Behavior-Based Insider Threat Detection Using Machine Learning: A Comparative Study of Isolation Forest, Random Forest, and XGBoost

Anonymous Authors

Anonymous Institution

Abstract. Insider threats significantly complicate cybersecurity, as malicious activities can be carried out using legitimate user credentials and may closely resemble normal organizational behavior. This study investigates insider threat detection using the CERT Insider Threat Dataset r4.2 through user behavioral analytics and machine learning. Logon, device, file, and email activity logs were processed and transformed into user-day behavioral records. A total of 49 behavioral features were constructed to capture user activity frequency, computer usage patterns, temporal behavior, email activities, and deviations from historical user behavior. Due to the highly imbalanced nature of the dataset, precision, recall, F1-score, ROC-AUC, and PR-AUC were considered alongside accuracy for comprehensive evaluation. Among the evaluated techniques, XGBoost using the complete 49-feature set achieved the best overall performance, with an accuracy of 99.78%, precision of 79.23%, recall of 59.43%, F1-score of 67.92%, ROC-AUC of 99.57%, and PR-AUC of 78.97%. The results demonstrate that temporal, activity-based, and historical behavioral features play a crucial role in improving the detection of anomalous insider activities compared with simpler behavioral representations and conventional anomaly detection approaches. The findings highlight the effectiveness of behavioral feature engineering combined with machine learning for detecting insider threats in highly imbalanced cybersecurity environments.
Keywords: Insider Threat Detection, User Behavioral Analytics, Machine Learning, CERT Dataset, XGBoost, Anomaly Detection

1 Introduction

The problem of insider threats has emerged as an important cybersecurity issue because there are some malicious actions that can be committed by insiders who are authorized users that have legitimate access to system infrastructure, applications, and data of organizations. In contrast to attacks from the outside, insiders can perform their malicious actions without going through the process of breaching the security mechanisms of an organization like authentication mechanisms, firewalls, or intrusion detection systems. As a result, it is very hard to differentiate between legitimate activities and malicious ones committed by insiders. The insiders can use their access to commit different kinds of inappropriate or malicious activities such as stealing the information, misusing their privileges, sabotaging the system, and gaining unauthorized access to organizational resources [1, 2, 4].

An issue encountered while detecting insider threats is the presence of overlap between benign and malicious actions of the insiders. The insiders have a valid authentication and perform actions that can be considered normal on their own but whose sequence, frequency, timing, and combination can be used as indicators of malicious behavior. Therefore, traditional signature-based and rule-based security methods may fail in recognizing such activities since the actions are not recognized beforehand. The authors Kim et al. highlight the significance of modeling individual users' behaviors through conversion of unstructured organizational actions into structured behaviors and detection of anomalies [2]. In a similar fashion, the authors Hall et al. convert the organizational logs into daily activities of the users and use various machine learning techniques to recognize malicious insider threat situations with the help of CERT dataset [3].

User Behavioral Analytics (UBA) constitutes an appropriate method to solve this problem through modeling the typical behavior of users and detecting activities that deviate from normal behavioral patterns. Besides using signatures for identifying malicious software or malware, behavioral models can consider multiple data sources like user logons, file accesses, device accesses, emails, and computer usage. Such approaches' success heavily depends on how raw logs can be converted into useful behavioral features. Recent literature suggests the usage of structured user-level, daily-level, and session-level representations to leverage machine learning and anomaly detection techniques for insider threat detection [2, 5]. Muñoz et al. also showed that converting raw logs to structured session-level features allowed for applying supervised and unsupervised machine learning algorithms to CERT and SPEDIA datasets, where XGBoost provided excellent detection results [5].

Machine learning has consequently become an important research direction for insider threat detection because it can learn behavioral patterns from large volumes of organizational activity data. Both supervised and unsu-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

pervised approaches have been investigated. Supervised classifiers can learn the distinction between benign and malicious behavioral records when labeled data are available, whereas unsupervised anomaly detection methods attempt to identify deviations from normal behavior without requiring extensive malicious examples. Previous studies have evaluated algorithms such as Random Forest, Support Vector Machines, XGBoost, Isolation Forest, and other anomaly detection techniques on insider threat datasets [3–5]. Bin Sarhan and Altwaijry, for example, investigated both anomaly detection and classification approaches on the CERT Insider Threat Dataset and showed that behavioral feature extraction and machine learning can be used to identify insider threats [4]. However, the highly imbalanced nature of insider threat data remains a significant challenge because malicious activities represent only a very small proportion of overall organizational activity [4, 6].

The CERT Insider Threat Dataset r4.2 is widely used as a benchmark for evaluating insider threat detection methods. However, its severe class imbalance creates a challenging detection environment in which a model can obtain high overall accuracy while still failing to identify a substantial proportion of malicious activities. Recent research has also pointed out that existing insider threat datasets can suffer from issues such as limited realism, severe imbalance, and outdated behavioral scenarios [6]. These characteristics make the selection of evaluation metrics particularly important. Precision, recall, F1-score, ROC-AUC, and PR-AUC provide more informative measures of detection capability than accuracy alone when the malicious class is heavily underrepresented.

Another important consideration is behavioral feature engineering. Existing research has explored daily activity summaries, email communication patterns, sequential behavioral features, and other representations to capture deviations from normal user behavior [2, 8]. Recent studies have also investigated sequential models such as BiLSTM and Transformer-based approaches to capture temporal dependencies in user activity [8, 9]. While these approaches demonstrate the potential of sequential modeling, there remains value in systematically evaluating conventional supervised and unsupervised machine learning methods using carefully engineered behavioral features. In particular, comparing a classical anomaly detection method such as Isolation Forest with supervised ensemble methods such as Random Forest and XGBoost provides a useful basis for understanding the effect of behavioral representation and learning strategy on insider threat detection.

