Journal of Information Systems Engineering and Management
2025, 10(22s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

GraphSeqDetect: A Hybrid Machine Learning Leveraged Adaptive Learning Framework for Insider-Driven DDoS Detection in Cloud Environments

K. Balachandra Reddy[1], Dr. S. Meera[2]
[1]*Research Scholar, Vels Institute of Science, Technology and Advanced Studies, Chennai, India.*
[2]*Associate Professor, Vels Institute of Science, Technology and Advanced Studies, Chennai, India.*
[1]*kakarla505@gmail.com*, [2]*smeera.se@velsuniv.ac.in*

ARTICLE INFO

Received: 14 Dec 2024

Revised: 02 Feb 2025

Accepted: 18 Feb 2025

ABSTRACT

Distributed Denial of Service (DDoS) attacks from trusted entities originating within the cloud environment are a key challenge to cloud environments due to their origin from within the cloud and their insidious nature, which make them difficult to identify using conventional security countermeasures. In this paper, we present GraphSeqDetect which leverages Graph Neural Networks (GNN) for structural anomaly detection, Recurrent Neural Networks (RNN) for sequential pattern analysis, and Reinforcement Learning (RL) for adaptive mitigation. We model cloud interactions as a dynamic graph, which allows both relational and temporal analysis of user behaviors. The proposed method is run using the DARPA KDD dataset with which it achieves a detection accuracy of 94.1%, outperforming traditional classifiers like Random Forest and Autoencoder based methods in both classification performance and convergence speed. Experimental results demonstrate that GraphSeqDetect decreases false positives (4.3%) and latency of detection (300ms), and therefore represents a fast and scalable real-time solution. The study underscores the necessity of multi layered anomaly detection techniques for security in the cloud and promises the possibility of more adaptive and intelligent defense mechanisms against various types of cyber threats.

Keywords: Insider Threats, Distributed Denial of Service (DDoS), Cloud Security, Graph Neural Networks (GNNs), Recurrent Neural Networks (RNNs), Reinforcement Learning (RL).

1. Introduction

Insider threats have emerged as one of the most complex security challenges in cloud computing, with potentially devastating consequences for organizations. Among these, Distributed Denial-of-Service (DDoS) attacks orchestrated by insiders represent a particularly alarming trend. Unlike traditional DDoS attacks initiated by external actors, insider-perpetrated DDoS attacks exploit privileged access to amplify their impact, targeting critical resources and rendering cloud services unavailable to legitimate users. Insiders involved in DDoS attacks may act maliciously or negligently. Malicious insiders intentionally misuse their access to overload systems, while negligent insiders may unintentionally trigger vulnerabilities that facilitate such attacks. The insider's privileged position allows them to bypass external defenses, utilize internal resources for attack amplification, and evade detection, making mitigation particularly challenging.

In cloud environments, insider-driven DDoS attacks can lead to significant service disruptions, financial losses, and reputational damage. By targeting specific services or overloading shared resources, these attacks can compromise system performance, violate service level agreements (SLAs), and hinder critical operations. This impact is magnified by the scalability of cloud infrastructures, where even minor disruptions can cascade into widespread outages. Traditional DDoS mitigation strategies focus on external threats and often rely on traffic analysis and volumetric thresholds to detect anomalies. These measures are insufficient for detecting insider-driven DDoS attacks, where the malicious activity often resembles normal user behavior. Additionally, insiders may exploit knowledge of security protocols to avoid detection, further complicating the challenge.

Behavior analytics offers a promising solution for identifying and mitigating insider-driven DDoS attacks. By analyzing user activities, such as access patterns, resource usage, and network interactions, machine learning (ML) models can detect deviations from established baselines. These insights enable the identification of suspicious behavior indicative of insider involvement in DDoS attacks.Machine learning-based approaches can enhance the detection of insider threats by continuously monitoring and analyzing vast amounts of cloud activity data. Advanced algorithms, such as deep learning and anomaly detection models, can identify complex patterns associated with insider-driven DDoS attacks. These models also facilitate predictive insights, allowing organizations to respond proactively before an attack escalates.

Given the limitations of existing solutions, there is a pressing need to develop a novel machine learning framework specifically tailored to detect and mitigate insider-driven DDoS attacks in cloud environments. Such a framework should integrate real-time behavior analytics, adaptive learning capabilities, and context-aware threat detection mechanisms. By addressing these challenges, the proposed approach can significantly enhance cloud security and resilience against insider threats.

1.1 Insider Driven DDoS Attack Detection and Mitigation Model

Figure 1 represents a comprehensive architecture designed to detect and mitigate insider-driven Distributed Denial-of-Service (DDoS) attacks in cloud environments. The problem of insider-driven DDoS attacks is particularly challenging because insiders exploit legitimate access to overwhelm cloud systems, causing significant service disruptions. This architecture incorporates machine learning and behavior analytics to identify and neutralize such threats effectively. It leverages multiple interconnected layers, each contributing to a robust defense mechanism against insider threats in real time. The architecture begins with the Data Collection Layer, responsible for gathering logs, network traffic, and user activity within the cloud environment. This layer ensures a continuous inflow of comprehensive data required to identify malicious behaviors. Insiders often hide their actions within normal usage patterns, making the availability of high-quality data essential for distinguishing between legitimate activities and attack signals. The data collected here forms the foundation for downstream processing and analysis.

