Journal of Future Artificial Intelligence and Technologies

E-ISSN: 3048-3719

Research Article

Proactive Insider Threat Detection Framework: An Explainable AI and Behavioral Analytics-Driven Approach

Oladapo Adeduro [1], Olabisi Josh-Falade [2,*], and Ayobami Mesioye [3]

[1] Department of Cybersecurity, McPherson University, Seriki-Sotayo110001, Ogun State, Nigeria; e-mail : adedurooo@mcu.edu.ng
[2] Department of Information Technology, McPherson University, Seriki-Sotayo110001, Ogun State, Nigeria; e-mail : faladeoa@mcu.edu.ng
[3] Department of Cybersecurity, McPherson University, Seriki-Sotayo110001, Ogun State, Nigeria; e-mail : mesioyeae@mcu.edu.ng
* Corresponding Author : Olabisi Josh-Falade

Abstract: Insider threats remain a persistent challenge in organizational cybersecurity, requiring advanced AI-based behavioral analytics to detect subtle and long-term malicious activity. However, the practical deployment of such systems is constrained by two critical issues: strict data protection regulations that limit centralized access to sensitive user logs, and the lack of transparency in deep learning models, which reduces analyst trust and operational adoption. This study proposes a unified insider threat detection framework that integrates Federated Learning (FL), Differential Privacy (DP), and Explainable AI (XAI) to address these challenges simultaneously. An LSTM-based sequential model is trained in a federated manner, ensuring that user data remains local while privacy is enforced through input perturbation using the Laplace mechanism. Model predictions are interpreted using SHAP and LIME to provide actionable explanations for security analysts. To evaluate robustness and generalizability, the framework is validated across two fundamentally different environments: the synthetic, user-centric CERT dataset representing traditional enterprise systems, and the real-world, cloud-native BETH dataset capturing low-level system behavior from live attacks. Experimental results show that the proposed framework achieves competitive detection performance, with F1-scores of 0.88 on CERT and 0.86 on BETH, while providing formal privacy guarantees. Qualitative evaluation further indicates that the XAI layer improves clarity, actionability, and trust in model outputs. These findings demonstrate that accurate detection, privacy preservation, and explainability can be jointly achieved, enabling the practical deployment of trustworthy AI for insider threat detection in modern IT infrastructures.

Keywords: Behavioral Analytics; Cybersecurity Analytics; Differential Privacy; Explainable Artificial Intelligence; Federated Learning; Insider Threat Detection; LSTM Networks; Privacy-Preserving Machine Learning.

Received: December, 6th 2025
Revised: January, 28th 2026
Accepted: January, 29th 2026
Published: February, 4th 2026
Curr. Ver.: February, 4th 2026

1. Introduction

The global increase in remote work and cloud adoption by contemporary organizations has expanded the digital threat landscape, prompting significant investments in fortifying infrastructure against external adversaries [1]. However, an equally—if not more—deceptive challenge persists from within: insider threats. An insider is typically defined as a current or former employee, contractor, or business partner with legitimate, authorized access to an organization's network or data, who is therefore capable of inflicting severe damage [2]. Whether driven by malicious intent, such as espionage or sabotage, or arising from negligence, insider actions can result in devastating data breaches, substantial financial losses, and irreparable reputational harm [3]. The trust and privileged access granted to insiders fundamentally undermine traditional security mechanisms, such as firewalls and intrusion detection

systems—which are primarily designed to protect the boundary between external and internal networks, rendering them inadequate for addressing insider threats [4].

To address this critical security gap, advanced analytical approaches—most notably User and Entity Behavior Analytics (UEBA)—have been introduced. By leveraging Artificial Intelligence (AI) and Machine Learning (ML), UEBA systems continuously monitor user and entity activities to establish baselines of normal behavior and identify deviations that may signal emerging threats [5]. These systems analyze vast streams of sensitive employee data, including login patterns, file access events, email communications, and application usage, to detect subtle anomalies indicative of malicious or negligent behavior. Despite their effectiveness, centralized UEBA systems face several inherent challenges, including data privacy and regulatory compliance risks, scalability and performance bottlenecks, and data heterogeneity, which collectively contribute to high false-positive rates [6]. In the context of insider threats, centralized architectures further expose systems to evasion, as insiders often possess legitimate credentials and detailed knowledge of internal security policies, enabling them to bypass centralized detection mechanisms [7]. These limitations have led to growing advocacy for decentralized ML approaches, particularly Federated Learning (FL), in which models are trained on localized data and only parameter updates are shared with a global model [8]. Among deep learning techniques, Long Short-Term Memory (LSTM) networks have demonstrated strong potential in this domain due to their ability to learn complex sequential patterns from time-series data, making them well-suited for proactive insider threat detection [9]. The capacity of LSTMs to capture long-term temporal dependencies is especially important for accurately profiling malicious behavior, which often unfolds across multiple, non-consecutive steps. Although Transformer-based models have gained popularity for sequence modeling tasks, this study adopts LSTM architectures due to their lower computational overhead, a critical requirement for client-side training in FL environments. In this context, a client refers to a network-connected device such as a mobile phone, tablet, router, or workstation.

Despite the efficiency of these advanced AI-driven systems, their deployment introduces a dual dilemma that negatively affects organizational trust and operational effectiveness: the black-box problem and the privacy paradox. First, the algorithmic complexity that enables deep learning models to achieve high detection accuracy also renders their decisions opaque. Models may generate high-risk alerts without providing clear, human-understandable justifications. For Security Operations Center (SOC) analysts, such unexplainable alerts are not actionable and are often perceived as noise, leading to alert fatigue, delayed incident response, and an increased risk that genuine threats are overlooked amid numerous false positives.

Second, UEBA systems rely heavily on highly sensitive individual data, such as system access logs, usernames, IP addresses, and activity records [10]. To construct accurate behavioral profiles, these systems must process detailed digital representations of employees' daily activities. Centralizing such information creates a substantial privacy risk, as privileged insiders or malicious administrators may gain access to the complete digital footprint of all employees [11]. This practice exposes organizations to severe legal and financial penalties under data protection regulations such as the General Data Protection Regulation (GDPR) and the Nigeria Data Protection Regulation (NDPR), while also fostering a culture of mistrust that can negatively affect employee morale and productivity [12]. Organizations are therefore confronted with a fundamental dilemma: they must monitor internal behavior to ensure security while simultaneously safeguarding the privacy of the individuals being monitored. Traditional data anonymization techniques are increasingly insufficient in this context, as they remain vulnerable to correlation attacks that enable re-identification through auxiliary information, necessitating the adoption of mathematically rigorous privacy-preserving mechanisms that combine FL and Differential Privacy.

In response to this dual challenge, this paper proposes and evaluates a novel framework that integrates three complementary technological pillars to deliver a system that is simultaneously effective, privacy-preserving, and transparent. These pillars are not competing objectives but interrelated components of a trustworthy AI system for organizational security. Specifically, the proposed framework incorporates:

- Federated Learning (FL): a decentralized training paradigm in which models learn from local data without transferring raw, sensitive information to a central server.
- Differential Privacy (DP): a formal privacy mechanism that injects calibrated statistical noise during training, ensuring that individual user contributions cannot be inferred from model outputs.

- Explainable AI (XAI): model-agnostic interpretation techniques that provide transparent, evidence-based explanations for each alert generated by the system.

Furthermore, recognizing that the robustness and adaptability of a security framework can only be demonstrated across diverse operational environments, the proposed approach is evaluated using two distinct datasets: the CERT Insider Threat Dataset and the BETH dataset. CERT is a well-established synthetic benchmark that simulates a traditional corporate IT environment, while BETH represents a complex, modern dataset comprising real attack traffic and kernel-level logs collected from cloud-native infrastructure. By comparing performance across these two settings, this study demonstrates the framework's ability to deliver privacy-preserving and explainable security analytics in both conventional enterprise systems and next-generation computing environments.

The remainder of this paper is structured as follows. Section 2 reviews related work on insider threat detection, privacy-preserving machine learning (PPML), and explainable AI. Section 3 describes the proposed methodology and system architecture. Section 4 presents and discusses experimental results from both datasets. Finally, Section 5 concludes the study and outlines directions for future research.

2. Related Works

The development of a privacy-preserving and explainable insider threat detection system lies at the intersection of three major research domains: advanced behavioral analytics for threat detection, PPML, and XAI. This section reviews recent advances in these areas and identifies key limitations that motivate the proposed framework.