Motivated by these challenges, this study investigates the effectiveness of user behavioral analytics for detecting insider threats using the CERT Insider Threat Dataset r4.2. Logon, device, file, and email activity logs are transformed into user-day behavioral records. A set of 49 behavioral features is engineered to capture activity volume, computer usage, temporal behavior, email characteristics, and deviations from users' historical behavior. The resulting behavioral representation is evaluated using three machine learning approaches: Isolation Forest as an unsupervised anomaly detection baseline, Random Forest as a supervised ensemble classifier, and XGBoost as the primary supervised model. A chronological train-validation-test split is used to provide a more realistic evaluation setting and to reduce the possibility of temporal information leakage.

The primary focus of this research project would be to find out the effectiveness of identifying anomalies in insider behavior using behavioral attributes extracted from users' logs and the comparison between the supervised and unsupervised techniques of machine learning for the task in question. The particular focus would be on finding out the efficacy of using temporal, behavioral, and historic behavioral attributes to detect the malicious user-days despite extreme class imbalance.

The main contributions of this study are as follows:

- A user-day behavioral representation of CERT r4.2 activity logs is developed by integrating logon, device, file, and email events into a unified feature space.
- A 49-feature behavioral representation is engineered to capture activity volume, computer usage, temporal characteristics, email behavior, and deviations from historical user behavior.
- Three different learning paradigms are comparatively evaluated: Isolation Forest for unsupervised anomaly detection, Random Forest for supervised classification, and XGBoost for supervised classification.
- The models are evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC, with particular emphasis on metrics suitable for highly imbalanced insider threat data.
- The study analyzes the effect of behavioral feature representation on detection performance and demonstrates the importance of combining activity-based, temporal, and historical behavioral information for insider threat detection.

Overall, this research focuses on the relationship between behavioral feature representation and machine learning model performance in insider threat detection. Rather than relying solely on overall classification accuracy, the study emphasizes the ability of models to identify the relatively rare malicious behaviors within a large volume of legitimate organizational activity. The comparative evaluation provides insights into the strengths and limitations of supervised and unsupervised approaches and identifies XGBoost with the proposed behavioral representation as the strongest performing approach in the experimental setting.

2 Related Work

Insider threat detection has been well researched through the application of rule-based approaches, user behavior analysis, machine learning, and anomaly detection techniques. Insiders possess legitimate access to organizational assets and therefore malicious insider behaviors can mimic typical behavior patterns, thereby complicating detection [1, 2].

User Behavior Analytics (UBA) models user activity to identify deviations from normal behavior. Kim et al. used daily user activity and communication patterns for anomaly-based insider threat detection [2]. Hall et al. used daily user summaries from the CERT Insider Threat Dataset and evaluated multiple machine learning classifiers for detecting malicious insider scenarios [3]. Bin Sarhan and Altwaijry also investigated machine learning and anomaly detection methods using CERT data and highlighted the challenge of severe class imbalance [4].

Both supervised and unsupervised approaches have been applied to CERT. Isolation Forest has been used for unsupervised anomaly detection, while Random Forest and XGBoost have been evaluated as supervised classifiers [4, 5]. Muñoz et al. compared several supervised and unsupervised methods on CERT and SPEDIA and reported strong performance from XGBoost [5]. Recent studies have also explored sequential models such as BiLSTM and Transformers for capturing temporal user behavior [8, 9].

Class imbalance poses another significant problem with respect to the CERT-based approach since malicious events form a small subset of the total number of events that take place. As a result, accuracy can be deceiving in such situations, making precision, recall, F1 score, ROC-AUC, and PR-AUC critical measures of evaluation [4, 5].

2.1 Research Gap

Although previous studies have evaluated machine learning and anomaly detection methods for insider threats, there is a need for a consistent comparison of supervised and unsupervised models using the same behavioral representation and chronological evaluation setting. This study addresses this gap by developing a 49-feature user-day behavioral representation and comparing Isolation Forest, Random Forest, and XGBoost on CERT r4.2. The study also examines the contribution of temporal, activity-based, email, computer-usage, and historical behavioral features under severe class imbalance.

3 Problem Statement and Research Objectives

3.1 Problem Statement

Insider threat detection is challenging because malicious users can operate using legitimate credentials, making their activities difficult to distinguish from normal organizational behavior [1, 2]. The problem becomes more difficult in the CERT Insider Threat Dataset because malicious activities form a very small proportion of the overall user activity, resulting in severe class imbalance [4, 5]. Although both supervised classification and unsupervised anomaly detection methods have been investigated for insider threat detection, their effectiveness can vary depending on how user behavior is represented and how the models are evaluated [2, 5].

Therefore, this study investigates whether a user-day behavioral representation based on activity, temporal, computer-usage, email, and historical behavioral features can effectively identify insider-threat anomalies. The study specifically compares three machine-learning approaches—Isolation Forest, Random Forest, and XGBoost—using the CERT Insider Threat Dataset r4.2.