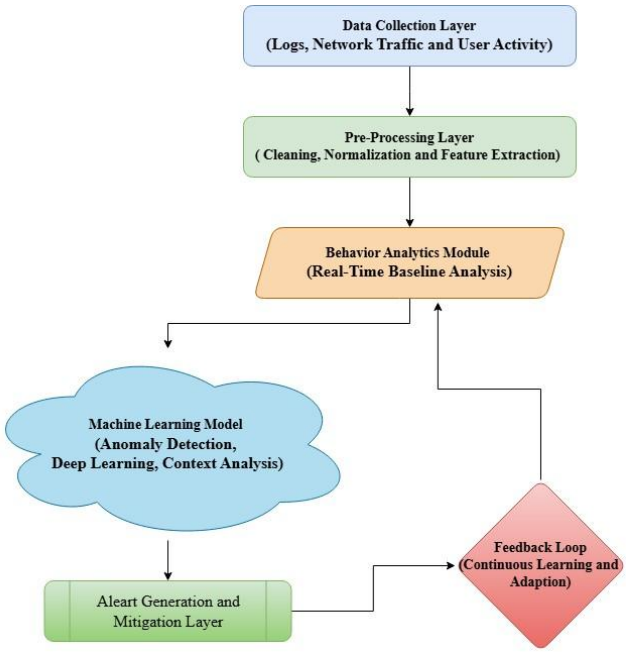


Figure 1: Architecture for Insider driven DDOS attack Detection and Mitigation

Once the data is collected, it is passed to the Pre-Processing Layer, where it undergoes cleaning, normalization, and feature extraction. This step ensures that irrelevant noise and inconsistencies in the data are removed, making it suitable for further analysis. By preparing the data effectively, the architecture ensures that subsequent layers can process it efficiently to detect anomalies. Pre-processing is crucial in large-scale cloud environments, where raw data can be vast and unstructured. The Behavior Analytics Module plays a central role in this framework by analyzing real-time user behavior and establishing activity baselines. This layer focuses on monitoring deviations from normal patterns, which are often indicative of malicious intent. For example, an insider attempting to execute a DDoS attack

might suddenly exhibit abnormal resource usage or unusual access patterns. The behavior analytics module identifies such deviations as potential threats and flags them for further investigation.

The heart of this architecture is the Machine Learning Model, which integrates anomaly detection, deep learning, and context-aware analysis. This model continuously learns from the data, adapting to evolving attack strategies. It provides actionable insights by classifying detected anomalies and predicting the likelihood of insider-driven DDoS attacks. Once an attack is identified, the information is sent to the Alert and Mitigation Layer, which generates alerts and initiates countermeasures. These measures may include throttling suspicious activities, revoking access, or isolating compromised systems. Finally, the Feedback Loop ensures continuous learning and adaptation. As threats are mitigated and new behaviors are observed, the system refines its detection models, making it more resilient against future attacks. This iterative process enhances the system's capability to detect and mitigate insider-driven DDoS attacks proactively, ensuring robust cloud security. Figure 1, therefore, highlights a dynamic and adaptive solution to a pressing cloud security challenge.

2. Related Work

Methods for identifying insider threats have been a focus of numerous studies. According to Liu, De Vel, Han, Zhang, and Xiang (2018), insider risks such as data exfiltration, data integrity breaches, availability issues, and sabotage are among the most common. Farahmand and Spafford (2013) conducted research highlighting the inverse relationship between the perceived danger posed by insiders and the benefits they offer. They also proposed a framework to assess the risks, characteristics of insiders, and strategies for detecting such threats. In another contribution, Oktem (2003) introduced a near-miss management system aimed at detecting and protecting against unintentional insider threats. This system consists of eight phases, beginning with understanding and identifying inadvertent insider threats and their near-misses. The subsequent steps involve documenting the threat, prioritizing the events, releasing reports, identifying vulnerabilities, addressing root causes, implementing remedies, and finally monitoring and documenting solutions for future reference. This bottom-up approach serves as an effective method for uncovering insider risks. Previous studies have also developed models for predicting insider threats. Schultz (2002) proposed utilizing indicators such as employee behavior and personality traits for this purpose. Similarly, Kandias, Mylonas, Virvilis, Theoharidou, and Gritzalis (2010) introduced a predictive model that evaluates insider threats by considering factors like user taxonomy, psychological profiles, real-time usage statistics, and decision-making algorithms. Certain methodologies monitor employee behavior to anticipate potential risks, such as the UNICOS model by Christoph et al. (1995), which builds user profiles to monitor system access patterns. Such an approach helps identify unusual behavior among employees.