2.1. Deep Learning Models for Insider Threat Detection

Early approaches to insider threat detection primarily relied on signature-based and rule-based systems, often deployed within Security Information and Event Management (SIEM) platforms. These systems compare observed user activities against predefined policies or known malicious behavioral patterns [13]. Although such approaches are relatively simple and interpretable, their effectiveness is limited by their static nature. They are inherently incapable of adapting to novel, zero-day attacks or complex threat scenarios that deviate from established rules, frequently resulting in high false-positive and false-negative rates.

To overcome these limitations, researchers increasingly turned to ML techniques to develop more adaptive and data-driven detection systems. Early ML-based approaches employed classical algorithms such as Support Vector Machines (SVMs), Bayesian networks, and Random Forests to classify user behavior using statistical features extracted from audit logs and activity traces [14]. In parallel, Hidden Markov Models (HMMs) were widely adopted to model sequences of user actions, providing probabilistic measures of deviation from normal behavior [15]. While these methods represented a significant advancement over rule-based systems, they often struggled to capture long-term dependencies and subtle contextual patterns inherent in human behavior.

The introduction of Recurrent Neural Networks (RNNs), and in particular LSTM networks, marked a paradigm shift in insider threat detection research. LSTMs are explicitly designed to learn from sequential data, making them well suited for modeling temporal user behavior over extended periods. Numerous studies have demonstrated the superiority of LSTM-based models in detecting insider threats by analyzing sequences of system calls, login events, and file access patterns [16], [17]. By learning typical behavioral sequences, these models are capable of identifying deviations that signal potential malicious or negligent activity.

Subsequent research has further explored the effectiveness of deep learning models for analyzing behavioral features in insider threat scenarios [6]. In particular, modern deep learning and natural language processing (NLP) approaches applied to the CERT dataset have reinforced the value of proactive, data-driven threat detection strategies [5]. Variants such as Bi-directional LSTM (Bi-LSTM) networks have also been employed for feature extraction, demonstrating improved performance over traditional methods by capturing both past and future contextual information [18]. More recently, the Deep Synthesis-based Insider Intrusion Detection (DS-IID) model has been proposed to address emerging threats posed by generative AI, achieving 97% accuracy on the CERT dataset and underscoring the need for insider threat detection models to evolve alongside the threat landscape [19].

Beyond sequence-based models, Graph Neural Networks (GNNs) have emerged as a promising alternative for insider threat detection. GNNs model the complex relational structure between users, devices, processes, and data objects, enabling the extraction of contextual insights that are difficult to capture with purely sequential models. By leveraging relational dependencies between events, GNN-based approaches have been shown to reduce false positives and improve detection performance in complex environments [20].

2.2. Privacy-Preserving Machine Learning (PPML) in Security

The intensive data requirements of behavioral analytics systems have raised significant privacy concerns, motivating the development of PPML techniques. In security-sensitive domains such as insider threat detection, two approaches have emerged as particularly prominent, i.e., FL and DP.

2.2.1. Federated Learning

Federated Learning (FL) is a decentralized ML paradigm that enables collaborative model training across distributed data sources without transferring raw data to a central server [21]. This property makes FL especially suitable for scenarios involving sensitive information, such as employee activity logs or personal user data [22]. In FL, multiple participating entities jointly train a shared global model while retaining their local datasets, thereby reducing the risk of centralized data exposure.

Prior studies have categorized FL architectures based on data distribution into Horizontal FL (HFL), Vertical FL (VFL), and Federated Transfer Learning (FTL) [23]. HFL, which involves datasets with overlapping features but different users, has been the most widely applied variant in cybersecurity contexts. Its adoption has been shown to expand the effective training sample space and improve model generalization [24]. From a regulatory standpoint, FL has also been discussed as a viable mechanism for supporting compliance with data protection frameworks such as the General Data Protection Regulation (GDPR) in collaborative security systems [25]. However, despite these advantages, FL introduces new attack surfaces, including data poisoning and model reconstruction attacks, which necessitate additional safeguards and robust defense mechanisms [26].

2.2.2. Differential Privacy

Differential Privacy (DP) provides a formal, mathematical guarantee of privacy by injecting calibrated statistical noise into data or algorithmic outputs [27]. This mechanism ensures that the contribution of any single individual cannot be inferred from the trained model or its predictions. In the context of insider threat detection, DP is particularly relevant because training data often contain highly sensitive employee information, including system access records, communication metadata, and database interaction logs.

Existing research indicates that DP is most effective when combined with other privacy-preserving techniques rather than applied in isolation [28]. Such integration strengthens privacy protection while maintaining acceptable utility, especially in high-risk domains where regulatory compliance and ethical considerations are critical.

2.3. Explainable AI (XAI) in Cybersecurity

Explainable AI (XAI) has emerged as a key research direction aimed at improving the transparency and trustworthiness of AI-driven decision-making systems [29]. Model-agnostic techniques such as LIME [16] [30] and SHAP [31] [30] are widely adopted to explain predictions generated by complex deep learning models without requiring access to their internal structure.

In cybersecurity applications, XAI has been shown to enhance analyst trust and accountability by clarifying how models identify and prioritize threats. Within the specific context of insider threat detection, XAI enables actionable response strategies by revealing the behavioral factors that contribute to risk assessments [32]. This transparency supports effective human–AI collaboration and helps mitigate alert fatigue by distinguishing meaningful alerts from noise [33]. More recent studies have demonstrated the feasibility of integrating XAI with privacy-preserving approaches, including the combination of FL and XAI for intrusion detection, indicating that security, privacy, and interpretability can be jointly achieved within a unified framework [1].

2.4. Critical Analysis and Gap Summary

A review of existing literature reveals that, despite notable progress, research at the intersection of FL, DP, and XAI for insider threat detection remains limited. Table 1 presents a comparative summary of representative studies and contrasts them with the proposed framework in terms of dataset usage, privacy mechanisms, explainability, and validation environments.

Table 1. Comparison of the proposed framework with related works.

Reference	Dataset Used	Privacy Technique	XAI	Validation Environment
Nasir et al. [9]	CERT (Synthetic)	None	No	Centralized DL
Mothukuri et al. [24]	IoT / Network Traffic	FL	No	Distributed / Edge
Karn et al. [30]	Syscalls Dataset (Private)	None	SHAP / LIME	Centralized
Fatema et al. [34]	Network Traffic	FL	SHAP	Simulated FL
Proposed Work	CERT (Synthetic), BETH (Real-World Cloud)	FL + DP (Laplace)	SHAP & LIME	Multi-environment (Corporate & Cloud)

As summarized in Table 1, most existing studies address only one or two dimensions of the insider threat detection problem. Many works prioritize detection accuracy while overlooking privacy implications [6], propose privacy-preserving architectures without providing transparent and actionable explanations [24], or apply XAI techniques exclusively to centralized, non-private datasets [32]. Furthermore, a significant portion of prior research validates proposed methods on a single dataset type—often the synthetic CERT dataset—leaving their generalizability to modern, cloud-native environments largely unproven.

These limitations highlight a clear research gap for a holistic framework that simultaneously delivers accurate detection, privacy by design through DP, and meaningful explainability via XAI, while also demonstrating robustness across diverse operational environments. The proposed framework is explicitly designed to address this gap by unifying FL, DP, and XAI and evaluating their combined effectiveness in both traditional corporate and cloud-native settings.

3. Proposed Method

3.1. Overall Proposed Framework

Figure 1 illustrates the overall architecture of the proposed privacy-preserving and explainable insider threat detection framework. The framework is designed as a layered pipeline that processes raw activity logs into actionable, privacy-aware threat intelligence through four tightly coupled components. The Data Ingestion and Integration Layer serve as the system's entry point. This layer collects heterogeneous data streams generated by organizational activities, including application logs, physical access records, VPN connections, and database queries. Since these data sources differ in format, granularity, and semantics, the ingestion layer performs normalization and integration to construct a unified representation of user and entity activities. This unified stream provides a consistent behavioral view that can be consumed by downstream analytics components while preserving the temporal ordering of events.

The Behavioral Analytics and Profiling Layer is responsible for learning normal behavioral patterns from the integrated data stream. At this stage, user and entity activities are transformed into structured behavioral features and modeled as temporal sequences. The system continuously updates behavioral profiles to reflect evolving work patterns, enabling adaptive learning rather than static rule-based detection. This continuous learning mechanism allows the framework to distinguish between legitimate behavioral drift and suspicious deviations over time.

The Proactive Threat Assessment Layer constitutes the core detection engine of the framework. Using the learned behavioral profiles as a baseline, this layer identifies anomalous activity sequences that deviate from established norms. Each detected deviation is assigned a

risk score that reflects the severity and likelihood of malicious intent. This risk quantification enables the system to prioritize alerts and supports proactive mitigation before an insider attack fully materializes.