3.2 Research Objectives

Key goals of this study include:

1. Extracting user-day behavioral profiles based on logon, device, file, and email activity logs of CERT Insider Threat Dataset r4.2.
2. Building a 49-feature behavioral profile based on activity volume, computer usage, temporal behavior, email activity, and deviation from previous user behavior.
3. Comparing Isolation Forest, Random Forest, and XGBoost algorithms for insider threat detection.
4. Assessing the effectiveness of different machine learning algorithms through measures like accuracy, precision, recall, F1 score, ROC AUC, and PR AUC, especially with focus on metrics designed for class imbalance problem.
5. Studying the influence of behavioral feature richness on performance of insider threat detection algorithms.

4 Proposed Methodology

4.1 Overall Architecture

The insider threat detection framework developed in this research identifies aberrant corporate behavior by aggregating activity logs generated across diverse organizational assets. Rather than attempting to evaluate isolated system actions—which rarely carry sufficient contextual

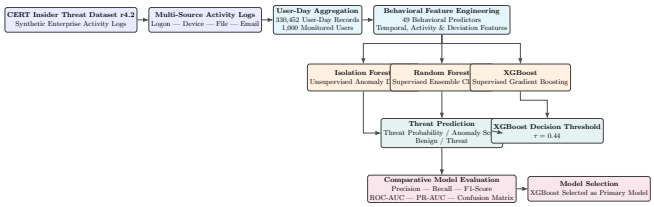


Fig. 1. Model-focused architecture of the proposed insider threat detection framework.

Table 1. Distribution of normalized CERT r4.2 audit events

Event Source	Events	Percentage
Logon	854,859	19.72%
Device	405,380	9.35%
File	445,581	10.28%
Email	2,629,979	60.65%
Total	**4,335,799**	**100.00%**

evidence to distinguish legitimate tasks from malicious behavior—our approach unifies diverse events into structured daily observations per user. Evaluating telemetry over a full calendar day provides visibility into subtle operational variations and behavioral anomalies that point-in-time checks routinely miss [2, 3].

The analytical pipeline processes audit logs extracted from release 4.2 of the CERT Insider Threat Dataset. We examine four primary telemetry channels: system logons, removable peripheral usage, file operations, and corporate email. Because each source uses its own schema and naming conventions, raw events are first converted to a uniform data structure. These harmonized events are then grouped by employee account and calendar date to construct a unified user-day behavioral dataset [3, 5].

From each aggregated user-day record, we extract forty-nine quantitative behavioral predictors. These variables reflect operational volume, diurnal timing distributions, electronic correspondence, computer switching, and individualized historical deviations from past behavior. The engineered feature matrices are provided to three machine learning models: an unsupervised Isolation Forest baseline, a standard supervised Random Forest, and an Extreme Gradient Boosting (XGBoost) model. XGBoost serves as the primary detection algorithm due to its superior discriminative capacity on both validation and holdout evaluation sets. Previous CERT-based studies have similarly evaluated supervised and unsupervised learning approaches, including Random Forest, Isolation Forest, and XGBoost [4, 5].

To ensure the model's outputs can be interpreted by security teams, we integrate the trained XGBoost model with SHAP (SHapley Additive exPlanations). This attribution mechanism calculates the local marginal impact of each behavioral feature, identifying the key indicators that drove a given alert. Both the inference pipeline and explanation module are hosted within an asynchronous FastAPI service, enabling direct integration with an operational security operations dashboard. The use of SHAP for interpreting machine learning-based insider threat detection is also supported by recent research on trustworthy cybersecurity and insider threat classification [10].

4.2 Data Preprocessing

Raw system logs in the CERT archive feature distinct schemas and domain-specific attributes. Directly merging these streams without reconciliation produces misaligned and missing attributes. The proposed pipeline standardizes four telemetry streams: authentication, peripheral device connections, file system manipulations, and email communications. Web proxy (HTTP) records were intentionally excluded from this study, as their disproportionately large volume introduces memory and compute overheads that outweigh their marginal benefit in this baseline design.

The first preprocessing stage loads each source file and converts proprietary field names into a unified attribute schema. System identifiers are mapped to an overall event identifier, event timestamps are converted to standardized UTC datetimes, and host workstations are designated as computers. For messaging logs, sender addresses, recipient arrays, byte footprints, and attachment counts are similarly mapped to uniform fields, while operational actions such as connection or disconnection map directly to action fields.

In addition to schema mapping, each record is tagged with an explicit event type identifier corresponding to its source domain—logon, device, file, or email. Once normalized, the four channels are combined into a single chronologically sorted corpus of 4,335,799 event records. Preserving this temporal sequence is critical to preventing lookahead bias when constructing historical baselines in subsequent stages.

Within this unified dataset, email logs account for the largest share with 2,629,979 events (roughly 60.65% of the total). Authentication logs contribute 854,859 events (19.72%), while file operations and peripheral device connections represent 445,581 events (10.28%) and 405,380 events (9.35%) respectively. This harmonization process yields a unified audit sequence ready for user-level daily aggregation. The use of heterogeneous CERT audit sources and their transformation into structured behavioral representations is consistent with previous CERT-based insider threat studies [3, 5].

4.3 User-Day Behavioral Representation

Individual audit events rarely carry definitive evidence of an insider threat on their own. Inserting an external storage device, accessing remote files, or authenticating outside normal business hours can all reflect legitimate operational duties depending on an employee's responsibilities. To establish appropriate context, our framework adopts the user-day as its core analytical unit. Daily user-level behavioral representations have previously been used for insider threat detection on the CERT dataset [2, 3].

