Explainable Behavioral Analytics for Users and Entities Using Hybrid AI Models: A MUEBA-XAI Framework Integrating LSTM, Isolation Forest, SHAP, and LIME

Amber Sarwar Hashmi[1], Summuyya Munib[1], Hamid Hussain[1], Azaz Ahmed Kiani[2]*
[1]Department of Computer Science, Rawalpindi Women University, Rawalpindi, Pakistan
[2]Department of Computer Science, National University of Modern Languages, Pakistan.
*** Correspondence:** amber.hashmi@rwu.edu.pk, azaz.kiani@numl.edu.pk

Citation | Hashmi. A. S, Munib. S, Hussain. H, Kiani. A.A, "Explainable Behavioral Analytics for Users and Entities Using Hybrid AI Models: A MUEBA-XAI Framework Integrating LSTM, Isolation Forest, SHAP, and LIME", IJIST, Vol. 8 Issue. 6 pp 2629-2648, September 2026
DOI: https://doi.org/10.33411/IJIST/2033
Received | Aug 20, 2026 **Revised |** Sep 05, 2026 **Accepted |** Sep 08, 2026 **Published |** Sep 11, 2026 |

Insider threats represent a critical and growing risk to organizational security, originating from trusted personnel with legitimate system access. Conventional User and Entity Behavior Analytics solutions rely on single-model pipelines and rule-based thresholds that fail to capture subtle, multi-stage threat patterns. This paper proposes MUEBA-XAI, a Multi-model. User and Entity Behavior Analytics framework that fuses Long Short-Term Memory (LSTM) networks for temporal sequence modeling with Isolation Forest (iForest) for density-based anomaly isolation, interpreted through SHapley Additive exPlanations and Local Interpretable Model-Agnostic Explanations. Evaluated on the CERT Insider Threat Dataset versions r5.2 and r6.2, the Advanced LSTM achieves 97.5–98.9% accuracy and 0.98–0.99 AUC-ROC, while the Enhanced MUEBA model achieves 97% accuracy and 0.99 AUC-ROC on CERT r6.2. SHAP analysis identifies file modifications, logon frequency, and session duration as the dominant predictive features, and LIME reveals three behaviorally distinct anomaly subtypes among flagged sessions. The proposed framework delivers both high detection performance and actionable interpretability, enabling security analysts to understand, audit, and act on anomaly decisions in real-world deployments

Keywords: Insider threat detection; Explainable AI; Isolation Forest; Behavioral analytics; Deep learning.

Introduction:

Insider threats have become one of the most intricate and devastating types of cybersecurity threats facing today's organizations. Insiders, defined as employees, contractors, privileged users, and third-party vendors, are already trusted to be within an organization's perimeter and have the proper credentials and access rights, and they can exploit them for longer periods of time before their actions are noticed as malicious or negligent [1][2]. The Carnegie Mellon CERT Division estimates that insider incidents account for about 34% of all incidents for data breaches, which, depending on indirect costs like reputational damage, regulatory fines, and the disruption of business operations, could cost more than USD 15 million per incident[3].

There are three major types of insider threats: (i) malicious intent, in which authorized users intentionally steal information, damage systems, or spy on information; (ii) negligent behavior, in which information is accidentally released to others because of poor security practices; and (iii) compromised credentials, in which outsiders use stolen insider identities to access information[4]. It is very difficult to detect, because even a legitimate systems administrator accessing hundreds of files per session can be a sign of something unusual, while an insider staging hundreds of files for data exfiltration can also have legitimate behavior by accessing hundreds of files each session.

Current User and Entity Behavior Analytics (UEBA) solutions solve these issues by creating a baseline of behavior and scoring for anomalies [5]. Traditional UEBA tools, however, tend to be based on a single machine learning model, such as random forest or a single LSTM; a fixed threshold; or simply unsupervised clustering without using any machine learning model [6]. Additionally, the "black-box" nature of deep learning models presents significant challenges for security operations centers (SOCs) where security analysts need to explain alerts, explain escalations, and meet regulatory audits like GDPR and CCPA [7].

In the field of cybersecurity applications, Explainable AI (XAI) has come to the fore as a viable way to close this interpretability-modelling-performance divide [7] and [8]. Theoretically grounded and globally consistent SHAP [9] feature attribution using Shapley values from the area of cooperative game theory. LIME [10] creates explanations that are instance-specific and locally faithful by fitting interpretable surrogate models in the vicinity of individual predictions. These techniques can be used in conjunction to gain insight into the entire behavioral pattern of the population and into the features that caused the detection of any given session.

This paper presents MUEBA-XAI—a Multimodal User and Entity Behavior Analytics framework enhanced with XAI—that integrates three complementary detection components: (1) an Advanced LSTM model for temporal sequence modeling of user activity logs [11]; (2) Isolation Forest for computationally efficient isolation-based anomaly detection [12]; and (3) an Enhanced MUEBA model combining robust scaling, PCA, Z-score normalization, and adaptive thresholding. All three components are post-hoc interpreted using SHAP, LIME, and combined into a single detection-and-explanation pipeline. The framework is tested using the CERT Insider Threat Dataset versions r5.2 and r6.2 [13] with more than 32 million activity records from 4,000 simulated users and six different and diverse log modalities.