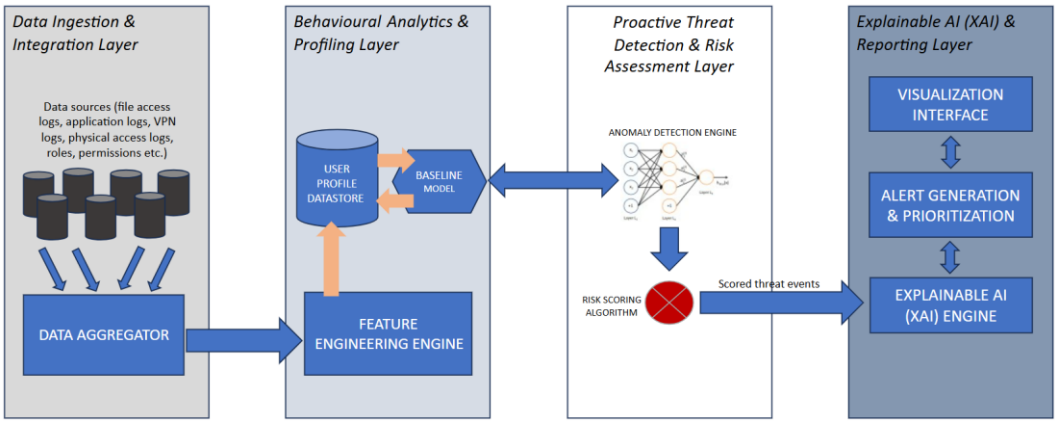

Figure 1. System architecture of the proposed FL–DP–XAI-based insider threat detection framework.

Finally, the XAI and Reporting Layer translates model outputs into transparent and actionable insights for human analysts. Instead of producing opaque alerts, this layer applies XAI techniques to identify the key behavioral factors contributing to each detected anomaly. Explanations are presented through an interface or dashboard, enabling Security Operations Center (SOC) analysts to understand, validate, and respond to alerts with greater confidence and reduced cognitive burden.

3.2. Methodology

This study adopts a structured methodology to design, train, and evaluate an insider threat detection system that is private-by-design and inherently explainable. As illustrated in Figure 2, the methodology follows an end-to-end pipeline consisting of data acquisition across heterogeneous environments, preprocessing and feature engineering, privacy-preserving federated training, deep learning–based behavioral modeling, and a dual evaluation of predictive performance and interpretability.

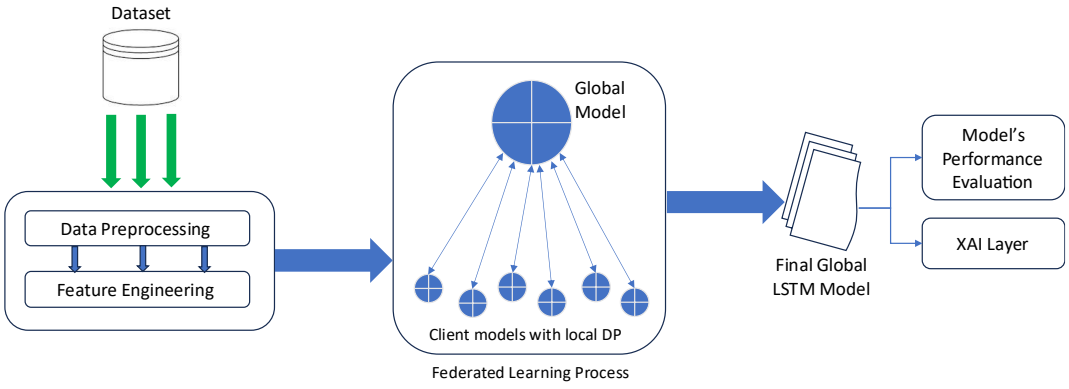

Figure 2. End-to-end methodology of the proposed framework across data preprocessing, federated learning, and model evaluation.

3.2.1. Data Acquisition and Environment Setup

To assess the robustness and generalizability of the proposed framework, two datasets with fundamentally different characteristics were selected, representing both conventional enterprise environments and modern cloud-native infrastructures. The CERT Insider Threat Dataset represents a synthetic yet well-controlled corporate IT environment. It consists of seven interconnected log files capturing user activities over a period of 517 days for 1,000 users. These logs include login and logoff events, file access records, email metadata, web

browsing activity, and removable media usage. Ground-truth labels identifying three distinct insider threat scenarios are provided through the accompanying answers.csv file. Owing to its structured nature and well-defined labeling, CERT serves as a controlled baseline for evaluating model behavior under interpretable and reproducible conditions.

In contrast, the BETH (BPF-Extended Tracking Honeypot) dataset reflects a real-world, cloud-native operational environment collected from honeypot deployments. The dataset contains more than 8 million events capturing low-level system interactions, including kernel-level process execution, system calls, and network communications generated by in-the-wild attacks. Unlike CERT, which is user-centric, BETH is entity-centric and characterized by high noise, high velocity, and complex interdependencies among system components. These properties make BETH a challenging and realistic testbed for evaluating the adaptability of insider threat detection models in modern infrastructures [1], [33].

3.2.2. Data Preprocessing and Feature Engineering

Raw security log data are inherently heterogeneous, noisy, and unstructured, rendering them unsuitable for direct input into deep learning models. Consequently, an extensive preprocessing and feature engineering pipeline was designed and applied separately to each dataset. This design reflects the fundamentally different characteristics of CERT and BETH while maintaining consistency in downstream modeling and evaluation.

For the CERT dataset, multiple log files were first merged and temporally aligned to construct a unified behavioral representation for each user. Daily aggregation was performed to generate a single behavioral snapshot per user per day, enabling consistent temporal modeling of long-term activity patterns. From these aggregated records, a total of 42 behavioral features were engineered to capture diverse aspects of user behavior. These features span several categories, including session-level activity (e.g., total sessions, average session duration, and after-hours logon count), file access behavior (e.g., total file accesses, sensitive file access count defined using keyword-based filtering, and file creation-to-deletion ratios), email usage patterns (e.g., number of emails sent, total email volume, and ratio of external recipients), and device and web interactions (e.g., USB insertion count and suspicious HTTP traffic volume). Each user–day instance was subsequently labeled as malicious or normal using the dataset's ground-truth annotations.

The BETH dataset, by contrast, captures fine-grained, low-level system activity from cloud-based environments and therefore requires a different preprocessing strategy. Instead of daily aggregation, raw events were grouped into sessions based on host identity, user context, and temporal proximity to form coherent activity sequences. Feature engineering focused on capturing low-level system behavior within each session, including process dynamics (e.g., process creation rate, number of unique parent processes, and frequency of sensitive commands such as sudo and chmod), system call behavior (e.g., entropy of the syscall distribution and counts of anomalous system calls associated with memory manipulation or privilege escalation), and network activity patterns (e.g., number of outbound connections, unique destination IPs, and UDP-to-TCP traffic ratios). Each session was labeled using honeypot-provided ground-truth annotations, ensuring precise identification of malicious activity.

To enable sequential learning, both datasets were transformed into overlapping time-series sequences, where each input sequence represents 10 consecutive time steps (days for CERT and sessions for BETH). This window length was selected to balance temporal context with computational efficiency. All feature values were normalized using Min–Max scaling, ensuring stable model convergence and preventing features with larger numeric ranges from dominating the learning process.

Given the relatively high dimensionality of the engineered feature sets (42 features for CERT and 38 for BETH), a feature selection stage was introduced to improve model stability and interpretability. A Random Forest–based feature importance analysis was conducted on centralized training data to rank features according to their predictive contribution. An elbow analysis of the resulting importance scores was then used to retain the top 20 most discriminative features for each dataset. This dimensionality reduction not only enhances training efficiency but also plays a critical role in improving the clarity and usability of downstream XAI explanations (SHAP and LIME) by focusing attention on the most influential behavioral indicators of insider threat activity.

3.2.3. Privacy-Preserving Federated Training Architecture

The core of the proposed methodology is a simulated FL architecture integrated with DP to ensure privacy preservation throughout the training process. To emulate a realistic distributed deployment, the preprocessed data from each dataset (CERT and BETH) were partitioned and distributed across N = 20 client nodes.

Rather than employing a random shuffle that assumes Independent and Identically Distributed (IID) data, a Non-IID partitioning strategy was adopted to reflect realistic organizational heterogeneity. For the CERT dataset, users were grouped according to their department or role (e.g., IT, Sales, HR), with each group assigned to a dedicated client node. For the BETH dataset, data were partitioned by host or service clusters. This strategy introduces natural data heterogeneity and enables evaluation of the model's ability to generalize across clients with distinct behavioral baselines without sharing local data distributions.