Formally, for any given user u on calendar date d, the event set $D_{u,d}$ comprises all events e_i such that the user associated with e_i matches u and the calendar date of e_i matches d:

$$D_{u,d} = \{e_i \mid user(e_i) = u,\ date(e_i) = d\}. \tag{1}$$

From this collection of daily events, summary metrics are computed across the user's activities. These include total event volume, logon attempts, device attachments, file touches, emails sent, unique workstations accessed, and actions occurring during off-hours.

Combining these baseline activity counts with temporal, email, endpoint, and historical deviation metrics transforms the original 4,335,799 raw log records into 330,452 user-day behavioral instances. Each record contains forty-nine predictive features along with the corresponding user identifier, calendar date, and ground-truth threat label. This aggregation abstracts high-frequency event noise into clear, daily behavioral profiles, consistent with the behavioral aggregation strategies investigated in previous insider threat research [2, 3, 5].

Table 2. Behavioral feature categories used in the proposed model

Category	Representative Features
Activity Volume	Total events, logon attempts, device interactions, file touches, emails sent.
Temporal Behavior	First/last activity hour, active hours, weekend and late-night activity, working-hours ratios.
Email Behavior	Total byte volume, average email size, attachment count, unique recipients/senders, oversized emails.
Computer Usage	Distinct workstations accessed, host-switching frequency, machine deviation from baseline.
Historical Deviation	Historical means and current-to-historical ratios across events, off-hours, emails, and endpoints.
Total	**49 behavioral predictors**

4.4 Feature Engineering

Insider threats seldom manifest as isolated, conspicuous spikes. Instead, they typically appear as subtle, compound changes in daily routine, working hours, communications, workstation usage, and departures from past habits. We engineered forty-nine features organized into five distinct behavioral categories. The use of activity, temporal, communication, and historical behavioral characteristics is consistent with existing user behavior modeling approaches for insider threat detection [2, 5].

Activity volume features capture the employee's aggregate throughput across operational channels. These metrics include total logged events, logon attempts, peripheral device interactions, file modifications, and email transmissions. Together, they establish an individual's gross daily activity level.

Temporal behavior features capture when activity occurs throughout the day. Insider activity frequently shifts outside normal working hours to avoid direct workplace observation. Extracted temporal indicators include the hour of first activity, the hour of last activity, total active hours, weekend event counts, late-night events occurring between 22:00 and 06:00, and standard working-hours events logged between 08:00 and 18:00 on weekdays. We also derive the corresponding ratios for late-night, weekend, and working-hours activity relative to daily volume.

Email behavioral features characterize corporate messaging habits beyond raw transmission counts. Data exfiltration often relies on outbound electronic messaging. The extracted predictors include total email byte volume, average email size, cumulative attachment count, number of unique recipients, number of unique senders, total messages containing attachments, count of oversized transmissions, recipient diversity, attachment ratio, and large-email ratio.

Computer usage features track workstation access patterns and detect potential lateral movement across internal endpoints. These variables quantify the total number of distinct workstations accessed during the day, the frequency of host switching, historical endpoint baselines, and deviations from an employee's typical assigned machine.

Historical behavioral features evaluate whether the current day's activity aligns with the user's historical habits. Moving baseline means are calculated across several operational dimensions, including overall event counts, emails dispatched, file touches, logon events, device connections, off-hours actions, weekend work, late-night tasks, and workstation diversity. Comparing current metrics against these historical baselines provides personalized context that population-wide thresholds cannot capture [2].

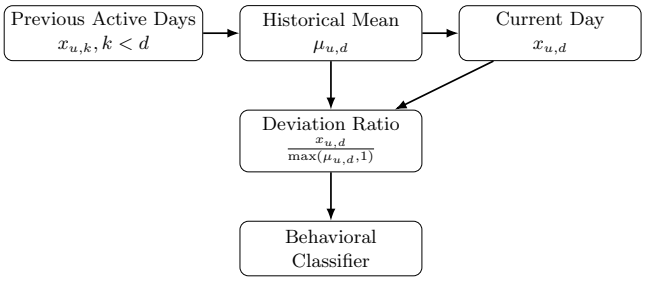


Fig. 2. User-specific historical baseline and behavioral deviation computation.

4.5 Historical Behavioral Deviation

Normal baseline activity varies widely across an enterprise; a database administrator or developer legitimately generates far more system activity than an administrative staff member. Relying strictly on fixed, organization-wide thresholds will generate frequent false alarms for active roles while missing anomalous behavior from employees with low baseline volume. User-specific behavioral modeling has therefore been investigated as a means of providing individualized context for anomaly detection [2].

To resolve this issue, the proposed pipeline tracks individual behavioral baselines. For any user u on calendar day d, we calculate the historical mean $\mu_{u,d}$ for each feature x across all preceding active days $k < d$:

$$\mu_{u,d} = \frac{1}{N} \sum_{k<d} x_{u,k} \tag{2}$$

where N represents the count of previously observed active days for that user. The behavioral deviation ratio is then calculated as:

$$Deviation_{u,d} = \frac{x_{u,d}}{\max(\mu_{u,d}, 1)} \tag{3}$$

Restricting the calculation to $k < d$ prevents the current day's activity from influencing its own historical reference, avoiding data leakage. The denominator floor of 1 protects against division-by-zero errors when past activity is zero. For example, if an employee historically averages 10 emails per day and suddenly dispatches 80, the deviation ratio will register an eight-fold increase (8.0). This signals an anomalous shift regardless of where that volume falls relative to the broader organization. The use of historical user behavior as a personalized reference is consistent with prior user behavior modeling approaches for insider threat detection [2].

