Profiling User Behavior to Identify Insider Threats in Enterprise Information Systems

Shantabhushana. B M[1], *Sudarsanan.* D[2], *Praveen* Kumar[3], *Lokesh*[4], *Mithu*n. B M[5], *Sandeep* Kumar. S[6], and *Rakesh.* V S[4*]

[1]Department of Computer Science and Engineering, A.G.M. Rural College of Engineering and Technology, Dharwad, India
[2]Department of Information Science and Engineering, Cambridge Institute of Technology, Bengaluru, India
[3]Department of Computer Science and Engineering, Global Academy of Technology, Bengaluru, India
[4]Department of Computer Science and Engineering, Cambridge Institute of Technology, Bengaluru, India
[5]Department of Information Science and Engineering, A P S College of Engineering, Bengaluru, India
[6]Department of Computer and Communication Engineering, NMAM Institute of Technology, Nitte (Deemed to be University), Nitte, Udupi, India

Abstract. Insider threats pose a significant challenge to enterprise information systems due to their subtle and context-dependent nature. Unlike external attacks, these threats emerge from authorized users whose behavior gradually deviates from established norms. This work presents a lightweight, interpretable framework for detecting insider threats through user behavior profiling. Session-based features such as login variability, off-hours activity, file access diversity, and USB bursts are extracted to characterize behavioral deviations over time. The framework employs Isolation Forest and One-Class SVM for anomaly detection, combining their outputs using a weighted score fusion strategy. Experiments were conducted on both a custom-generated synthetic dataset and the publicly available CERT Insider Threat Dataset v6.2. Results show that the fusion-based approach outperforms traditional baselines—including Z-score, Local Outlier Factor, and Autoencoders—achieving an F1-score of 0.89 on synthetic data and 0.83 on CERT, with corresponding AUC scores of 0.94 and 0.89. These findings confirm the effectiveness of combining interpretable features with ensemble anomaly detection in identifying insider risks, while maintaining compatibility with privacy-aware and distributed enterprise environments.

1 Introduction

Insider threats are a growing concern in enterprise cybersecurity, stemming from individuals within the organization who misuse legitimate access, intentionally or unintentionally. Unlike external attacks, these threats often bypass traditional defenses and are frequently hidden within routine operations, making them more challenging to detect. As businesses adopt more cloud-based and distributed infrastructures, static rule-based systems struggle to keep up with evolving user behavior.

To address this, recent approaches focus on behavioral modeling that captures patterns, sequences, and contextual variations in user activity. One emerging technique involves converting log data into structured visual formats, allowing systems to detect subtle anomalies using spatial analysis. These image-like representations offer interpretable insights and are better suited for identifying both isolated and systemic behavioral shifts [1][2].

Detection models are also becoming more scenario-specific, tailored to different threat types, such as privilege misuse, identity theft, or data leakage. These systems analyze communication context and behavioral timelines to catch risks that develop gradually over time [3]. In parallel, privacy-preserving frameworks such as federated learning allows for decentralized model training, helping organizations comply with data regulations while still learning from distributed behavior patterns [4]. This paper builds on these directions by proposing a lightweight and explainable user profiling framework for detecting insider threats, with a focus on practicality, accuracy, and deployment in real-world enterprise environments.

The contribution of this paper includes:

- A behavior profiling framework utilizing interpretable features such as login patterns, file access rates, and off-hours activity.
- Lightweight anomaly detection with Isolation Forest and One-Class SVM, requiring low computational overhead.
- Context-aware enhancement of behavioral vectors using email, file, and web activity data.
- A privacy-preserving design compatible with federated systems for distributed enterprise environments.

The remainder of this paper is organized as follows: Section 2 presents relevant literature on behavior-based insider threat detection. Section 3 describes the proposed methodology, including data preprocessing,

* Corresponding author: rakesh.tech102@gmail.com

feature extraction, and anomaly detection models. Section 4 outlines the experimental setup and evaluation metrics. Section 5 discusses the results and performance analysis. Finally, Section 6 concludes the paper and outlines directions for future research.

2 Related works

Insider threat detection research has evolved to include organizational, technical, and psychological aspects. A blend of systematic reviews, behavior modeling, and real-time detection frameworks contributes to the growing body of work in this domain.