Model training was conducted over 50 communication rounds using the Federated Averaging (FedAvg) algorithm. At initialization, the central server defines the LSTM model architecture and randomly initializes its parameters. During each communication round, the server distributes the current global model weights to all participating clients. Each client (i.e., a network-connected device) receives the global model and performs localized training with the local epoch set to 3, after which only the updated model weights are returned to the server.

To mitigate inference attacks during federated model updates, Local Differential Privacy (LDP) was applied through input perturbation. Unlike gradient perturbation approaches such as DP-SGD, which inject noise during optimization, input perturbation obfuscates the normalized feature vectors before they are processed by the local LSTM model. This ensures that the training procedure itself never accesses exact raw behavioral values. Specifically, the Laplace mechanism was applied once to each normalized input feature vector x prior to training:

$$x' = x + \text{Laplace}\left(\frac{\Delta f}{\varepsilon}\right) \tag{1}$$

where x denotes the normalized input vector, Δf is the L_1 sensitivity (set to 1 due to Min–Max scaling), and ϵ represents the privacy budget. Values of $\varepsilon = 3.0$ and $\epsilon = 1.0$ (strong). Because noise is added to static input data rather than iteratively to gradients, privacy loss does not accumulate across communication rounds, rendering advanced accounting mechanisms such as Rényi Differential Privacy unnecessary for this implementation.

Although input noise generally challenges model convergence, LSTM networks are inherently robust to such perturbations due to their ability to capture temporal dependencies. By analyzing sequences of events rather than isolated observations, the LSTM effectively acts as a denoising filter, identifying persistent behavioral trends (signal) despite the presence of statistical noise (interference) introduced by the Laplace mechanism. This behavior is conceptually aligned with recurrent architectures used in denoising autoencoders, where temporal context enables recovery of coherent patterns from corrupted inputs.

After completing local training on perturbed data, each client transmits only its updated model weights to the central server. The server then performs secure aggregation by computing a weighted average of the received updates, producing an improved global model that is redistributed in the subsequent communication round.

3.2.4. Long Short-Term Memory (LSTM) Model Architecture

User behavior is inherently sequential rather than a collection of isolated events. The context and temporal ordering of actions are therefore critical for distinguishing between benign and malicious activities. LSTM networks are particularly well suited to this task due to their ability to model temporal dependencies and retain long-range contextual information.

Unlike standard RNNs, LSTMs employ a gating mechanism that regulates information flow through memory cells, enabling the retention of relevant behavioral context over extended periods [35]–[37]. This capability is essential for insider threat detection, where malicious behavior often unfolds gradually across multiple, non-consecutive actions. In addition, LSTMs are capable of automatically learning complex, non-linear interactions between features over time, directly from data. As a result, LSTM-based architectures are widely regarded as state-of-the-art (SOTA) for time-series classification tasks in insider threat detection.

The detection model is implemented as a stacked LSTM network, deployed identically at both the client and server levels. The input layer accepts sequences of shape (10, 20), where

10 represents the time-step window and 20 corresponds to the selected behavioral features. This configuration enables the model to analyze sequences of 10 consecutive events, each represented by a 20-dimensional feature vector.

The first LSTM layer contains 128 units and outputs the full sequence to capture fine-grained temporal patterns in user behavior. A dropout rate of 0.2 is applied to reduce overfitting and improve generalization. The second LSTM layer consists of 64 units, enabling hierarchical feature learning by modeling higher-level patterns derived from the first layer. A second dropout layer with a rate of 0.2 provides additional regularization. The output layer comprises a single neuron with a sigmoid activation function, producing a normalized risk score in the range $[0,1]$:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

Gate-level computations follow standard LSTM formulations. For example, the input gate is defined as:

$$i_t = \sigma(W_i x_t + W_h h_{t-1} + b_i) \tag{3}$$

where i_t is the input gate at time step t, x_t denotes the input vector at time t, h_{t-1} is the hidden state from the previous time step, W_i and W_h are weight matrices, b_i is the bias term, and $\sigma(\cdot)$ denotes the sigmoid activation function. The model is optimized using binary cross-entropy loss and the Adam optimizer.

3.2.5. Explainable AI (XAI) for Model Interpretation

Once the global FL model converged, two model-agnostic XAI techniques—SHAP and LIME—were applied to interpret model predictions at both global and local levels [38]. For SHAP, the KernelExplainer was employed due to its compatibility with LSTM-based architectures. To facilitate explanation, the temporal dimension of the input sequences was flattened during the explanation phase, transforming each input into a single vector of $10 \times 20 = 200$ dimensions. Global explanations were obtained by ranking features according to their average absolute Shapley values, revealing the most influential behavioral indicators across the dataset. Local explanations, in contrast, highlighted how specific features contributed to increasing or decreasing the risk score for individual alerts.

The Shapley value for feature i is computed as:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} [f(S \cup \{i\}) - f(S)] \tag{4}$$

where F is the full feature set, M is the number of features, S is a subset excluding feature i, and $f(\cdot)$ denotes the model prediction.

Table 2. Mitigation of privacy risks associated with XAI

Privacy Risk	Mitigation Strategy
Feature reconstruction / model inversion	Explanations are generated solely from the differentially private global model, whose linkage to individual users' raw data is already obfuscated by calibrated noise, thereby limiting the feasibility of feature reconstruction and model inversion attacks.
Membership inference	The influence of any single user on the global model parameters is mathematically bounded under Differential Privacy, making it computationally impractical to infer whether a specific user contributed to the training data from model explanations.
General information leakage	Access to XAI outputs is restricted to authorized security analysts on a strict need-to-know basis. In addition, explanations can be abstracted to conceal raw data values while preserving semantic meaning and investigative relevance.

In parallel, LIME was applied as a complementary technique to generate instance-level explanations. For each prediction, LIME constructs a local surrogate model by generating perturbed samples around the original input and fitting a sparse linear approximation. The

explanation complexity was constrained to the top 3–5 most influential features, ensuring clarity and actionability for security analysts. The LIME optimization objective is given by:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \quad (5)$$

where f is the original model, g is the interpretable surrogate model, π_x is the proximity measure, and $\Omega(g)$ controls explanation complexity.

Although XAI techniques may introduce potential privacy risks, these risks are mitigated by the privacy-by-design nature of the proposed framework. Explanations are generated exclusively from the differentially private global model, whose parameters have a mathematically bounded dependence on any individual user's data. Furthermore, explanation access is restricted to authorized analysts on a need-to-know basis, and explanations can be abstracted to conceal raw data values while preserving semantic meaning. A summary of these mitigation strategies is provided in Table 2.

3.2.6. Experimental Setup and Evaluation Metrics

To rigorously evaluate the effectiveness of the proposed framework, a comparative experimental setup was designed that contrasts privacy-preserving federated training with a traditional centralized baseline. For each dataset, a centralized LSTM model trained without privacy mechanisms was used as the reference baseline. This allows direct quantification of the utility cost introduced by FL and DP. Both centralized and federated models shared the same architecture, hyperparameters, and feature sets to ensure fair comparison.

Model performance was evaluated using Accuracy, Precision, Recall, F1-score, and AUC-ROC, with recall as the primary metric given its importance in minimizing missed insider threats. All experiments were repeated five times with different random seeds for weight initialization and client data partitioning. Results are reported as mean ± standard deviation to ensure statistical robustness. In addition to quantitative evaluation, the quality of explanations was assessed through a simulated user study involving 15 participants with professional experience in SOC operations or incident response. Participants were presented with 20 anonymized alerts (10 from CERT and 10 from BETH), including a balanced mix of true-positive and false-positive cases. Each alert was accompanied by SHAP and LIME explanations.

Participants rated each explanation using a 5-point Likert scale across three dimensions:

- Clarity: ease of understanding the explanation,
- Actionability: usefulness for guiding next investigative steps,
- Trust: confidence in the model's prediction after reviewing the explanation.

The study was conducted asynchronously using a structured online questionnaire, and participants were given 72 hours to complete the evaluation. Aggregate scores were used to assess the practical effectiveness of the explanation methods in real-world analytical workflows.

4. Results and Discussion

This section presents a comprehensive analysis of the experimental results. We first report the quantitative detection performance of the proposed privacy-preserving framework and compare it with a non-private centralized baseline across both the CERT and BETH datasets. We then contextualize the results against SOTA approaches and discuss the observed privacy–utility trade-off. Finally, we analyze the computational overhead to assess the practical feasibility of deployment in real organizational environments.