5 Dataset and Experimental Setup

5.1 CERT Dataset

Empirical evaluations are conducted using version 4.2 of the CERT Insider Threat Dataset, an established benchmark offering synthetic enterprise audit logs designed for insider threat research. The corpus contains machine-generated audit records covering user authentications, workstation access, file manipulation, and electronic messaging across 1,000 corporate users and 1,003 workstations over an observation window extending from January 2, 2010 to May 17, 2011. CERT r4.2 has been widely used in previous insider threat detection studies for evaluating machine learning, behavioral modeling, and anomaly detection techniques [3–5].

Our pipeline processes four key telemetry sources: logon, device, file, and email activity. These channels cover the primary vectors of insider risk—unauthorized access, peripheral data transfer, file staging, and external communications. HTTP proxy logs are omitted due to their large data footprint and processing overhead. The resulting dataset condenses 4,335,799 raw events into 330,452 user-day records, each described by forty-nine behavioral predictors.

5.2 Ground Truth Generation

The CERT r4.2 release includes scenario documentation identifying seventy recorded insider threat incidents mapped across three specific threat profiles: thirty instances of data exfiltration via unauthorized removable storage (Scenario 1), thirty instances of intellectual property theft and unauthorized data staging (Scenario 2), and ten instances of privilege escalation and IT system sabotage (Scenario 3).

Each ground-truth incident records the compromised user account alongside explicit start and end timestamps, denoted as $[t_{start}, t_{end}]$, demarcating the active malicious window. Ground-truth labels are first applied at the raw event level: an individual event e_i is labeled as malicious ($y_i = 1$) if and only if its timestamp falls within the active incident window for that specific user:

$$y_i = \begin{cases} 1, & t_{start} \le t_i \le t_{end}, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Activity generated by an eventual insider outside this active window is treated as benign ($y_i = 0$). When individual events are grouped into daily summaries, any user-day containing at least one event where $y_i = 1$ is marked as positive. This preserves realistic operational noise by ensuring an insider's benign daily activities are not artificially separated from their malicious actions.

Table 3. CERT r4.2 insider threat scenarios

Scenario	Threat Profile	Incidents
Scenario 1	Removable-media data exfiltration	30
Scenario 2	Unauthorized access / IP theft	30
Scenario 3	Privilege misuse / IT sabotage	10
Total		**70**

Table 4. Class distribution before and after user-day aggregation

Level	Benign Count	Malicious Count
Event level	4,311,759 (99.45%)	24,040 (0.55%)
User-day level	329,088 (99.59%)	1,364 (0.41%)
Total user-days		**330,452**

5.3 Class Distribution

Like enterprise security environments, the dataset exhibits severe class imbalance, with malicious actions representing a small fraction of total activity. At the event level, the corpus contains 4,311,759 benign events (99.45%) and 24,040 malicious events (0.55%). Severe class imbalance is a recurring challenge in CERT-based insider threat research and can substantially affect classification performance [4–6].

Aggregating these events into daily summaries yields 329,088 benign user-days (99.59%) and 1,364 malicious user-days (0.41%) out of 330,452 total records. This low positive rate means standard classification accuracy is an uninformative metric; a naive model predicting only the majority class would achieve over 99.58% accuracy while missing every insider threat. Consequently, model performance must be evaluated using Precision, Recall, F1-Score, Area Under the Receiver Operating Characteristic (ROC-AUC), Precision-Recall AUC (PR-AUC), and complete confusion matrices. The importance of evaluating insider threat models using measures beyond accuracy has been highlighted in previous studies addressing highly imbalanced security data [4, 5].

5.4 Temporal Train/Validation/Test Split

Standard random k-fold cross-validation is unsuitable for time-series security data because it introduces lookahead bias, allowing models to train on future records while testing on past records from the same campaign. To preserve temporal validity, the 330,452 user-day records are ordered chronologically and divided into non-overlapping training, validation, and holdout test partitions.

The training partition spans all activity recorded prior to November 18, 2010, encompassing 221,306 user-day records and 852 positive threat instances. The validation partition covers November 18, 2010 through February 5, 2011, containing 47,673 user-day records and 268 malicious instances; this partition is reserved for hyperparameter tuning and decision threshold optimization. Finally, the test partition covers February 6, 2011 through May 17, 2011, comprising 61,473 user-day records and 244 malicious instances. This holdout set remains untouched throughout development and is used strictly for final model evaluation.

Table 5. Chronological train, validation, and test partitions

Partition	Date Range	Records	Positive
Training	Before 18 Nov 2010	221,306	852
Validation	18 Nov 2010–05 Feb 2011	47,673	268
Test	06 Feb 2011–17 May 2011	61,473	244
Total		**330,452**	**1,364**

Table 6. Machine learning model configurations

Model	Configuration Details
Isolation Forest	Unsupervised anomaly detection; trained strictly on benign baseline feature vectors.
Random Forest	300 bagged trees; class weights adjusted inversely proportional to label frequencies.
XGBoost	300 trees; max depth 6; $\eta = 0.05$; subsample 0.80; column subsample 0.80; pos-weight 258.75; threshold 0.44.

5.5 Experimental Configuration

Our evaluation examines three machine learning algorithms: an unsupervised Isolation Forest baseline alongside supervised Random Forest and XGBoost classifiers. The primary model, XGBoost, is trained on the forty-nine engineered behavioral features. To mitigate the severe class imbalance, we apply an internal positive-class loss weight of 258.7488, matching the ratio of benign to malicious training instances ($220,454/852$).