A study on SMEs highlights that insider threat preparedness is closely tied to leadership involvement, awareness programs, and perceived cyber risks, emphasizing the importance of organizational culture in threat mitigation [5]. Behavior-based detection systems classify user activities—such as logins, USB access, and emails—into logical categories and utilize ML/DL classifiers, enhancing accuracy and clarity through domain-specific modeling and data balancing methods like SMOTE [6]. In federal systems, real-time frameworks integrate Isolation Forest, LSTM Autoencoders, and graph-based models within a standards-compliant structure. These systems provide rapid, scalable detection suited for high-risk environments [7].

A systematic review of CPS-focused insider threat detection identifies three main methods—specification-based, cryptographic, and ML-based—but notes that many existing models lack flexibility to adapt to changing behaviors in dynamic CPS environments [8]. A taxonomy-driven framework categorizes insider threats across CIA dimensions and suggests a multi-layered prevention model, synthesizing insights from 83 studies to highlight key research gaps and strategic directions [9]. The CPID framework creates cyber-personas by analyzing user behavior with semantic and statistical features. It employs a two-tier detection system to identify anomalies against learned profiles, providing high accuracy but depending on detailed logs and a complex architecture [10].

Image-based methods transform activity logs into structured visual formats to detect behavioral anomalies using spatial pattern analysis. These approaches enhance interpretability and precision but require large training sets and involve image-processing overhead [11]. A Siamese neural network approach compares behavior pairs to learn a similarity metric, enabling the detection of subtle anomalies through contrastive learning. It performs well with limited labeled data but demands careful pair construction and is computationally intensive [12]. Hybrid statistical-sequential models combine user profile baselines with sequence modeling (e.g., Hidden Markov Models, HMMs) to capture temporal deviations. These frameworks strike a balance between real-time detection and interpretability, although scalability may suffer with complex sequences [13].

In IoT environments, federated learning enables private, distributed model training across edge devices. These systems incorporate interpretable rule sets and preserve data privacy, but are domain-specific and less transferable beyond IoT contexts [14]. The LaAeb framework combines structured logs with unstructured text (such as emails and commands) to detect threats, enriching the detection with contextual language cues. It improves detection accuracy and reduces false positives, but demands complex preprocessing and dual data availability [15].

The reviewed literature highlights the growing diversity of techniques used for insider threat detection, ranging from cyber persona modeling and visual encoding to sequence learning and federated architectures. A comparative summary of objectives, strengths, and limitations across these methods is presented. Table 1 presents a comprehensive overview of current research directions and existing gaps.

Table 1. Comparative summary of insider threat detection approaches.

Ref. No	Objective	Strengths	Limitations
[5]	Examine SME readiness and response to insider threat risks	Organizational insight: bridges the tech-policy gap	No ML/detection implementation
[6]	Categorize behavioral features and apply ML/DL to detect insider threats	Structured feature modeling; classifier benchmarking	Evaluation limited to offline CERT dataset
[7]	Propose a real-time, modular detection system for federal insider threat prevention.	Multi-model integration, real-time response, compliance-ready	Domain-specific; may need re-tuning for non-federal contexts
[8]	Review 69 CPS-focused ITP papers and classify detection techniques	Broad survey identifies key research gaps in CPS monitoring	Does not propose/test a new model
[9]	Propose a taxonomy-driven, behavior-focused ITP architecture based on CIA objectives.	Thematic clarity; multi-tier prevention strategy	Lacks experimental validation or dataset-based implementation
[10]	Build cyber-personas from user logs and detect anomalies	Personalized detection, high accuracy	Requires rich log data; complex model layers
[11]	Image-based log pattern recognition for insider detection	Visual, interpretable patterns; high precision	Needs extensive data; image encoding overhead
[12]	Use a Siamese network for similarity-based anomaly detection	Works with few labels; low false positives	Demands pair sampling; compute-heavy
[13]	Blend statistical and sequence modeling using HMMs	Interpretable, fast runtime	Scalability issues with complex logs
[14]	Federated, explainable insider threat detection for IoT	Privacy-friendly, explainable rules at the edge	Limited to IoT setups; generalization concerns

[15]	Fuse logs and text to detect insider threats via context fusion	Context-rich detection, lower false positives	Needs text + logs; heavy preprocessing

While the CERT dataset remains the most widely used benchmark in insider threat detection, recent studies have also explored alternative sources such as IoT environments, healthcare systems, and real-world enterprise case studies. Incorporating such perspectives ensures broader applicability beyond the CERT benchmark.