4.1. Model Performance: Comparative Analysis

The primary objective of this evaluation is to determine whether the proposed framework can achieve accurate insider threat detection while preserving privacy across heterogeneous environments. To this end, the performance of the FL model with DP ($\varepsilon = 3.0$) was compared with that of a standard centralized LSTM model trained without privacy mechanisms. Table 3 summarizes the comparative results for both datasets.

As shown in Table 3, adopting a federated, privacy-preserving training architecture results in a marginal decrease in performance relative to centralized training. On the CERT dataset, the reduction in F1-score is approximately 2.2%, which is statistically significant (paired t-test, $p = 0.042$) but remains operationally acceptable given the privacy guarantees

provided by the framework. Similar trends are observed on the BETH dataset, indicating that the privacy–utility trade-off is consistent across both synthetic and real-world environments.

Table 3. Comparative performance metrics of detection models.

Dataset	Model Configuration	F1-Score	Accuracy (%)	Precision	Recall	Privacy Guarantee (ε)
CERT	Centralized	0.90 ± 0.01	94.7	0.89	0.92	None
	Federated + DP	0.88 ± 0.02	92.5	0.86	0.90	3.0
BETH	Centralized	0.88 ± 0.02	91.8	0.87	0.89	None
	Federated + DP	0.86 ± 0.03	90.1	0.84	0.88	3.0

To further contextualize these results, Table 4 compares the proposed framework with representative SOTA approaches evaluated on the CERT dataset. Centralized models reported in prior studies generally achieve higher accuracy and F1 Scores by leveraging unrestricted access to raw and aggregated user data. In contrast, the proposed federated framework operates under strict privacy constraints and therefore accepts a controlled performance degradation in exchange for formal privacy guarantees and model transparency.

Table 4. Comparison with SOTA approaches on CERT.

Reference	Method	Accuracy (%)	F1-Score	Privacy Mechanism	Explainability
Nasir et al. [9]	Deep LSTM	93.2	0.91	None	No
Kotb et al. [19]	DS-IID (GenAI)	97.0	0.96	None	No
Proposed Work	Fed-LSTM + DP	92.5	0.88	LDP + FL	Yes

Centralized approaches, such as the DS-IID model proposed by Kotb et al. [19], achieve higher detection performance by exploiting direct access to raw user data. The proposed framework achieves a competitive F1 Score of 0.88 while providing formal privacy guarantees and explainability. The observed performance gap of approximately 8% represents the privacy cost associated with decentralized training and input perturbation. Importantly, unlike prior SOTA methods that operate as opaque centralized models, the proposed framework offers two system-level advantages: data sovereignty through FL and actionable transparency through XAI. These properties are critical for organizations operating under strict privacy regulations, where marginal performance gains do not justify centralized collection of sensitive employee data.

4.2. Confusion Matrix Analysis

While aggregate performance metrics provide a high-level comparison between detection models, confusion matrix analysis offers deeper insight into the operational behavior of the proposed framework, particularly with respect to false positives and false negatives. In the context of insider threat detection, recall is treated as the primary performance indicator, as missed malicious activities pose substantially higher risk than false alarms, with precision serving as an operational constraint.

4.2.1. Confusion Matrix for the CERT Dataset

The confusion matrix for the CERT dataset is reported in Table 5, detailing the distribution of true positives, false positives, true negatives, and false negatives produced by the proposed framework. As shown in Table 5, the resulting recall is 90.0%, indicating that the framework successfully identifies 9 out of 10 malicious insider activities. The 304 false negatives represent residual detection risk, while the 86% precision demonstrates strong operational viability. Specifically, for every 100 alerts generated, 86 correspond to genuine threats, ensuring that security analysts are not overwhelmed by excessive false alarms.

4.2.2. Confusion Matrix for the BETH Dataset

Similarly, the confusion matrix for the BETH dataset is presented in Table 6, reflecting the model's behavior under the more complex, noisy conditions of a cloud-native environment. As indicated in Table 6, the framework achieves a recall of 88.0%, demonstrating robust

generalization to real-world cloud environments characterized by high-volume and low-level system activity. This result confirms the model's ability to detect the majority of subtle attacks captured in kernel and system-call data. The corresponding precision of 84.0% indicates that improved sensitivity is achieved without imposing excessive operational burden, thereby maintaining a manageable alert volume for security teams.

Table 5. Confusion matrix for the CERT dataset.

	Predicted: BENIGN	**Predicted: THREAT**
Actual: BENIGN	TN: 6,512	FP: 446
Actual: THREAT	FN: 304	TP: 2,738

Table 6. Confusion matrix for the BETH dataset.

	Predicted: BENIGN	**Predicted: THREAT**
Actual: BENIGN	TN: 5,981	FP: 577
Actual: THREAT	FN: 413	TP: 3,029

4.3. Computational Overhead Analysis

Beyond detection performance, the computational feasibility of the proposed framework was evaluated to assess its suitability for real-world deployment. The LSTM model size is approximately 450 KB, enabling efficient distribution across client nodes. Over 50 communication rounds, the total data transfer per client is approximately 22.5 MB, which is negligible for modern enterprise networks.

Moreover, using input perturbation (LDP) instead of gradient-based DP (e.g., DP-SGD) reduced local training time per epoch by 40%, significantly lowering the computational burden on client devices. This reduction empirically supports the design choice of applying privacy at the input level, confirming that the framework can be deployed on standard employee workstations without disrupting normal operations or productivity.

4.4. Analyzing the Privacy–Utility Trade-off

To explicitly examine the relationship between privacy strength and detection performance, the F1-score was plotted against different values of the privacy budget ε, as shown in Figure 3. In the context of Differential Privacy, smaller values of ε correspond to stronger privacy guarantees achieved through higher noise injection, which typically results in reduced model utility.

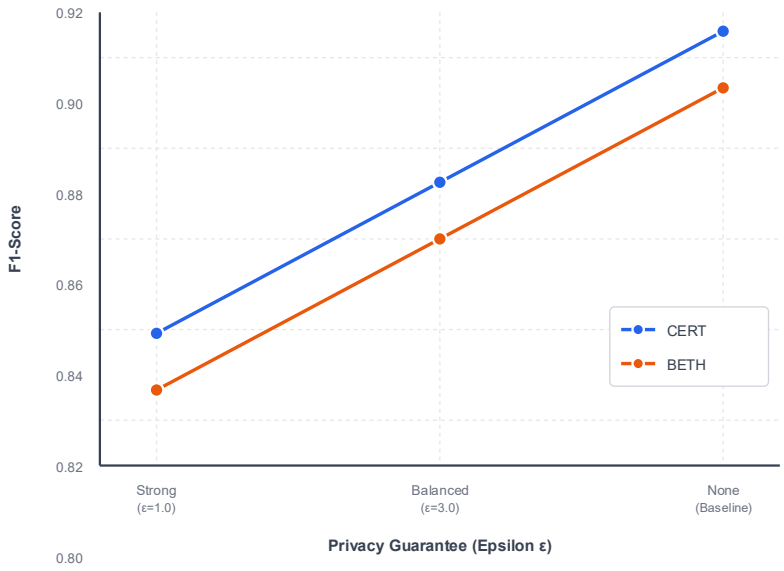

Figure 3. Privacy–utility trade-off showing the effect of different privacy budgets (ε) on F1-score for CERT and BETH datasets.

Figure 3 clearly illustrates a graceful and predictable trade-off between privacy and utility for both datasets. The baseline models, which operate without privacy constraints ($\varepsilon = \infty$), achieve the highest F1-scores. As privacy is introduced by lowering ε, performance decreases gradually rather than abruptly, indicating that the proposed framework maintains stability even under privacy constraints.

At the balanced privacy level ($\varepsilon = 3.0$), the performance reduction is minimal and remains within operationally acceptable bounds for both CERT and BETH, consistent with the quantitative results reported in Table 2. In contrast, at the strong privacy level ($\varepsilon = 1.0$), the F1-score decreases more noticeably, reaching approximately 0.85 for CERT and 0.82 for BETH. While this setting offers stronger privacy guarantees, the resulting utility degradation may limit practical deployment in high-stakes detection environments.

From an operational perspective, this trade-off curve provides a practical decision-making tool for organizations. It allows system operators to select an appropriate privacy level based on regulatory requirements, threat tolerance, and acceptable detection performance. It is important to note that within the LDP setting, a privacy budget of $\varepsilon = 3.0$ represents a pragmatic compromise. Although theoretical cryptographic literature often advocates for $\varepsilon < 1.0$, our empirical results demonstrate that such strict privacy bounds in an LDP context degrade utility below acceptable levels for insider threat detection. Therefore, $\varepsilon = 3.0$ offers a balanced operating point that preserves plausible deniability for users while maintaining actionable detection capability for security teams.