Model hyperparameters were tuned via validation tracking: we used 300 boosting trees, a maximum tree depth of 6, a learning rate of 0.05, a row subsampling ratio of 0.8, and a column subsampling ratio of 0.8, optimized against binary logistic loss using PR-AUC as the primary metric. Rather than defaulting to a standard 0.5 decision threshold, the validation set was used to set an optimal classification threshold of 0.44. This threshold was then applied directly to the holdout test set.

6 Machine Learning Models

6.1 Isolation Forest

Isolation Forest serves as an unsupervised baseline to evaluate whether anomalous user-day behavior can be detected without training on known malicious labels. The algorithm isolates unusual data points by randomly choosing a feature and selecting a split value between that feature's minimum and maximum. Because anomalies are sparse and distinct, they require fewer random partitions to isolate, resulting in noticeably shorter average path lengths within the generated isolation trees. Isolation Forest and related unsupervised methods have previously been evaluated for insider threat detection on CERT data [4, 5].

In this experiment, the Isolation Forest is trained exclusively on benign instances from the training partition. Anomaly scores are then computed across the validation and test partitions. While Isolation Forest does not require labeled attack examples, it cannot easily distinguish between benign operational spikes (such as system maintenance) and genuine insider threats. This tendency toward higher false positive rates makes it a valuable baseline against which to evaluate supervised alternatives [5].

6.2 Random Forest

Random Forest serves as a strong supervised baseline, capable of capturing nonlinear relationships across our forty-nine behavioral features without making strong parametric assumptions. The algorithm constructs an ensemble of 300 bagged decision trees, with each tree trained on a bootstrap sample of the training data and evaluating a random subset of features at each candidate split. Individual tree predictions are then combined to generate the final classification. Random Forest has previously been evaluated as a supervised approach for insider threat detection using CERT-based behavioral data [3–5].

Class weights are adjusted inversely proportional to class frequencies to prevent the ensemble from skewing toward the dominant benign class. While Random Forest handles complex feature interactions well, its independent bagging approach does not iteratively correct errors across iterations, presenting a useful comparison with the sequential boosting strategy of XGBoost.

6.3 XGBoost

XGBoost is selected as the primary classification engine for the proposed framework. Unlike bagging methods that train trees independently, gradient boosting constructs decision trees sequentially. Each successive tree fits directly to the pseudo-residuals of the existing ensemble, progressively refining decisions near complex class boundaries. Previous research using CERT-based insider threat datasets has reported strong performance from XGBoost among supervised approaches [5].

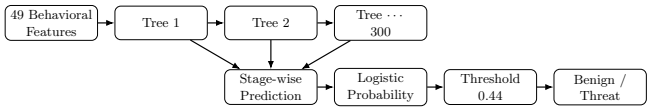


Fig. 3. XGBoost-based insider threat prediction pipeline.

The production configuration uses 300 boosting rounds, a maximum depth of 6, and a learning rate of 0.05. Both row subsampling and column feature subsampling are set to 0.8 to introduce stochastic regularization and prevent overfitting. To account for the heavy class imbalance, we set the positive class weight to 258.7488, scaling the gradient updates of the rare positive class accordingly.

For an input user-day vector X, the model produces an estimated threat probability $P(y = 1 \mid X)$. Applying the validation-tuned threshold ($\tau = 0.44$) converts this probability into a binary classification:

$$\hat{y} = \begin{cases} 1, & P(y = 1 \mid X) \geq 0.44, \\ 0, & P(y = 1 \mid X) < 0.44. \end{cases} \tag{5}$$

This probability is also mapped to a 0-to-100 continuous risk score for the deployment layer:

$$RiskScore = 100 \times P(y = 1 \mid X) \tag{6}$$

This dual output provides both a discrete threat classification and a granular risk score for alert prioritization.

Finally, we integrate TreeSHAP with the trained XGBoost model to provide feature-level explanations for security analysts. Rather than outputting an unexplained risk score, SHAP decomposes individual predictions into localized feature attributions. This highlights the specific behavioral shifts—such as off-hours file touches or spikes in email attachments—that drove the alert, providing analysts with the context needed to triage and investigate potential insider threats. Recent work has similarly investigated SHAP-based explanations for XGBoost insider threat detection to improve model interpretability and analyst trust [10].

6.4 XGBoost Evaluation Results

The final XGBoost model was evaluated on the temporally held-out test partition. The model achieved an accuracy of 99.78%, precision of 79.23%, recall of 59.43%, and F1-score of 67.92%. The ROC-AUC was 99.57%, while the PR-AUC reached 78.97%.

The high ROC-AUC demonstrates strong overall separation between benign and malicious user-day observations. However, the lower recall compared with precision

Table 7. XGBoost performance on the holdout test set

Metric	Score
Accuracy	99.78%
Precision	79.23%
Recall	59.43%
F1-Score	67.92%
ROC-AUC	99.57%
PR-AUC	78.97%
Decision Threshold	0.44

indicates that some malicious user-day instances remain difficult to identify. This behavior is expected in a highly imbalanced insider-threat setting, where aggressive increases in sensitivity can result in substantially more false positives.