3 Methodology

This section describes the end-to-end design of a behavior-based insider threat detection framework that operates in both synthetic and real-world enterprise environments. The system architecture integrates interpretable behavioral features with unsupervised anomaly detection models—Isolation Forest and One-Class SVM—and supports deployment in privacy-conscious infrastructures.

3.1 Data sources and modeling

To ensure generalizability and realism, the framework was evaluated using two datasets: The Synthetic Dataset and the Public Dataset, CERT.

3.1.1 Synthetic dataset

A controlled dataset was generated to simulate user behavior in a mid-sized enterprise. Activity logs included login attempts, file operations, USB events, and email communications. Simulated user roles (e.g., Admin, Developer, HR) followed standard behavioral profiles for 30 days, with 10% of users injected with insider-like anomalies (e.g., after-hours access + bulk file transfer).

3.1.2 Public dataset

The CERT Insider Threat Dataset v6.2, created by Carnegie Mellon University's SEI, offers anonymized enterprise activity logs including email, logon, device, and file access events [16]. Although originally published in 2013, this reference is retained because it introduces the CERT Insider Threat Dataset, which remains a widely used benchmark in insider threat detection research. It contains structured logs of approximately 1,000 users over an 18-month period. Malicious insiders and their timelines are labeled, supporting benchmark evaluation.

Each log source was pre-processed into fixed-length time windows (e.g., daily), with per-user behavior aggregated for profile-based modeling.

3.2 Behavioral feature extraction

A total of seven interpretable behavioral features were extracted from each session window to capture different dimensions of user activity. These features were chosen for their ability to reflect temporal regularity, access diversity, and interaction intensity—factors often associated with anomalous insider behavior. The selected features are described in Table 2.

Table 2. Description of extracted behavioral features.

Feature Name	Description
Login Variability	Standard deviation of user login times across recent days
Off-Hours Activity Ratio	Percentage of events occurring outside standard working hours (9AM–6PM)
File Access Diversity	Count of distinct file types or categories accessed within the session
USB Activity Bursts	Detection of abnormal spikes in removable device usage
Email Volume Delta	Deviation in number of sent emails from the user's recent average
Session Duration Entropy	Entropy (variability) in session length across days
Web Access Rarity Score	Frequency-weighted score of URLs accessed, highlighting access to uncommon sites

All features were normalized using min–max scaling. Behavioral baselines were kept for each user to support adaptive thresholding.

3.3 Anomaly detection and alert generation

To identify behavioral deviations indicative of insider threats, two unsupervised anomaly detection models were employed: Isolation Forest (IF) and One-Class SVM (OCSVM). These models operate on the extracted session-level feature vectors and are designed to detect rare or contextually abnormal patterns without requiring labeled data. Their complementary strengths improve detection reliability in enterprise settings where malicious actions are subtle and infrequent.

The IF randomly partitions data to isolate anomalies with fewer splits. Suited for high-dimensional behavior data. Whereas, OCSVM constructs a tight boundary around normal behavior using a radial basis function (RBF) kernel. Both models were trained on baseline (non-anomalous) user behavior. No labeled anomalies were used in training.

To enhance robustness, both models independently scored each behavior vector. A final composite score was computed using a weighted average as shown in Equation (1):

$$S_{final} = \alpha . S_{IF} + (1 - \alpha) . S_{OCSVM} \quad (1)$$

Where $\alpha = 0.6$ (set to 0.6 in experiments to slightly favour Isolation Forest due to its resilience to sparse outliers). S_{IF} is the anomaly score from Isolation Forest. S_{OCSVM} is the anomaly score from One-Class SVM.

This weighted combination of scores from both models is referred to as a score fusion strategy, which aims to leverage the strengths of each detector. An alert was triggered if the final score exceeded the user's

dynamic threshold (computed as baseline mean + 2σ deviation).