4.5. Impact and Quality of Explainable AI (XAI)

While quantitative metrics demonstrate the predictive capability of the proposed model, its practical value in real-world security operations ultimately depends on the quality of the explanations it provides. Following the evaluation protocol described in Section 3.2.6, the XAI layer was assessed across three dimensions: clarity, actionability, and trust. The results of the qualitative user study confirm that the XAI component significantly enhances the system's operational usability and trustworthiness. As shown in Figure 4, explanations generated for both the CERT and BETH datasets received consistently high ratings across all three dimensions, with average scores exceeding 4.2 on a 5-point scale. This indicates that the explanations were not only understandable but also directly useful in guiding investigative actions. The strong performance in actionability further validates the feature selection strategy described in Section 3.2.2, which deliberately reduced the feature space to the 20 most discriminative indicators, ensuring that explanations remained focused, concise, and cognitively manageable for analysts.

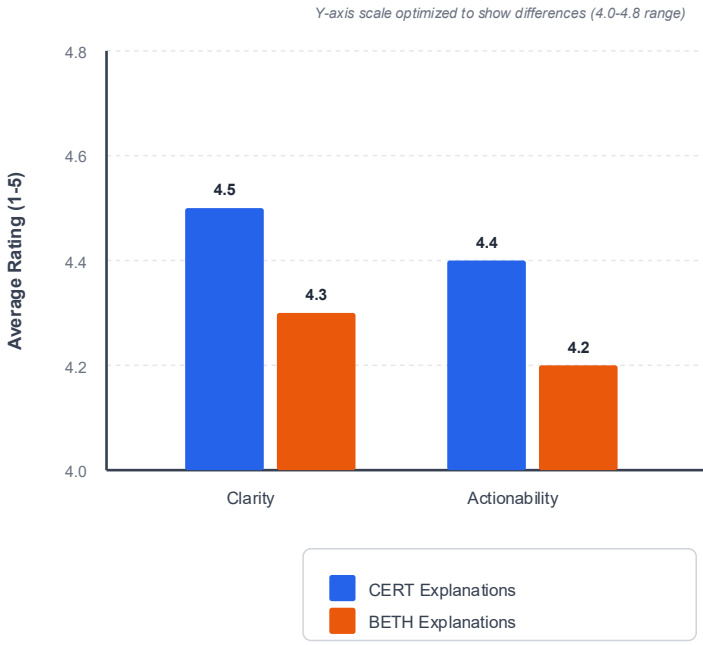

Figure 4. Qualitative evaluation of XAI layer outputs through user surveys

Beyond aggregate scores, the operational value of the XAI layer is best illustrated through representative case studies drawn from both datasets, which highlight how explanations adapt to different data environments while preserving interpretability. In the CERT case study (corporate data exfiltration scenario), an alert was generated for a user exhibiting anomalous behavior. The SHAP force plot in Figure 5 shows that features such as email_to_external_domain_size = 750MB, file_access_after_hours = 312, and usb_insertion_weekend = TRUE contributed strongly and positively to the predicted risk score. The corresponding LIME explanation in Figure 6 reinforces this narrative by ranking the same features as the most influential contributors. Together, these explanations provide a coherent and intuitive story: a user accessed a large number of files outside working hours and subsequently transferred a substantial volume of data externally using removable media. This pattern represents a classic, actionable data exfiltration scenario, enabling analysts to quickly validate the alert and prioritize the response.

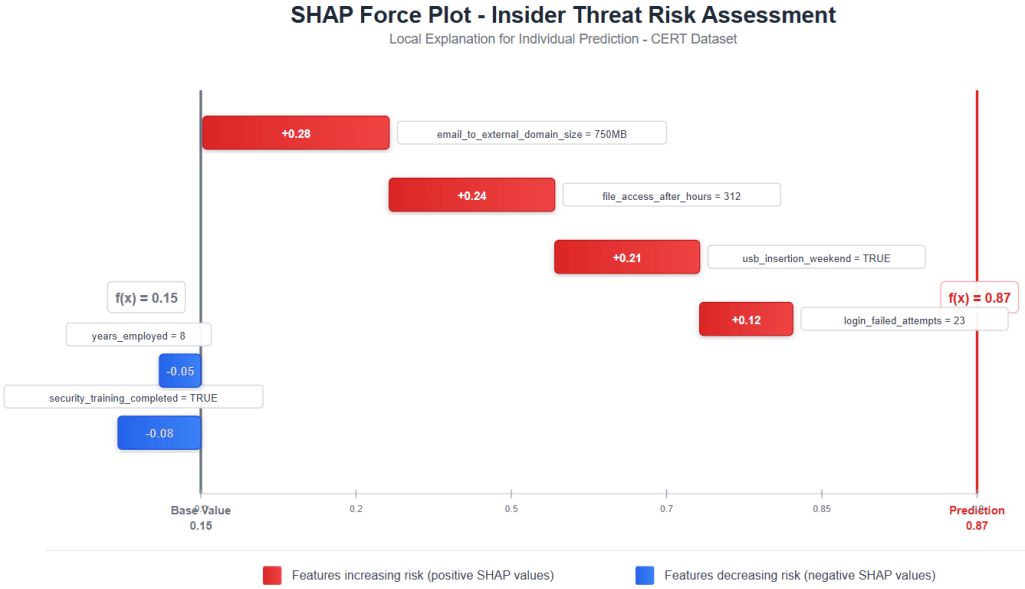

Figure 5. SHAP force plot for an insider threat

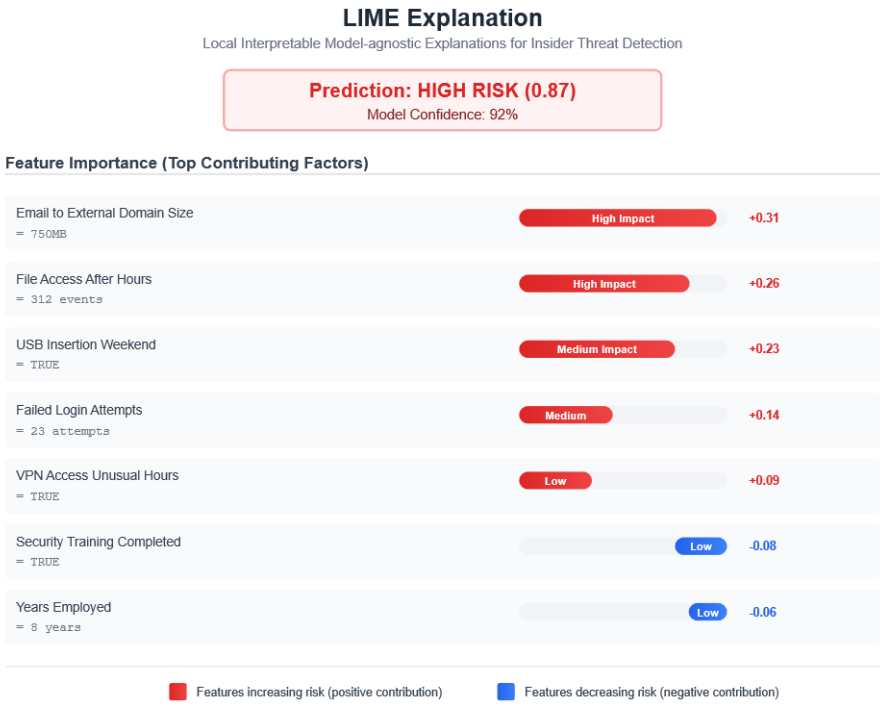

Figure 6. LIME explanation for an insider threat

In contrast, the BETH case study (cloud instance compromise scenario) demonstrates the XAI layer's adaptability to low-level system telemetry. As illustrated in Figures 7 and 8, the model identified a different but equally interpretable set of indicators, including anomalous_process_spawn = /bin/bash, outbound_connection_to_rare_ip = 1, and frequency_of_syscall_setuid = 5. These features collectively suggest the potential for remote shell execution, followed by privilege escalation and command-and-control activity. Importantly, the explanations shift from user-centric semantics (CERT) to system-centric semantics (BETH) without loss of clarity, indicating that the XAI layer generalizes across heterogeneous environments.