7 Results and Evaluation

The CERT Insider Threat Dataset r4.2 has been used for comparing the performance of three algorithms: Isolation Forest, Random Forest, and XGBoost. The comparison is based on user-day behavioral characteristics obtained from logon, device, file, and email events. Because of the highly imbalanced nature of the dataset, accuracy alone cannot be used to measure the performance of the algorithms for insider-threat detection. Therefore, the models were evaluated using accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC [4, 5].

7.1 Evaluation of Models

The evaluation shows clear differences between the three models. Isolation Forest achieved an accuracy of 76.52% and a recall of 57.38%, but its precision was only 0.97% and its F1-score was 1.90%. Although the model was able to identify a considerable number of abnormal cases, many of these cases were actually normal activities. As a result, the model produced a large number of false positives. Similar problems with unsupervised anomaly detection techniques have been reported in previous studies using CERT data [4, 5].

Random Forest achieved an accuracy of 99.48%, with a precision of 24.68%, recall of 15.57%, and F1-score of 19.10%. The low recall means that many malicious cases were not successfully identified despite the high accuracy of the model. This result shows the limitation of relying only on accuracy when the data is highly imbalanced [3, 4].

The best-performing model was XGBoost using the complete 49-feature representation. It achieved an accuracy of 99.78%, precision of 79.23%, recall of 59.43%, and F1-score of 67.92%. It also achieved a ROC-AUC of 99.57% and a PR-AUC of 78.97%. Compared with Isolation Forest and Random Forest, XGBoost provided better performance in detecting malicious activities while producing fewer false positives. These results are consistent with previous studies that have reported the effectiveness of XGBoost for insider-threat detection using CERT data [5].

Table 8. Performance comparison of the evaluated models (%)

Model	Acc	Prec	Rec	F1	ROC	PR
Isolation Forest	76.52	0.97	57.38	1.90	74.77	0.78
Random Forest	99.48	24.68	15.57	19.10	89.30	11.09
XGBoost (11 feat.)	99.23	26.00	50.82	34.40	95.21	24.48
XGBoost (49 feat.)	**99.78**	**79.23**	**59.43**	**67.92**	**99.57**	**78.97**

7.2 Effect of Behavioral Features

The results show that the number and quality of behavioral features have an important effect on detection performance. The XGBoost model using 11 features achieved an accuracy of 99.23%, precision of 26.00%, recall of 50.82%, F1-score of 34.40%, ROC-AUC of 95.21%, and PR-AUC of 24.48%.

When the feature set was extended to 49 features, the performance improved considerably. Precision increased from 26.00% to 79.23%, recall increased from 50.82% to 59.43%, and the F1-score increased from 34.40% to 67.92%. ROC-AUC also improved from 95.21% to 99.57%, while PR-AUC increased from 24.48% to 78.97%.

This improvement suggests that using a wider range of behavioral information helps the model distinguish between normal and suspicious user activities. The 49 features include information related to activity frequency, time of activity, computer usage, email behavior, and changes in a user's previous behavior. These findings support previous research that emphasizes user behavior modeling and feature engineering for insider-threat detection [2, 3].

7.3 Evaluation Under Class Imbalance

The results also show the effect of class imbalance on model performance. Random Forest achieved an accuracy of 99.48%, but its recall was only 15.57%. This shows that the high accuracy was influenced by the large number of normal observations in the dataset. Therefore, a high accuracy score does not necessarily mean that the model can detect most malicious activities.

Isolation Forest achieved a recall of 57.38%, but its precision was only 0.97%. This indicates that a significant

number of normal observations were incorrectly classified as anomalies.

The 49-feature XGBoost model performed better in this situation. It achieved a precision of 79.23% and a recall of 59.43%, resulting in an F1-score of 67.92%. The PR-AUC of 78.97% was also considerably higher than the values obtained by the other models. This shows that XGBoost was better able to identify the minority malicious class while producing fewer false-positive cases. This is important for insider-threat detection because a large number of false alarms can make it more difficult to identify real threats [4, 5].

7.4 Comparative Analysis

The findings show different levels of performance for each model on the CERT insider-threat dataset. Isolation Forest can identify unusual activities without depending on labeled data, but its low precision reduces its usefulness when false positives need to be controlled. Random Forest achieved high accuracy but performed poorly in identifying the minority malicious class, as shown by its low recall and F1-score.

XGBoost performed better than the other models, particularly when the 49-feature representation was used. The comparison also shows that improving the feature set can be as important as selecting the machine-learning algorithm.

The 49-feature representation combines different types of information, including behavioral activity, temporal data, computer usage, email activity, and historical changes in user behavior. Using these features together provides more information about the user's current activity in relation to their previous behavior. Similar importance has been given to behavioral representation and feature engineering in previous CERT-based studies [2, 4, 5].

Based on the overall results, XGBoost with the 49-feature representation was the best-performing model in this study. It achieved the highest precision, F1-score, ROC-AUC, and PR-AUC, while maintaining a recall of 59.43%. The results suggest that combining detailed user behavior with supervised boosting can be useful for detecting insider-threat anomalies.

8 Discussion

The models used in this study produced different results on the CERT Insider Threat Dataset r4.2. Although Random Forest achieved an accuracy of 99.48%, its recall was only 15.57%. This shows that accuracy should not be used as the only important metric when evaluating insider-threat detection on an imbalanced dataset [3, 4].

Isolation Forest achieved a better recall of 57.38%, but its precision was only 0.97%. This means that many normal activities were classified as anomalies, resulting in a large number of false alerts. This is one of the main challenges of using unsupervised anomaly detection for insider-threat data [4, 5].