3.4 Algorithm for detection pipeline

The main logic of the detection framework is outlined in Algorithm 1. It explains how session-wise feature vectors are processed, scored by both models, and evaluated against dynamic thresholds.

Algorithm 1. Behavior-Based Insider Threat Detection

Input:
- Normalized behavioral feature vectors $F(u, w)$ for each user u and time window w
- Pre-trained Isolation Forest model M_{IF}
- Pre-trained One-Class SVM model M_{OC}
- User-specific dynamic threshold T_u

Output:
- Alert flags for anomalous sessions
- Ranked contributing features per alert

For each user $u \in U$:
  For each time window $w \in W_u$:
    Extract feature vector $F(u, w)$
    Normalize features to $[0, 1]$ scale
    Compute anomaly score from Isolation Forest:
      $S_{IF} = M_{IF}.predict - score(F(u, w))$
    Compute anomaly score from One-Class SVM:
      $S_{OCSVM} = M_{OC}.predict - score(F(u, w))$
    Combine Scores :
      $S_{final} = \alpha.S_{IF} + (1 - \alpha).S_{OCSVM}$
    If $S_{final} > T_u$:
      Raise alert for session w
      Rank contributing features by deviation magnitude
    End If
  End For
End For

In this framework, the parameter α sets the relative weight between the two models and is empirically adjusted to 0.6, emphasizing the general anomaly sensitivity of the Isolation Forest. The threshold T_u is user-specific and calculated as the sum of the historical mean and twice the standard deviation of past behavior scores, enabling the system to adapt dynamically to individual behavioral baselines.

The framework supports federated deployment, where models and thresholds can be trained and used locally at the department or branch level. Only anonymized threat scores are sent to a central security monitor, in line with enterprise privacy policies and to reduce the risk of sensitive log exposure.

4 Experimental setup and evaluation

This section outlines the datasets used, system setup, preprocessing steps, model training parameters, and baseline methods for comparison. The framework was evaluated with both real-world and synthetic data to assess its generalizability, interpretability, and detection performance.

4.1 Datasets

To evaluate the performance of the proposed framework in detecting insider threats, two types of datasets were used: one synthesized to mimic enterprise environments under controlled conditions, and another publicly available for benchmarking against established research.

4.1.1 Synthetic dataset generation

A custom dataset was generated to simulate user activity within a mid-sized enterprise, encompassing four roles: the Admin, the Developer, HR, and Finance. Logs were collected for 100 users over a 30-day period, revealing typical behavioral patterns for each role, such as HR users accessing personnel files and Developers running shell commands.

To simulate the insider threats, the anomalies were introduced in 10% of users. These included after-hours logins, repetitive access to sensitive files, bursts of USB usage, and abnormal email volumes. Logs recorded five event types—login, file, email, web, and USB—along with timestamps and user metadata.

Session-wise windows (daily) were extracted for analysis. Events were generated using structured templates and Python scripts to ensure temporal realism and behavioral variation.

4.1.2 Public dataset – CERT

The CERT Insider Threat Dataset v6.2 is utilized. What makes this dataset especially valuable is that it includes labeled instances of insider threats, where users gradually or suddenly shift from their normal behavior before carrying out malicious actions. These labeled segments support both supervised model benchmarking and unsupervised anomaly detection experiments. The dataset has been widely used in previous research and provides a strong baseline for comparative evaluation.

A comparison of key dataset attributes of the two datasets is shown in Table 3.

Table 3. Overview of the synthetic and CERT insider threat datasets.

Attribute	Synthetic Dataset	CERT Dataset v6.2
Source	Simulated enterprise environment	Carnegie Mellon SEI
Duration	30 days	18 months
Number of users	100	~1000
Total log events	~75,000	>15 million
Event types	Login, file, email, USB, web	Logon, email, file, HTTP, device
Labeled anomalies	Yes (manually injected – 10%)	Yes (multiple insider threat scenarios)
Roles simulated	Admin, Developer, HR, Finance	Multiple user roles (anonymized)
Features extracted	7 (login time, session length, email delta, etc.)	Same features after transformation and normalization

4.2 Preprocessing and feature engineering

Logs from both datasets were pre-processed and segmented into fixed-length daily windows, with each window representing a user session. For each session, the seven behavioral features described in Section 3.2 were extracted. Feature vectors were normalized using per-user min–max scaling to reduce variation across individuals.