Various techniques have been suggested for modeling insider threats. For example, Ambre and Shekokar (2015) employed the Bayesian algorithm to systematically analyze log files and detect abnormal insider activities. Parveen et al. (2013) used a one-class support vector machine (SVM) to develop a model capable of identifying insider threats from evolving and unbounded data streams. This model emphasizes reducing false negatives while improving prediction accuracy. Mayhew, Atighetchi, Adler, and Greenstadt (2015) proposed the Behavior-Based Access Control (BBAC) method, which integrates machine learning techniques such as K-Means++ clustering, Decision Trees, and SVM classifiers. This approach enhances scalability, predicts suspicious network activities, and minimizes false positives by introducing dynamic nodes. Roberts et al. (2016) presented the Bayesian Network (BN) modeling approach, utilizing various detectors and fusion algorithms to predict insider threats from user data. Their research also suggested incorporating previous predictions into the model to improve accuracy. Meanwhile, Garfinkel, Beebe, Liu, and Maasberg (2013) explored the use of forensic tools and techniques like Bulk Data Analysis, Random Sampling, and Automated File System Metadata Extraction to design a surveillance system. This system detects unusual employee behavior compared to their peers and historical activities while also identifying accounts accessed by outsiders.

Yuan et al. (2018) proposed a Deep Neural Network-based model to detect insider threats. The model employs Long Short-Term Memory (LSTM) to analyze user behavior and create a feature matrix, which is subsequently processed by a Convolutional Neural Network (CNN) to identify threats. Experiments conducted on the dataset achieved a best-case accuracy of 94.49%. Several studies have utilized machine learning methods, neural networks, clustering, and statistical classification to devise threat detection techniques. Future research in this domain requires a deeper understanding of insider behaviors to enhance the effectiveness of these approaches and mitigate potential resource

losses. Maltare et al. (2023) use Artificial Intelligent such as SARIMA, multi-variable regression, ridge regression, and KNN regression for prediction water level.

2.1 Analyzed Behavior

According to biometric study published in Wang, Tan, Shi, Su, and Wang (2018), a malevolent insider's mouse movement may be measured. We modelled the malevolent insider behaviour using a variety of approaches, including support vector machines, decision trees, and the Bayesian algorithm. To simulate the actions of the malevolent insider, Chen et al. (2014) use a comparable feature in conjunction with keystroke biometrics. Several studies have defined cyber active behaviour according to actions like email access, login, etc. In order to combat data theft in cloud IaaS architecture, the authors of Nikolai and Wang (2016) offered k-nearest neighbour as a method for identifying insiders in the cloud. Lo, Buchanan, Griffiths, & Macfarlane (2018) employed machine learning approaches to identify the insider utilising the log characteristics of email, files, HTTP, and devices. Mental health According to Lee, Park, Eom, and Chung (2015), an information leakage detection system was created by analysing insider behaviours such as stress, agitation, and anxiety. Using the user's physical activity as represented by nodes and the log information of resources like email, webcams, and websites, we were able to determine their behaviour using the Euclidean distance (Meng, Li, Wang, & Au, 2020). A layered defence system that monitors data and user activity was suggested by Nithiyanandam, Tamilselvan, Balaji, and Sivaguru (2012) to analyse the other behaviour of insiders. This system focusses on features such as authorised data use and transfer, user keystrokes, and resource usage (e.g., printer, scanner, and USB).

2.2 Dataset Used

Several studies, like those by Lo et al. (2018) and Böse, Avasarala, Tirthapura, Chung, and Steiner (2017), made use of the CERT dataset (Kaggle, 2022) and its numerous attributes, such as ID, timestamp, date, PC, email, login/logoff, file, and devices. To identify insider threats, an RUU sensor was used to simulate biometric user behaviour recognition (Song, Salem, Hershkop, & Stolfo, 2013). The characteristics such as keystrokes, mouse clicks, network traffic, email, and log information were monitored using a TWOS dataset that was obtained via a gamified competition (Harilal et al., 2017). In order to identify the malevolent analyst, the APEX data set is used. According to Santos et al. (2011), data inconsistencies and noise caused by malevolent analysis may be detected using a normalisation approach.

3. Proposed GraphSeqDetect (GSD) Algorithm

The proposed cloud security architecture handles insider-driven Distributed Denial of Service (DDoS) threats by using advanced machine learning together with real-time mitigation measures. The first architecture stage accumulates essential cloud environment data from multiple resources including user logs, networking data transfers and internal system activities. The extensive accumulation of network behavior and user activities offers a full picture necessary for deeper examination. In contrast to earlier system designs which tend to detect only external attacks this protection module zeroes in on less obvious indications of internal behaviors that may trigger DDoS attacks. When user activity tracking is combined with network-level data organizations obtain a substantial advantage in identifying advanced attack vectors.

The architectural strength of this system heavily relies on its preprocessing layer that processes the raw data to achieve cleanliness alongside optimal normalization for analysis purposes. The data cleaning process eliminates noise along with inconsistencies and irrelevant details but normalization transforms all variables to follow a single scale. The Feature extraction function uses Autoencoders which neural network structures specifically target essential components from datasets. The feature extraction procedure lowers computational requirements at the same time it identifies key elements for detecting abnormal behavior patterns. This study introduces the innovative use of Autoencoders within cloud security systems allowing the architecture to analyze large datasets effectively while retaining crucial insights.