XGBoost was more effective than the other models. The 49-feature model achieved 79.23% precision, 59.43% recall, 67.92% F1-score, and 78.97% PR-AUC. These results show a better balance between detecting malicious activities and reducing false alarms, which is also supported by earlier research using CERT data [5].

The comparison between the 11-feature and 49-feature XGBoost models also shows the importance of behavioral features. After adding more features, precision increased from 26.00% to 79.23%, while F1-score increased from 34.40% to 67.92%. This improvement suggests that information about activity frequency, time, computer usage, email activity, and changes in previous behavior can help the model recognize suspicious activity more effectively [2, 3].

Based on the results, XGBoost with 49 features was the most effective model in this study. However, its recall of 59.43% also shows that some malicious activities were still missed. Future work should therefore focus on improving recall while maintaining the precision achieved by the current model.

9 Conclusion and Future Work

9.1 Conclusion

The main goal of this research was to detect insider-threat cases based on user behavioral activity using the CERT Insider Threat Dataset r4.2. The activity logs were transformed into user-day records using 49 behavioral characteristics, including activity, time, computer usage, email behavior, and changes in previous user behavior.

Three machine-learning techniques were examined: Isolation Forest, Random Forest, and XGBoost. The results showed that Isolation Forest was able to detect anomalous activity but produced a large number of false positives. Random Forest achieved very high accuracy, but its ability to detect malicious activity was limited by its low recall. XGBoost performed better than both approaches, particularly when the 49 behavioral features were used.

The 49-feature XGBoost model achieved 99.78% accuracy, 79.23% precision, 59.43% recall, 67.92% F1-score, 99.57% ROC-AUC, and 78.97% PR-AUC. The comparison also showed that using a larger and more detailed set of behavioral features improved the detection results. Therefore, the findings suggest that detailed user behavior combined with XGBoost can be a useful approach for insider-threat detection.

9.2 Future Work

There are several areas that can be explored in future research. The current model can be improved to increase recall so that more malicious activities can be detected while keeping the false-positive rate under control. Further feature-selection and feature-engineering techniques can also be investigated to identify the most useful behavioral characteristics.

Future work can also involve testing the proposed approach on real organizational data or other insider-threat datasets to examine how well it performs outside the CERT environment. More advanced temporal and deep-learning models can be compared with the current approaches to capture changes in user behavior over longer periods.

Finally, explainable AI techniques such as SHAP can be explored further to provide security analysts with a clearer understanding of why a particular user-day record was classified as suspicious. This can make the detection system more useful for practical security investigations.

References

1. A. Sanzgiri and D. Dasgupta, "Classification of Insider Threat Detection Techniques," in *Proceedings of the 2016 ACM Cyber and Information Security Research Conference (CISR)*, 2016.
2. J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, "Insider Threat Detection Based on User Behavior Modeling and Anomaly Detection Algorithms," *Applied Sciences*, vol. 9, no. 19, p. 4018, 2019.
3. A. J. Hall, N. Pitropakis, W. J. Buchanan, and N. Moradpoor, "Predicting Malicious Insider Threat Scenarios Using Organizational Data and a Heterogeneous Stack-Classifier," arXiv preprint arXiv:1907.10272, 2019.
4. B. Bin Sarhan and N. Altwaijry, "Insider Threat Detection Using Machine Learning Approach," *Applied Sciences*, vol. 13, no. 1, p. 259, 2023. doi: 10.3390/app13010259.
5. A. M. Muñoz, X. Larriva-Novo, C. Sánchez-Zas, M. Alvarez-Campana, and V. A. Villagra, "A Session-Based Insider Threat Detection Approach Leveraging the CERT and SPEDIA Datasets," *Machine Learning with Applications*, vol. 25, p. 100946, 2026. doi: 10.1016/j.mlwa.2026.100946.
6. D. Álvarez Muñiz, L. Perez Miguel, A. M. Muñoz, X. Larriva-Novo, M. Alvarez-Campana, and D. Rivera, "Design and Generation of a Dataset for Training Insider Threat Prevention and Detection Models: The SPEDIA Dataset," *Computers & Security*, vol. 161, p. 104743, 2026. doi: 10.1016/j.cose.2025.104743.
7. Z. Wei, U. Rauf, and F. Mohsen, "E-Watcher: Insider Threat Monitoring and Detection for Enhanced Security," *Annals of Telecommunications*, vol. 79, pp. 819–831, 2024.
8. P. Manoharan, W. Hong, J. Yin, H. Wang, Y. Zhang, and W. Ye, "Optimising Insider Threat Prediction: Exploring BiLSTM Networks and Sequential Features," *Data Science and Engineering*, vol. 9, pp. 393–408, 2024.
9. M. Elbasheer and A. Akinfaderin, "User-Based Sequential Modeling with Transformer Encoders for Insider Threat Detection," arXiv preprint arXiv:2506.23446, 2025.
10. R. Faqihi, P. Nanda, M. Mohanty, S. Alqahtani, and B. Alrashed, "Towards Trustworthy Cybersecurity: Reclassifying Insider Threat Detection Using SHAP," *Future Generation Computer Systems*, vol. 183, p. 108543, 2026.
11. A. K. Mahato, O. P. Mahato, R. Pokhrel, and K. Mahato, "User Behavior Analytics for Insider Threat using Transformer Based Approach," *International Journal of Computer Science and Network Technology*, vol. 15, no. 1, pp. 33–43, 2026. doi: 10.18486/ijcsnt/15.1.002.