In the synthetic dataset, insider labels were assigned based on the injected anomalies, while the CERT dataset provided labeled insider events. Sessions with missing or incomplete log types were excluded from the evaluation. The final modeling dataset consisted of 7-dimensional feature vectors for each user session, which were then passed to the detection models for training and testing.

4.3 System setup and model parameters

The insider threat detection framework was developed in Python and run in a standard computing environment to assess its practicality and reproducibility. Both models—Isolation Forest and One-Class SVM—were trained and tested on the same datasets using identical feature vectors and processing steps to ensure a fair comparison. Table 4 summarizes the software and hardware specifications used during the experiment.

Table 4. Experimental environment configuration.

Component	Specification
Programming Language	Python 3.10
Libraries Used	Scikit-learn 1.3, NumPy, Pandas
System OS	Ubuntu 22.04 LTS (64-bit)
Processor	Intel® Core™ i7-12700H CPU @ 2.30GHz
Memory	16 GB RAM
Execution Mode	Single-threaded (no GPU acceleration)

Following environment setup, both anomaly detection models were configured with lightweight parameters to reflect practical, deployable use in enterprise systems. Each model was trained separately on 70% of the dataset (per source) and tested on the remaining 30%. Both models generated anomaly scores for each session, which were later combined using a weighted average to produce a final detection score. Table 5 presents the complete set of parameters used for each detection model.

Table 5. Model configuration parameters.

Parameter	Isolation Forest	One-Class SVM
Contamination	0.1	0.10 (via nu)
Number of Estimators	100	N/A
Kernel	N/A	Radial Basis Function (RBF)
Gamma	N/A	'scale'
Max Samples	'auto'	N/A
Scoring Output	Anomaly score per session	Anomaly score per session

The selected parameters aim to strike a balance between detection capability and computational efficiency. Their lightweight design makes them suitable for deployment in real-time environments, particularly in distributed enterprise scenarios where centralized computing may be impractical.

4.4 Baselines and evaluation criteria

To evaluate the effectiveness of the proposed detection framework, three commonly used anomaly detection techniques were chosen as baseline methods.

- Z-score Detection: A statistical method that identifies feature values that fall outside a set number of standard deviations from the mean.
- Local Outlier Factor (LOF): A density-based algorithm that finds points with lower local density compared to their neighbors.
- Autoencoder: This is a neural network that aims to reconstruct normal input data. A high reconstruction error may indicate the presence of anomalies.

Each baseline model was trained and tested using the same feature set and session windowing approach as the primary detection models. Hyperparameters for all baselines were tuned through trial runs to ensure fair performance across all models.

To evaluate detection quality, the following metrics were used: Precision, Recall, F1-Score, and Area Under the Receiver Operating Characteristic Curve (AUC).

- Precision: Refers to the proportion of detected anomalies that are genuinely malicious.
- Recall: The percentage of actual anomalies that were correctly identified.
- F1-Score: The harmonic-mean of precision and recall, showing the balance between them.
- AUC: This assesses the model's capability to differentiate between normal and abnormal behavior at various thresholds.

These metrics were calculated separately for both datasets—CERT and synthetic—and for each model to ensure a thorough comparison of performance. While more recent deep learning–based approaches (e.g., GAN-based anomaly detection, Siamese networks) exist, they were excluded from this comparison due to their high computational complexity and limited interpretability. Since the focus of this work is on lightweight, explainable models suitable for enterprise deployment, we restrict our baselines to commonly used, efficient methods.

5 Results and discussion

The proposed insider threat detection framework was tested on both the synthetic dataset and the CERT Insider Threat Dataset v6.2. Performance was compared across three setups: Isolation Forest, One-Class SVM, and the hybrid score fusion method. These results were also benchmarked against three standard baselines—Z-score, Local Outlier Factor (LOF), and Autoencoder.

To assess how effectively each anomaly detection method detects insider threats in a controlled setting, the

models were tested using the synthetic dataset. The results demonstrate the capability of each model to identify deviations within daily session windows. Table 6 below summarizes the precision, recall, F1-score, and AUC values for all evaluated models. Although deep learning models such as Siamese networks provide strong accuracy, the proposed hybrid method achieves a better balance of accuracy, interpretability, and computational efficiency.