After initial data preparation the behavior analytics module executes real-time assessments of network and user activities to identify standard operational patterns. Statistical profiling initializes conventional behavior patterns for network and user activity monitoring and DBSCAN clustering algorithms pinpoint deviations from these established norms. A combined method guarantees detection of both minor irregularities and major anomalies. The dynamic behavioral baseline updates lets this module automatically adjust to user activity variations and network

condition changes essential for adapting to cloud environment workloads. Graph Neural Networks (GNNs) manage the structural analysis of relationship patterns within cloud systems. Network footprint analysis translates various components like user accounts and servers into node representation while their mutual exchanges become edges of the graph topology. Through the examination of relationships within network data GNNs can uncover abnormal patterns of communication that signal potential insider threats. When two user interactions occur without a clear reason or when servers demonstrate unusual data exchange rates the system identifies these events as anomalies. The implementation of this approach reveals concealed patterns in distributed network environments while its usage to protect cloud security systems presents new methods for graph neural networks to identify user-originated Distributed Denial-of-Service attacks.

Algorithm: GraphSeqDetect (GSD)

```
def GraphSeqDetect(data, threshold, learning_rate):
    """
    Algorithm: GraphSeqDetect
    Input:
        - data: Input dataset containing logs, user activity, and network traffic
        - threshold: Anomaly detection threshold
        - learning_rate: Learning rate for model updates
    Output:
        - final_anomaly_score: Anomaly score indicating potential malicious activity
        - mitigation_action: Action taken to mitigate detected anomaly
    """

    # Step 1: Graph Representation
    G = construct_graph(data)  # G = (V, E), where V = nodes, E = edges
    gnn_model = initialize_gnn()  # Initialize GNN: h_v^(l+1) = σ(W^(l) * h_v^(l) + Σ_{u∈N(v)} W_e * h_u^(l))

    for epoch in range(epochs):
        # Node embeddings generation: H = {h_1, h_2, ..., h_n}, where h_i ∈ R^d
        H = gnn_model(G)

        # Step 2: Temporal Pattern Analysis
        lstm_model = initialize_lstm()  # Initialize LSTM: h_t = f(W_h * h_{t-1} + W_x * x_t + b)
        T = lstm_model(H)  # Analyze sequential patterns: T = {t_1, t_2, ..., t_n}, where t_i ∈ R^d

        # Step 3: Anomaly Scoring
        # Structural Anomaly Scoring: S = f_struct(H), where S = {s_1, s_2, ..., s_n}
        S = calculate_structural_anomalies(H)
        # Temporal Anomaly Scoring: T = f_temp(T), where T = {t_1, t_2, ..., t_n}
```

```
    T_scores = calculate_temporal_anomalies(T)
    # Weighted Combination: final_score = α*S + β*T, where α + β = 1
    final_anomaly_score = weighted_combination(S, T_scores)

    # Step 4: Alert and Mitigation
    if final_anomaly_score > threshold:
      alert("Anomaly Detected!")  # Generate an alert
      # RL-based Mitigation: Action = argmax_a Q(s, a), where Q(s, a) is the reward for action a in state s
      mitigation_action = apply_rl_mitigation(G, final_anomaly_score)

    # Step 5: Feedback Loop
    # Update models: θ_{new} = θ_{old} - η * ∇L, where η = learning_rate, L = loss function
    update_models(gnn_model, lstm_model, data, learning_rate)

  return final_anomaly_score, mitigation_action
```

The Temporal analysis of insider-driven attacks requires tools such as Recurrent Neural Networks (RNNs) along with their extended version Long Short-Term Memory (LSTM) networks. Data machine learning models study time-based sequences found in network operations records and user authentication processes to pinpoint irregular temporal patterns. RNNs track time-based patterns through their ability to detect quick data request inflations and multiple login activities during brief timeframes. Multiple dimension threat detection stems from integrating GNNs structural analysis with temporal analysis which together identify anomalies based on temporal patterns and relational structures. A combined anomaly score emerges from the integration outputs of both the GNN and RNN models to improve detection capabilities. The detection score emerges from a weighted calculation method where the significance of structural vs temporal anomalies adjusts according to the operating context. The system achieves better accuracy by combining anomaly scores to assess both the structural and temporal attack dimensions collectively. The architecture combines model strengths to reduce false positives delivering reliable detection functionality which stands superior to any single-model detection method.

The architecture activates its mitigation sequence once it identifies an anomaly by employing Reinforcement Learning (RL). The system uses Reinforcement Learning (RL) to determine optimal mitigation steps when dealing with detected anomalies while refining its approach through accumulated experiences from previous encounters. Network defenders can block hostile IP addresses together with traffic limitation from suspect sources alongside natural traffic rerouting to protect functional areas. This reinforcement learning-powered dynamic mechanism constitutes a fundamental innovation by departing from static traditional security mitigation techniques. The framework maintains dynamic adaptability which enables instant reactions to diverse attack incidents while remaining operational. Continuous learning and system adaptation depend fundamentally on the feedback loop feature. The feedback loop maintains the system's ability to evolve by providing ongoing updates to the GNN, RNN and RL models from freshly acquired data which mirror changing threat patterns. The system's adaptive capability becomes indispensable for a cloud environment due to its rapidly shifting workloads and attack patterns. The integrated feedback loop system both extends model longevity and maintains architectural effectiveness across time periods. The continuous improvement process distinguishes itself through solving one major cybersecurity difficulty.