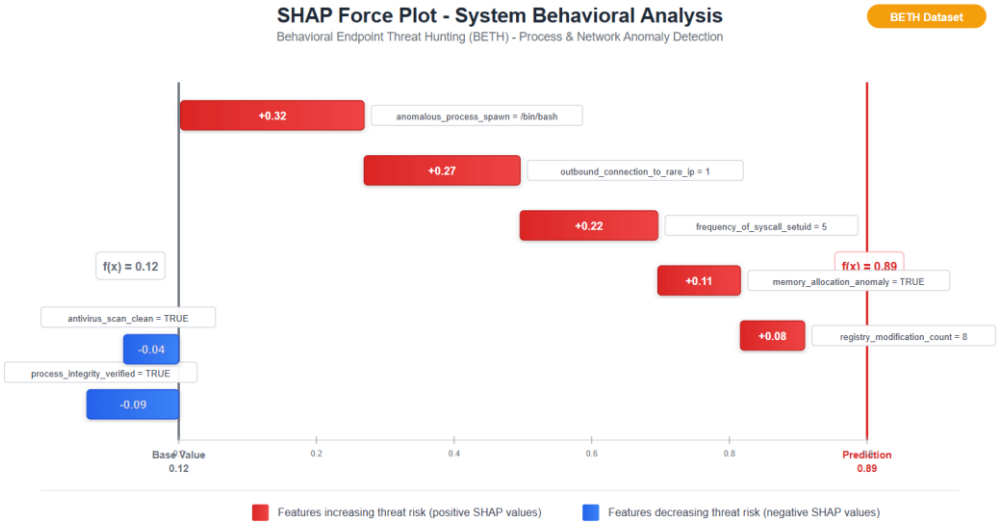

Figure 7. SHAP force plot of a cloud user's behavioural analysis

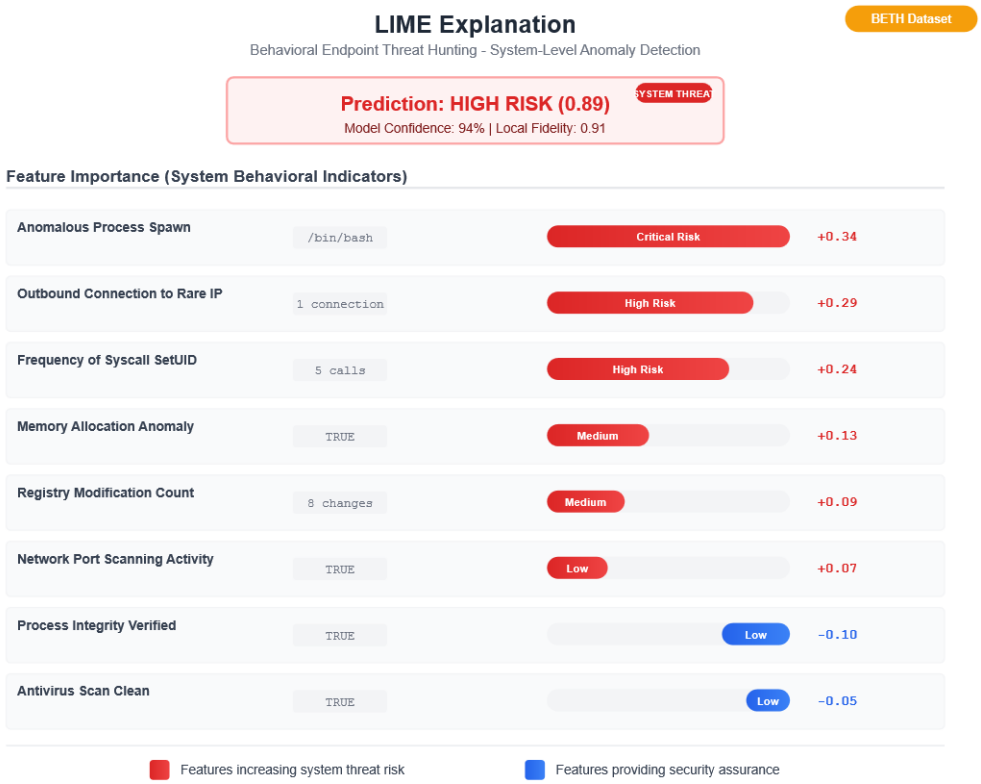

Figure 8. LIME Explanation for a Cloud User's Behaviour

Taken together, these results demonstrate that the XAI component does more than merely explain model predictions—it translates complex behavioral patterns into context-

specific, actionable intelligence that aligns with analyst workflows. This capability is essential for insider threat detection systems, where trust, accountability, and rapid interpretability are as critical as detection accuracy itself.

5. Conclusions

This study demonstrates that it is feasible to design an insider threat detection system that simultaneously achieves accurate detection, formal privacy protection, and operational interpretability, while remaining robust across heterogeneous IT environments. Experimental results on both the structured, user-centric CERT dataset and the noisy, entity-centric BETH dataset confirm the architectural versatility of the proposed framework and its ability to generalize across traditional enterprise and cloud-native infrastructures. These findings directly support the research objective of building a unified detection system that does not sacrifice trust or deploy ability for performance.

A key contribution of this work is reframing the privacy–utility trade-off from a limitation to a controllable design parameter. By explicitly quantifying the impact of privacy budgets on detection performance, the proposed methodology enables organizations to adopt risk-calibrated privacy strategies rather than rigid, all-or-nothing data protection policies. The empirical observation that meaningful privacy guarantees can be achieved with only a minor and operationally acceptable reduction in F1-score provides strong evidence that privacy-preserving insider threat detection is not only theoretically viable but also practically deployable in regulated environments.

Equally important, integrating XAI transforms the system from a purely predictive model into a decision-support tool. The consistently high clarity, actionability, and trust scores reported in the user study demonstrate that explanations effectively bridge the gap between model predictions and analyst reasoning. By providing transparent justifications for alerts, the framework reduces cognitive load, improves alert prioritization, and shortens investigation time, addressing one of the most persistent challenges in security operations: alert fatigue. This human–AI synergy represents a critical step toward the adoption of trustworthy AI in cybersecurity.

Despite these contributions, this study has limitations that open avenues for future research. First, the privacy mechanism currently applies a uniform privacy budget across all features; future work will explore adaptive privacy budgets, where noise levels are dynamically assigned based on feature sensitivity and predictive importance. Second, to better accommodate behavioral heterogeneity across roles, personalized FL will be investigated to allow local adaptation while maintaining global knowledge sharing. Finally, future iterations will consider integrating GNNs to explicitly model relational dependencies among users, hosts, and resources, further enhancing the detection of coordinated or multi-stage insider activities. In summary, this work contributes a practical, privacy-preserving, and explainable insider threat detection framework that advances both methodological rigor and operational relevance, offering a concrete pathway toward deploying trustworthy AI in real-world cybersecurity systems.

Author Contributions: Conceptualization: O. A. and O. J.-F.; Methodology: O. J.-F.; Software: O. J.-F.; Validation: O. A., O. J.-F. and A. M.; Formal analysis: O. A.; Investigation: A. M.; Resources: A. M.; Data curation: O. J.-F.; Writing—original draft preparation: O. J.-F.; Writing—review and editing: A. M.; Visualization: A. M.; Supervision: O. A.; Project administration: O. A.; Funding acquisition: Nil. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The datasets used in this research are the CERT (https://kilthub.cmu.edu/articles/dataset/Insider_Threat_Test_Dataset/12841247) and BETH (https://www.kaggle.com/datasets/katehighnam/beth-dataset) datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] B. Bin Sarhan and N. Altwaijry, "Insider Threat Detection Using Machine Learning Approach," *Appl. Sci.*, vol. 13, no. 1, p. 259, Dec. 2022, doi: 10.3390/app13010259.

[2] Matthew D. Waters, "Identifying and Preventing Insider Threats," Eastern Kentucky University, 2016. [Online]. Available: https://encompass.eku.edu/cgi/viewcontent.cgi?article=1371&context=honors_theses

[3] G. Mazzarolo and A. D. Jurcut, "Insider threats in Cyber Security: The enemy within the gates," *arXiv*. Nov. 21, 2019. [Online]. Available: http://arxiv.org/abs/1911.09575

[4] K. Fei and J. Zhou, "An Insider Threat Investigation Method by Graph Analysis with Log Texts," in *Proceedings of the 2024 3rd International Conference on Networks, Communications and Information Technology*, Jun. 2024, pp. 19–23. doi: 10.1145/3672121.3672126.

[5] A. Trivedi, "Cybersecurity and Insider Threat Detection: The Role of User Behavior Analytics (UBA) in Modern Defense Strategies," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 13, no. 1, pp. 455–466, Jan. 2025, doi: 10.22214/ijraset.2025.66298.