Table 6. Performance on synthetic dataset.

Model	Precision	Recall	F1-Score	AUC
Isolation Forest	0.89	0.83	0.86	0.91
One-Class SVM	0.86	0.79	0.82	0.89
Fusion (IF + OC-SVM)	0.92	0.86	0.89	0.94
Z-Score	0.71	0.64	0.67	0.76
LOF	0.76	0.68	0.72	0.8
Autoencoder	0.83	0.75	0.79	0.87

The hybrid (fusion) model, combining Isolation Forest and One-Class SVM, outperforms the baselines in all metrics, demonstrating strong generalization and low false-positive rates. The graph shown in Figure 1 visualizes the comparative performance across key evaluation metrics.

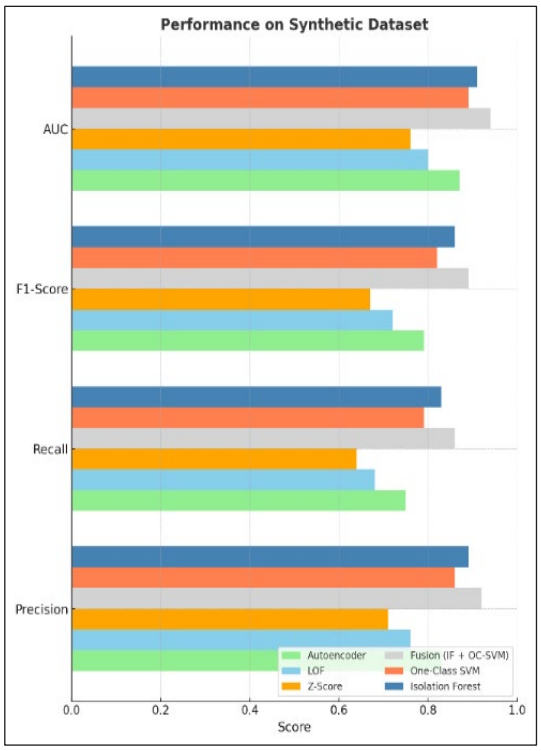

Fig. 1. Comparison of metrics on synthetic dataset.

To evaluate detection capability in a real-world enterprise environment, the models were also tested on the CERT Insider Threat Dataset v6.2. Compared to the synthetic dataset, the CERT dataset encompasses more nuanced user behaviors and realistic insider threat patterns, making it a crucial benchmark for external validity. Table 7 presents the performance metrics of each model on the CERT dataset.

Table 7. Performance on CERT dataset.

Model	Precision	Recall	F1-Score	AUC
Isolation Forest	0.85	0.77	0.81	0.88
One-Class SVM	0.8	0.73	0.76	0.85
Fusion (IF + OC-SVM)	0.87	0.8	0.83	0.89
Z-Score	0.69	0.61	0.65	0.75
LOF	0.72	0.66	0.69	0.78
Autoencoder	0.79	0.7	0.74	0.84

The fusion-based approach again demonstrated superior results, particularly in F1-score and AUC, indicating a strong balance between precision and recall in detecting insider deviations within authentic enterprise scenarios. Figure 2 visualizes the precision, recall, F1-score, and AUC of each model when tested on the CERT dataset.

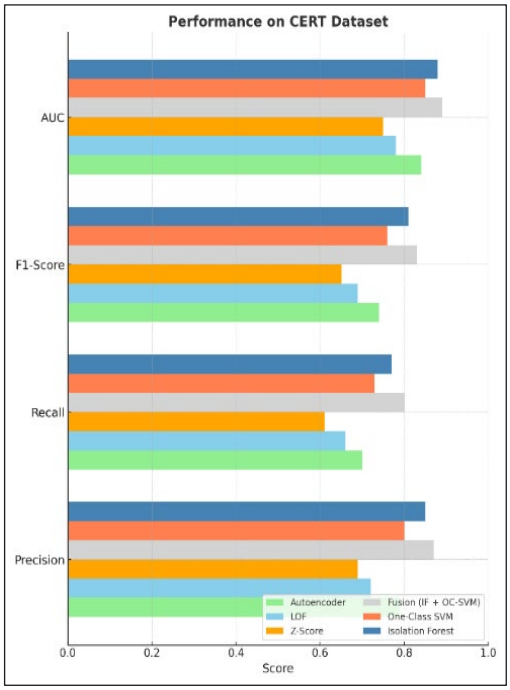

Fig. 2. Comparison of metrics on CERT dataset.