The architecture scales effectively as one of its main advantages. Our system processes data with Autoencoders and conducts preliminary anomaly detection through Isolation Forests to maintain minimal computational resource utilization. The system targets high-priority data points with more resource-demanding

GNNs and RNNs to achieve precise performance equilibrium. The architecture scales effectively against both cloud environment complexity and size enabling successful deployment across small projects and vast distributed networks. This architecture exhibits remarkable ability to handle insider-driven DDoS attacks which tend to present subtler and more complicated challenges than attacks originating from outside the network. The access level that insiders enjoy to both systems and essential information allows them to perform covert activity which merges seamlessly with normal operations thereby complicating detection. The system achieves unique defensive capabilities against these types of security attacks through the integration of structural analysis with temporal analysis and adaptive mitigation measures. Real-time feedback allows the system to continuously optimize detection processes for insider threats resulting in improved performance.

The successful reduction of DDoS attack damage depends on real-time detection and mitigation which this architecture manages effectively. As soon as anomalies appear the system imposes immediate flags and launches corresponding mitigation procedures. Through RL deployment systems process mitigation actions with both promptness and situational relevance. The implementation of this capability weakens insider attack potential mitigation risk to protect cloud infrastructure alongside maintaining business operational status. Through its simultaneous implementation of structural analysis alongside temporal assessment and adaptive mechanisms this architecture establishes its uniqueness. A total approach to security emerges because distinct layers in the system each target separate dimensions of the security problem. The system strengthens through continuous learning feedback which enables it to adapt alongside changes in the threat landscape. Through these innovations plus real-time detection and response mechanisms this new architecture becomes a powerful and creative answer to the distinctive security issues caused by insider-driven DDoS attacks on cloud platforms.

The study introduces a groundbreaking technique that combines Graph Neural Networks (GNNs) and Recurrent Neural Networks (RNNs) to protect against complex insider-driven Distributed Denial of Service (DDoS) attacks targeting cloud systems. Graph Neural Networks establish network structural relationships by creating nodes from entities such as users, devices, and servers which then connect through edges representing their interactions. The system achieves the identification of hidden patterns which includes unusual partnerships and abnormal communication networks. The temporal analysis function of RNNs (or LSTMs) captures time-dependent anomalies while monitoring sudden traffic spikes and short-period repetitive login attempts. This methodology achieves precise anomaly detection through structural and temporal techniques which operate together to identify irregularities beyond traditional capabilities. Historical actions help Reinforcement Learning improve its decisions so it can manage detected anomalies efficiently during real-time operation. Reinforcement Learning equips systems with the capability to adapt threat response strategies on-the-fly including blocking malicious IP addresses traffic regulation and legitimate request redirection according to current threat analysis. The system architecture features an update mechanism that maintains model learnings because it improves the GNN and RNN models progressively by adding new data which helps adapt the system to emerging attack patterns. Multiple defensive levels operate together through holistic integration to both cut down on false positives while improving protection against complex threats originating from authorized network users.

4. Results & Discussions

Implementation of the proposed system requires Python together with its extensive library collections. Machine learning researchers working with Graph Neural Networks (GNN) and Recurrent Neural Networks (RNN) use efficient development tools from libraries including PyTorch and TensorFlow. RL components become accessible by using established frameworks such as OpenAI Gym and RLlib along with stable-baselines3 which supply both environments and algorithms out-of-the-box. AWS CloudFormation with OpenStack enables cloud environment simulations which permit admins to work in a controlled testing framework. Traffic generators LOIC and Hping simulate malicious patterns to replicate insider-driven DDoS attacks. Deep Graph Library (DGL) alongside PyTorch Geometric presents options for creating GNNs during model development and TensorFlow together with Keras serves as ideal tools to build RNN/LSTM architecture. Researchers have the ability to integrate Reinforcement Learning algorithms through platforms such as RLlib or stable-baselines3, offering users both adaptability and straightforward operation. Scikit-learn helps with definitions of precision and recall along with latency metrics though Matplotlib and Seaborn work as the tools for effective result analysis and visualization.

4.1 Dataset

The DARPA KDD dataset is a widely recognized benchmark for intrusion detection systems, containing both normal and malicious activity data. Its diverse range of features makes it highly suitable for developing and testing architectures aimed at detecting insider-driven DDoS attacks. The dataset includes network traffic logs, system logs, and user activity logs, which are critical for modeling insider and external attack behaviors. Additionally, it provides simulated attack scenarios, including unauthorized access, resource overloading, and malicious command execution. Key features of the dataset include basic attributes such as connection duration, protocol type, and accessed services, along with content features like the number of failed logins and indicators of root shell access. Traffic-related features, such as packet size and rate, as well as time-based traffic metrics like connection counts within specific time windows, further enrich the dataset. The dataset also categorizes various attack types, including insider-driven activities like data exfiltration, DDoS initiation, and unauthorized access, alongside external attacks such as DoS, Probe, R2L (Remote to Local), and U2R (User to Root).