[6] K. Indranil Iyer, "Behavioral Intelligence at Scale: Implementing UEBA for Enhanced Security Posture," *Int. J. Sci. Res.*, vol. 11, no. 7, pp. 1971–1977, Jul. 2022, doi: 10.21275/SR22074090532.

[7] M. A. Salitin and A. H. Zolait, "The role of User Entity Behavior Analytics to detect network attacks in real time," in *2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Nov. 2018, pp. 1–5. doi: 10.1109/3ICT.2018.8855782.

[8] K. I. Iyer, "Proactive Threat Hunting: Leveraging AI for Early Detection of Advanced Persistent Threats," *Eur. J. Adv. Eng. Technol.*, vol. 11, no. 2, pp. 69–76, 2024, [Online]. Available: https://ejaet.com/PDF/11-2/EJAET-11-2-69-76.pdf

[9] R. Nasir, M. Afzal, R. Latif, and W. Iqbal, "Behavioral Based Insider Threat Detection Using Deep Learning," *IEEE Access*, vol. 9, pp. 143266–143274, 2021, doi: 10.1109/ACCESS.2021.3118297.

[10] I. Idris and A. N. Damilola, "Systematic Literature Review and Metadata Analysis of Insider Threat Detection Mechanism," *Int. J. Comput. Sci. Mob. Comput.*, vol. 12, no. 4, pp. 60–88, Apr. 2023, doi: 10.47760/ijcsmc.2023.v12i04.007.

[11] B. Luthor, L. Lex, and S. Iseal, "Integrating AI-Powered Behavioral Analytics into Cybersecurity Frameworks for Proactive Threat Detection," *Research Gate*. 2025. [Online]. Available: https://www.researchgate.net/publication/393325024_Integrating_AI-Powered_Behavioral_Analytics_into_Cybersecurity_Frameworks_for_Proactive_Threat_Detection

[12] U. Inayat, M. Farzan, S. Mahmood, M. F. Zia, S. Hussain, and F. Pallonetto, "Insider threat mitigation: Systematic literature review," *Ain Shams Eng. J.*, vol. 15, no. 12, p. 103068, Dec. 2024, doi: 10.1016/j.asej.2024.103068.

[13] I. A. Gheyas and A. E. Abdallah, "Detection and prediction of insider threats to cyber security: a systematic literature review and meta-analysis," *Big Data Anal.*, vol. 1, no. 1, p. 6, Dec. 2016, doi: 10.1186/s41044-016-0006-0.

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?' : Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-Augu, pp. 1135–1144. doi: 10.1145/2939672.2939778.

[15] M. Bandgar, "Intrusion Detection System using Hidden Markov Model (HMM)," *IOSR J. Comput. Eng.*, vol. 10, no. 3, pp. 66–70, 2013, doi: 10.9790/0661-01036670.

[16] S. Yuan and X. Wu, "Deep learning for insider threat detection: Review, challenges and opportunities," *Comput. Secur.*, vol. 104, p. 102221, May 2021, doi: 10.1016/j.cose.2021.102221.

[17] D. R. I. M. Setiadi, S. Widiono, A. N. Safriandono, and S. Budi, "Phishing Website Detection Using Bidirectional Gated Recurrent Unit Model and Feature Selection," *J. Futur. Artif. Intell. Technol.*, vol. 1, no. 2, pp. 75–83, Jul. 2024, doi: 10.62411/faith.2024-15.

[18] A. A. A. Al-Ameer and A. A. Huby, "Hybrid BiLSTM-SVM Intrusion Detection with Decision-Based Flow Ranking," *Int. J. Saf. Secur. Eng.*, vol. 15, no. 1, pp. 67–72, Jan. 2025, doi: 10.18280/ijsse.150107.

[19] H. M. Kotb, T. Gaber, S. AlJanah, H. M. Zawbaa, and M. Alkhathami, "A novel deep synthesis-based insider intrusion detection (DS-IID) model for malicious insiders and AI-generated threats," *Sci. Rep.*, vol. 15, no. 1, p. 207, Jan. 2025, doi: 10.1038/s41598-024-84673-w.

[20] Y. Gong, S. Cui, S. Liu, B. Jiang, C. Dong, and Z. Lu, "Graph-based insider threat detection: A survey," *Comput. Networks*, vol. 254, p. 110757, Dec. 2024, doi: 10.1016/j.comnet.2024.110757.

[21] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20 th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Jan. 2017. [Online]. Available: http://arxiv.org/abs/1602.05629

[22] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated Learning: Challenges, Methods, and Future Directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020, doi: 10.1109/MSP.2020.2975749.

[23] Y. Liu *et al.*, "FedVision: An Online Visual Object Detection Platform Powered by Federated Learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 08, pp. 13172–13179, Apr. 2020, doi: 10.1609/aaai.v34i08.7021.

[24] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Futur. Gener. Comput. Syst.*, vol. 115, pp. 619–640, Feb. 2021, doi: 10.1016/j.future.2020.10.007.

[25] N. Truong, K. Sun, S. Wang, F. Guitton, and Y. Guo, "Privacy preservation in federated learning: An insightful survey from the GDPR perspective," *Comput. Secur.*, vol. 110, p. 102402, Nov. 2021, doi: 10.1016/j.cose.2021.102402.

[26] S. Han *et al.*, "FedSecurity: A Benchmark for Attacks and Defenses in Federated Learning and Federated LLMs," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug. 2024, pp. 5070–5081. doi: 10.1145/3637528.3671545.

[27] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 1–19. doi: 10.1007/978-3-540-79228-4_1.

[28] I. Namatevs, K. Sudars, A. Nikulins, and K. Ozols, "Privacy Auditing in Differential Private Machine Learning: The Current Trends," *Appl. Sci.*, vol. 15, no. 2, p. 647, Jan. 2025, doi: 10.3390/app15020647.

[29] S. Javed *et al.*, "Secure and Interpretable Intrusion Detection through Federated and Ensemble Machine Learning with XAI," *J. Comput. Biomed. Informatics*, vol. 9, no. 1, 2025, doi: 10.56979/901/2025.

[30] R. R. Karn, P. Kudva, H. Huang, S. Suneja, and I. M. Elfadel, "Cryptomining Detection in Container Clouds Using System Calls and Explainable Machine Learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 3, pp. 674–691, Mar. 2021, doi: 10.1109/TPDS.2020.3029088.

[31] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *arXiv*, Nov. 2017. [Online]. Available: http://arxiv.org/abs/1705.07874

[32] G. Rjoub *et al.*, "A Survey on Explainable Artificial Intelligence for Cybersecurity," *IEEE Trans. Netw. Serv. Manag.*, vol. 20, no. 4, pp. 5115–5140, Dec. 2023, doi: 10.1109/TNSM.2023.3282740.

[33] K. Highnam, K. Arulkumaran, Z. Hanif, and N. R. Jennings, "BETH Dataset: Real Cybersecurity Data for Unsupervised Anomaly Detection Research," in *CEUR Workshop Proceedings*, 2021, vol. 3095, pp. 1–12. [Online]. Available: https://ceur-ws.org/Vol-3095/paper1.pdf

[34] K. Fatema *et al.*, "Federated XAI IDS: An Explainable and Safeguarding Privacy Approach to Detect Intrusion Combining Federated Learning and SHAP," *Futur. Internet*, vol. 17, no. 6, p. 234, May 2025, doi: 10.3390/fi17060234.

[35] N. Chen, "Exploring the development and application of LSTM variants," *Appl. Comput. Eng.*, vol. 53, no. 1, pp. 103–107, Mar. 2024, doi: 10.54254/2755-2721/53/20241288.

[36] D. R. I. M. Setiadi *et al.*, "Integrating Hybrid Statistical and Unsupervised LSTM-Guided Feature Extraction for Breast Cancer Detection," *J. Comput. Theor. Appl.*, vol. 2, no. 4, pp. 536–552, May 2025, doi: 10.62411/jcta.12698.

[37] Y. Görmez, H. Arslan, Y. E. Işik, and V. Gündüz, "Developing Novel Deep Learning Models to Detect Insider Threats and Comparing the Models from Different Perspectives," *Bilişim Teknol. Derg.*, vol. 17, no. 1, pp. 31–43, Jan. 2024, doi: 10.17671/gazibtd.1386734.

[38] A. I. U. Akpan Itoro Udofot, O. M. O. Omotosho Moses Oluseyi, and E. B. E. Edim Bassey Edim, "Explainable AI for cyber security. Improving transparency and trust in intrusion detection systems," *Int. J. Adv. Eng. Manag.*, vol. 06, no. 12, pp. 229–240, Dec. 2024, doi: 10.35629/5252-0612229240.