The hybrid model, which combines Isolation Forest and One-Class SVM, consistently outperformed all individual and baseline models on both datasets. On the synthetic dataset, it reached an F1-score of 0.89 and an AUC of 0.94, demonstrating a strong ability to identify diverse insider behaviors with high accuracy and few false positives. On the CERT dataset, which contains more nuanced and realistic user behaviors, the fusion model maintained competitive performance with an F1-score of 0.83 and an AUC of 0.89. Notably, individual models (IF or OC-SVM) showed solid results but were weaker in either precision or recall, highlighting the benefit of combining their strengths. To provide a consolidated view of the comparative performance across datasets, Figure 3 illustrates the F1-score and AUC values of all evaluated models, highlighting the superiority of the fusion-based approach.

Baseline methods, such as Z-score and LOF, performed significantly worse, particularly in recall,

indicating difficulty in detecting complex behavioral anomalies. While the autoencoder provided better generalization than statistical methods, it still lagged behind the hybrid approach, especially on the real-world CERT dataset.

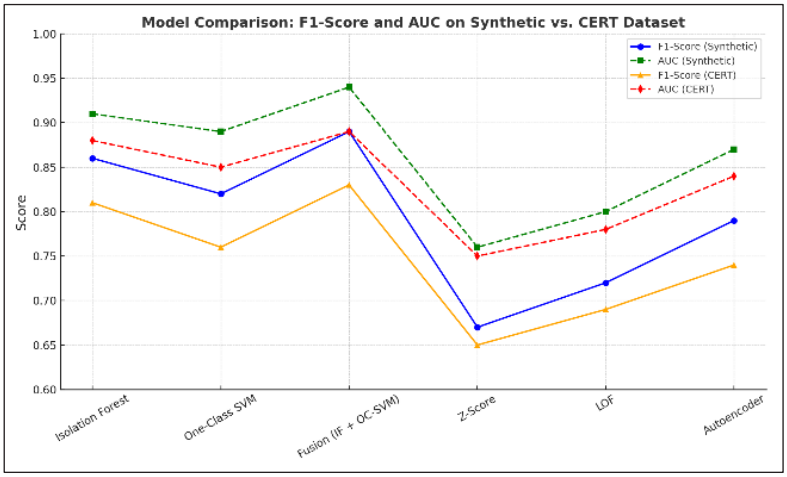

Fig. 3. Comparison of F1-score and AUC across models.

The results demonstrate the effectiveness of the profiling features and highlight that lightweight, interpretable models can achieve accurate detection. The incorporation of session-wise baselines, per-user normalization, and score fusion all contributed to a strong balance between performance and efficiency.

6 Conclusion and future scope

This study presents an interpretable and lightweight framework for insider threat detection based on behavioral profiling and anomaly scoring. By extracting session-level features from user activity logs and using hybrid detection methods that combine Isolation Forest and One-Class SVM, the proposed system showed strong performance on both synthetic and real-world datasets. Notably, the fusion approach achieved higher F1-scores and AUC values compared to traditional baselines, emphasizing its effectiveness in spotting subtle behavioral deviations. This method also remains suitable for privacy-sensitive environments, making it appropriate for distributed enterprise deployment.

Future research could focus on enhancing adaptability by developing self-updating behavior baselines, incorporating natural language content from communications, and exploring graph-based user interaction modeling to better understand temporal and relational contexts. Additionally, real-time deployment and cross-domain generalization are promising areas for further investigation and validation of the system. Although validated on CERT and synthetic datasets, future work should extend this framework to alternative domains such as IoT, healthcare, and finance to improve generalization.