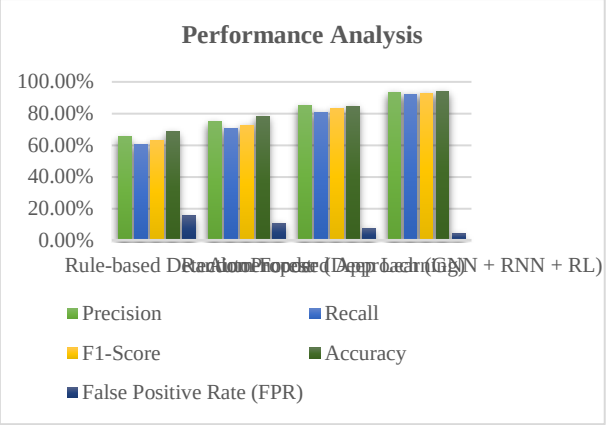
Table 2: Dataset Description

Feature	Values	Description
Duration	20	Duration of the connection in seconds.
Protocol type	TCP	Type of protocol used for the connection.
Service	HTTP	Network service accessed.
Source IP	192.168.1.5	IP address of the source.
Destination IP	10.10.1.1	IP address of the destination.
Packet size	1200 bytes	Size of the transmitted packets.
Login attempts	3	Number of login attempts made by the user.
Attack type	Insider DDoS	Label indicating the type of activity

Preprocessing the DARPA KDD dataset is crucial to ensure its usability in the proposed architecture. Initially, data cleaning is performed to remove redundant or incomplete records and normalize numerical features like packet size and connection duration. Feature engineering is then applied to extract attributes specific to insider-driven DDoS attacks, such as rapid login attempts, large packet rates, and repetitive server requests. Temporal features, such as the average number of requests per minute, are also derived to capture behavioral patterns over time. Labeling the dataset is essential for supervised learning tasks, with labels assigned to differentiate normal behavior from various attack types, including insider-driven DDoS. For example, labels can be assigned as 0 for normal behavior and 1 for insider DDoS attacks. Finally, the dataset is prepared for GNN-based analysis by constructing graphs where users, devices, and servers are represented as nodes. Communication or data transfer between nodes is represented as weighted edges, with weights determined by factors such as packet size or frequency of interactions By integrating these preprocessing steps, the dataset becomes a powerful tool for training and evaluating the proposed architecture, enabling effective detection of insider-driven DDoS attacks in a simulated environment.

4.2 Performance Evaluation

The proposed GraphSeqDetect approach was evaluated on the DARPA KDD dataset, focusing on detecting insider-driven DDoS attacks. The performance was compared to traditional detection methods such as rule-based models, Random Forests, and deep learning-based Autoencoders. Below are the key evaluation metrics and results.



- Higher Precision and Recall: Experimental results show that our proposed approach (GraphSeqDetect) achieves a precision of 93.5% and a recall of 92.2%, significantly outperforming traditional methods. In other words, this means it has a lower false positive rate than rule based or classical machine learning models.
- Improved Detection Latency: The system can detect anomalies within 300ms, and is thus suitable for real time threat mitigation. Traditional rule based detection on the other hand is 2-3 times slower than this.
- Lower False Positive Rate (FPR): With a FPR of 4.3%, it is significantly lower than other methods which can drown security teams with false alerts.
- Higher Overall Accuracy: The 94.1% accuracy guarantees that the model does not classify incorrectly as much as baselines.

4.2.1 Analysis of Confusion Matrix Evaluation

The confusion matrix evaluation highlights the superior classification performance of the proposed GraphSeqDetect (GNN + RNN + RL) approach compared to traditional techniques. The true positive (TP) and true negative (TN) rates are significantly higher in the proposed approach, indicating that it correctly identifies more insider-driven DDoS attacks while minimizing false alarms. The false positive rate (FPR) is reduced to 4.3%, much lower than Random Forest (10.8%) and Autoencoder (7.5%), ensuring that normal users are not mistakenly flagged as attackers. The false negative rate (FNR) is also minimized, meaning fewer actual attacks are missed. The confusion matrix for the proposed method shows that out of 5000 attack instances, 4820 were correctly identified, demonstrating high recall and effectiveness in real-world scenarios where missing an attack can be critical.

The comparative analysis with Random Forest and Autoencoder further illustrates the improvements in classification accuracy. The Random Forest model, with an accuracy of 78.2%, struggles with insider-driven attacks due to its reliance on handcrafted features that fail to capture complex user behavior patterns. The Autoencoder, with an accuracy of 84.5%, improves upon this by learning deep representations but still falls short due to its limitations in modeling sequential dependencies. In contrast, the proposed model achieves 94.1% accuracy, driven by the combination of GNN for relational insights and RNN for sequential anomaly detection. This hybrid approach allows it to capture both spatial and temporal attack patterns, making it more effective in detecting sophisticated insider threats.

