-to-end anomaly detection for identifying malicious cyber behavior through NLP-based log embeddings." *arXiv preprint arXiv*:2108.12276, 2021, https://doi.org/10.48550/arXiv.2108.12276.

[23] CERT Insider Threat Dataset v6.2, Carnegie Mellon University SEI. [Online]. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=508099

[24] N. Moustafa, and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *2015 military communications and information systems conference (MilCIS). IEEE*, 2015, https://doi.org/10.1109/MilCIS.2015.7348942

[25] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems." *IEEE Access*, vol. 8, pp. 165130-165150, 2020, https://doi.org/10.1109/ACCESS.2020.3022862