References

1. X. Ye, F. Luo, H. Cui, J. Wang, X. Xiong, W. Zhang, J. Yu, W. Zhao, Research on insider threat detection based on personalized federated learning and behavior log analysis. Sci. Rep. **15**, 19214 (2025). https://doi.org/10.1038/s41598-025-04029-w
2. S. Song, N. Gao, Y. Zhang, C. Ma, BRITD: Behavior rhythm insider threat detection with time awareness and user adaptation. Cybersecur. **7**, 2 (2024). https://doi.org/10.1186/s42400-023-00190-9
3. T. Tian, C. Zhang, B. Jiang, H. Feng, Z. Lu, Insider threat detection for specific threat scenarios. Cybersecur. **8**, 17 (2025). https://doi.org/10.1186/s42400-024-00321-w
4. X. Tao, J. Liu, Y. Yu, H. Zhang, Y. Huang, An insider threat detection method based on improved test-time training model. High-Confid. Comput. **5**, 100283 (2025). https://doi.org/10.1016/j.hcc.2024.100283
5. M.F. Arroyabe, C.F. Arranz, I. Fernandez de Arroyabe, J. C. Fernandez de Arroyabe, Revealing the realities of cybercrime in small and medium enterprises: Understanding fear and taxonomic perspectives. Comput. Secur. **141**, 103826 (2024). https://doi.org/10.1016/j.cose.2024.103826
6. S.S. Pennada, S.K. Nayak, Insider threat detection using behavioural analysis through machine learning and deep learning techniques. Int. Res. J. Multidiscip. Technovation **7**, 74-86 (2025). https://doi.org/10.54392/irjmt2527
7. H. Sivaraman, Real-time anomaly detection for insider threat prevention in federal systems. ESP Int. J. Adv. Comput. Technol. **2**, 62-67 (2024). https://doi.org/10.56472/25838628/IJACT-V2I4P109
8. M. N. Al-Mhiqani, T. Alsboui, T. Al-Shehari, K.H. Abdulkareem, R. Ahmad, M.A. Mohammed, Insider threat detection in cyber-physical systems: A systematic literature review. Comput. Electr. Eng. **119**, 109489 (2024). https://doi.org/10.1016/j.compeleceng.2024.109489

9. U. Inayat, M. Farzan, S. Mahmood, M.F. Zia, S. Hussain, F. Pallonetto, Insider threat mitigation: Systematic literature review. Ain Shams Eng. J. **15**, 103068 (2024). https://doi.org/10.1016/j.asej.2024.103068
10. B. Racherache, P. Shirani, A. Soeanu, M. Debbabi, CPID: Insider threat detection using profiling and cyber-persona identification. Comput. Secur. **132**, 103350 (2023). https://doi.org/10.1016/j.cose.2023.103350
11. K. Randive, R. Mohan, A.M. Sivakrishna, An efficient pattern-based approach for insider threat classification using the image-based feature representation. J. Inf. Secur. Appl. **73**, 103434 (2023). https://doi.org/10.1016/j.jisa.2023.103434
12. T. Patel, S.S. Iyer, SiaDNN: Siamese deep neural network for anomaly detection in user behavior, Knowl.-Based Syst. **324**, 113769 (2025). https://doi.org/10.1016/j.knosys.2025.113769
13. H. Xiao, Y. Zhu, B. Zhang, Z. Lu, D. Du, Y. Liu, Unveiling shadows: A comprehensive framework for insider threat detection based on statistical and sequential analysis. Comput. Secur. **138**, 103665 (2024). https://doi.org/10.1016/j.cose.2023.103665
14. M. Amiri-Zarandi, H. Karimipour, R.A. Dara, A federated and explainable approach for insider threat detection in IoT. Internet Things **24**, 100965 (2023). https://doi.org/10.1016/j.iot.2023.100965
15. K. Fei, J. Zhou, Y. Zhou, X. Gu, H. Fan, B. Li, W. Wang, Y. Chen, LaAeb: A comprehensive log-text analysis based approach for insider threat detection. Comput. Secur. **148**, 104126 (2025). https://doi.org/10.1016/j.cose.2024.104126
16. J. Glasser, B. Lindauer, Bridging the gap: A pragmatic approach to generating insider threat data, In 2013 IEEE Security and Privacy Workshops, IEEE, San Francisco, CA, USA, July 22 (2013), 98-104