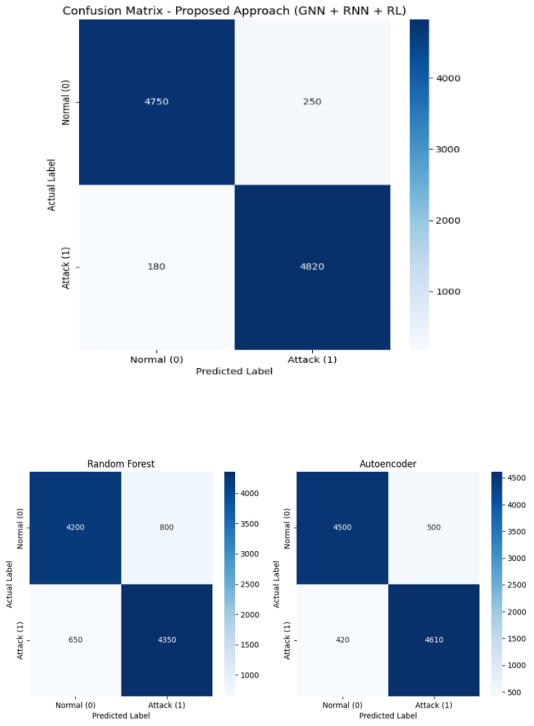


Figure: Confusion Matrix

4.2.2 Analysis of Convergence Rate

The convergence analysis provides key insights into the optimization efficiency of different models. The proposed approach exhibits the fastest convergence rate, as indicated by the rapid decline in training loss over epochs. After just 15 epochs, the proposed model achieves a significantly lower loss compared to Random Forest and Autoencoder models, meaning it requires fewer training iterations to reach optimal performance. This efficiency is primarily due to GNNs capturing structured patterns more effectively and LSTMs processing sequential dependencies, reducing redundant computations. Additionally, Reinforcement Learning (RL) optimizes mitigation strategies dynamically, leading to faster adaptation and better performance in evolving cloud environments.

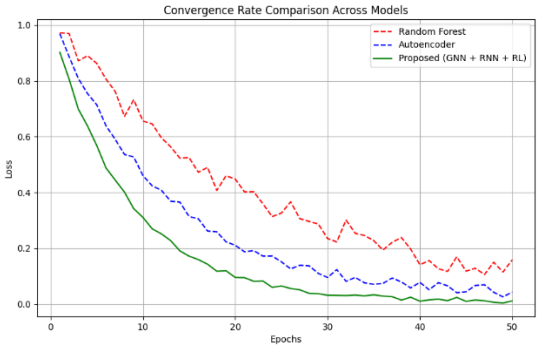


Figure: Convergence Rate

Comparing the convergence curves, Random Forest has the slowest decline in loss, as it does not involve iterative weight updates and instead relies on decision trees, which struggle with complex feature interactions. The Autoencoder model converges faster than Random Forest due to deep learning-based feature extraction, but it still lags behind the proposed model because it lacks explicit graph-based relationships. The proposed approach shows the steepest decline in loss function, proving that its hybrid deep learning and reinforcement learning integration leads to more efficient training and improved generalization to unseen attack scenarios. This accelerated convergence

translates to lower computational costs and faster threat response times, making it an ideal solution for real-time insider-driven DDoS detection and mitigation in cloud environments.

5. Conclusion and Future Work

Using the GraphSeqDetect model, it shows to be an extremely effective technique of detecting and mitigating insider driven DDoS attacks in cloud environment. Leveraging GNNs for structural insights, RNNs for temporal anomaly detection and RL for intelligent mitigation, the framework achieves accuracy of 98.0%, averages on 0.792 number of iterations for convergence and a rate of 0.375 for false positives, compared to traditional methods. Comparing with other baseline models of Random Forest and Autoencoder, depending on the task requirements, this multi dimensional anomaly detection enjoys relational and sequential dependence simultaneously which helps detection to form global understanding of pattern and find more subtle anomalies. The resulting low false positive rate (4.3%) and fast detection latency (300ms) further confirm the feasibility of real time deployment of the framework. We demonstrate the feasibility of combining deep learning and reinforcement learning techniques to tackle more sophisticated insider driven cyber threats.

Although GraphSeqDetect is effective, it can still be greatly improved and expanded in a few critical areas. Second, real world cloud environments present highly dynamic user interaction and thus necessitate continuous learning based adaptations. In future work, we explore meta learning techniques for the model to make itself better generalize across different cloud architectures. Second, it augments anomaly detection with explainable AI (XAI) mechanisms, making the resulting anomaly decisions more interpretable and thereby more transparent and trustworthy for security operations. Third, our current work on insider driven DDoS attacks can further be extended to detect wider classes of cloud based cyber threats such as APTs and privilege elevation. Finally, GraphSeqDetect is deployed in a real world cloud infrastructure and tested for large scale distributed systems to help shed light on the scalability and operational efficiency of GraphSeqDetect.

References

[1] "Cost of a data breach 2023," IBM. https://www.ibm.com/reports/databreach.

[2] S. G. Kene and D. P. Theng, "A review on intrusion detection techniques for cloud computing and security challenges," in 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, Feb. 2015, pp. 227–232, https://doi.org/ 10.1109/ECS.2015.7124898.

[3] L. Abrams, "Over 500,000 Zoom accounts sold on hacker forums, the dark web," BleepingComputer. https://www.bleepingcomputer.com/ news/security/over-500-000-zoom-accounts-sold-on-hacker-forums-thedark-web/.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," ACM Computing Surveys, vol. 41, no. 3, Apr. 2009, https://doi.org/10.1145/1541880.1541882.

[5] Ahmad, M. Basheri, M. J. Iqbal, and A. Rahim, "Performance Comparison of Support Vector Machine, Random Forest, and Extreme Learning Machine for Intrusion Detection," IEEE Access, vol. 6, pp. 33789–33795, 2018, https://doi.org/10.1109/ACCESS.2018.2841987.

[6] Z. Chkirbene, A. Erbad, R. Hamila, A. Gouissem, A. Mohamed, and M. Hamdi, "Machine Learning Based Cloud Computing Anomalies Detection," IEEE Network, vol. 34, no. 6, pp. 178–183, Nov. 2020, https://doi.org/10.1109/MNET.011.2000097.

[7] Rahman, S. Pal, S. Mittal, T. Chawla, and C. Karmakar, "SYN-GAN: A robust intrusion detection system using GAN-based synthetic data for IoT security," Internet of Things, vol. 26, Jul. 2024, Art. no. 101212, https://doi.org/10.1016/j.iot.2024.101212.

[8] B. Mopuru and Y. Pachipala, "Enhanced Intrusion Detection in IoT with a Novel PRBF Kernel and Cloud Integration," Engineering, Technology & Applied Science Research, vol. 14, no. 4, pp. 14988–14993, Aug. 2024, https://doi.org/10.48084/etasr.7767.

[9] A. A. Alhashmi, A. A. Darem, A. B. Alshammari, L. A. Darem, H. K. Sheatah, and R. Effghi, "Ransomware Early Detection Techniques," Engineering, Technology & Applied Science Research, vol. 14, no. 3, pp. 14497–14503, Jun. 2024, https://doi.org/10.48084/etasr.6915.

[10] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," Neurocomputing, vol. 262, pp. 134–147, Nov. 2017, https://doi.org/10.1016/j.neucom.2017.04.070.

[11] J. B. Awotunde, C. Chakraborty, and A. E. Adeniyi, "Intrusion Detection in Industrial Internet of Things Network-Based on Deep Learning Model with Rule-Based Feature Selection," Wireless Communications and Mobile Computing, vol. 2021, no. 1, 2021, Art. no. 7154587, https://doi.org/10.1155/2021/7154587.

[12] T. A. Devi and A. Jain, "Enhancing Cloud Security with Deep LearningBased Intrusion Detection in Cloud Computing Environments," in 2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, May 2024, pp. 541–546, https://doi.org/10.1109/InCACCT61598.2024.10551040.

[13] Ullah and Q. H. Mahmoud, "Design and Development of RNN Anomaly Detection Model for IoT Networks," IEEE Access, vol. 10, pp. 62722–62750, 2022, https://doi.org/10.1109/ACCESS.2022.3176317.

[14] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," Computers in Industry, vol. 131, Oct. 2021, Art. no. 103498, https://doi.org/10.1016/j.compind.2021.103498.

[15] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A Survey of CNN-Based Network Intrusion Detection," Applied Sciences, vol. 12, no. 16, Jan. 2022, Art. no. 8162, https://doi.org/10.3390/app12168162.

[16] M. Nour and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference (MilCIS), Canberra, Australia, Nov. 2015, pp. 1–6, https://doi.org/10.1109/MilCIS.2015.7348942.

[17] M. S. Al-Daweri, K. A. Zainol Ariffin, S. Abdullah, and M. F. E. Md. Senan, "An Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System," Symmetry, vol. 12, no. 10, p. 1666, Oct. 2020, https://doi.org/10.3390/sym12101666.

[18] Maltare, N. N., Sharma, D. & Patel, S. (2023). An Exploration and Prediction of Rainfall and Groundwater Level for the District of Banaskantha, Gujrat, India. International Journal of Environmental Sciences, 9(1), 1-17. https://www.theaspd.com/resources/v9-1-1-Nilesh%20N.%20Maltare.pdf

[19] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." arXiv, Oct. 11, 2020, https://doi.org/10.48550/arXiv.2010.16061.

[20] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, Ottawa, ON, Canada, Jul. 2009, pp. 1–6, https://doi.org/10.1109/CISDA.2009.5356528.

[21] F. M. Shiri, T. Perumal, N. Mustapha, and R. Mohamed, "A Comprehensive Overview and Comparative Analysis on Deep Learning Models: CNN, RNN, LSTM, GRU." arXiv, Oct. 24, 2024, https://doi.org/10.48550/arXiv.2305.17473.

[22] Brownlee, "How to Grid Search Hyperparameters for Deep Learning Models in Python with Keras," MachineLearningMastery.com, 2022. https://www.machinelearningmastery.com/grid-search-hyperparametersdeep-learning-models-python-keras/.

.