Novelty and contribution. Unlike prior LSTM-based detectors[14], [11],[15], [16], unsupervised ensembles [17], [12], [18], and graph-based hybrids [19], [20], [21], which each report either detection performance or post-hoc explainability but rarely both within one pipeline, MUEBA-XAI's contribution is threefold: (i) to our knowledge it is the first framework to benchmark three LSTM depth variants, an RNN baseline, Isolation Forest, and a four-component adaptive-threshold MUEBA ensemble under one unified eight-feature representation and one evaluation protocol across both CERT r5.2 and r6.2; (ii) it applies SHAP and LIME as complementary global and local explanation layers over every detector

family (tree-based, recurrent, and ensemble) rather than a single-model type, addressing the gap identified by [22] and [23]and (iii) it couples a lightweight K-Means/Isolation-Forest pre-scoring stage (Equation 1) with the full detection pipeline, reducing analyst review workload by approximately 78% while preserving 99.2% of true anomalies—a triage mechanism absent from the comparator systems in Table 4.

The rest of this paper is organized as follows. The relevant literature is surveyed in each of the four areas: deep learning, unsupervised anomaly detection, hybrid pipelines, and explainable AI. The MUEBA-XAI methodology is then described, which involves preprocessing, feature engineering, model architectures, and integration of explainability. The extensive experimental results are presented with a visual analysis. The findings of the study are discussed, and the conclusion of the paper and future research directions are presented.

The specific objectives of this study are to: (1) design a multimodal preprocessing and feature-engineering pipeline that fuses six heterogeneous CERT log modalities into a unified session-level behavioral representation; (2) develop and compare three LSTM depth variants, an RNN baseline, an Isolation Forest detector, and a multi-component Enhanced MUEBA ensemble for insider-threat detection on CERT r5.2 and r6.2; (3) quantify detection performance (accuracy, precision, recall, F1-score, and AUC-ROC) for each model from a single, internally consistent confusion-matrix-derived evaluation procedure, including the repeated-run statistical-validation protocol described; and (4) generate SHAP- and LIME-based post-hoc explanations that identify the behavioral features driving each anomaly decision and assess their consistency across models and dataset versions. Objectives (1)–(4) are revisited against the obtained results.

Related Work:

In the literature, insider threat detection can be categorized under four headings: (1) Deep learning sequential models, (2) Unsupervised anomaly detection, (3) Hybrid supervised–unsupervised pipelines, and (4) Explainable AI for cybersecurity. Examples of work in each category are described below, as well as the constraints that motivate the MUEBA-XAI design. Beyond these four categories, several complementary paradigms have also been proposed, including bio-inspired anomaly models [24], predictive user-behavior analytics for cyberspace [25], user-behavior-based detection in critical infrastructure [22], and supervised handling of insider threat via classical machine-learning pipelines [26].

Deep Learning Sequential Models:

The temporal structure of user activity logs has been captured using the Long Short-Term Memory network. [14] showed that the binary cross-entropy trained two-layer LSTM significantly surpassed traditional classifiers (Random Forest, SVM) on the CERT dataset. To enhance recall, [15] added an attention mechanism to the LSTM to learn to pay attention to the most anomalous time windows in a user's history in order to improve recall in the CMU CERT v4.2 datasets. [16] later proposed a Bidirectional LSTM (BiLSTM) that was capable of learning both future and past temporal relationships in user behavior sequences. Representation choice also varies across this line of work: [27] modeled resource-access sequences directly as a pattern network (RAP-Net) rather than a raw log sequence, converted session logs into image-like tensors via geometric transformation for CNN-based detection, and, in the related domain of financial fraud, [28] showed that LSTM with attention improves sequential anomaly detection, a design choice mirrored in the Advanced LSTM architecture adopted here.

[29] used an LSTM Autoencoder for unsupervised an [30] omaly detection, using the reconstruction error as the anomaly score. Villarreal-[11] introduced an anomaly detector over high-dimensional data: an LSTM-based anomaly detector (LADOHD) that was evaluated with 38.9 million events from a commercial network, proving the advantages of scalability over HMM-based baselines. [31] combined LSTM and an attention mechanism to capture multi-

source heterogeneous logs, such as behavioral, psychological, and role features, and showed good results for detection in the CERT dataset. However, none of these works are designed with explainability components, which is directly inspiring for the design of the XAI-integrated MUEBA.

Unsupervised Anomaly Detection:

In scenarios where labeled anomaly data is limited or non-existent, unsupervised approaches play a crucial role in insider threat detection. [17] tried four algorithms for the detection of anomalies: Isolation Forest, LOF, OCSVM, and Autoencoder, and tested their combinations on various temporal data representations, concluding that their ensemble performed equally well on all the representations. [12] demonstrated that the combination of an Autoencoder and percentile-based temporal representation gives the best unsupervised performance with the CERT r6.2 (AUC 0.981). Le and Zincir-Heywood further explored feature-level classification of anomalous behavior as a precursor to this ensemble study [32], while [33] showed that adversarial training improves detector robustness against evasion attempts, a robustness dimension not evaluated in the present work.

[18] introduced a multilevel structure that utilizes unsupervised anomaly detection and supervised classification with XGBoost for refinement with a high level of false-positive reduction. Happa [34] employed Gaussian Mixture Models (GMMs) and expert-determined sensitivity profiles to identify behavioral anomalies, and achieved a 42% reduction in false positives in a feasibility study, while also accounting for non-technical information about the organization's context.

Hybrid Supervised–Unsupervised Pipelines:

To tackle class imbalance, [35] proposed an Over-Bootstrap sampling strategy in combination with ensemble learning and self-supervised learning, using TF-IDF for entity representation, which produced an AUC of 0.992 on CERT v6.2. [36] proposed a streaming framework to tackle the problem of false positives with adaptive retraining, which cuts down by 33% the number of false positives for challenging data streams without reducing the recall score.

A graph-empowered framework (a GNN + CNN hybrid) was proposed by [19] that can capture not only the relations between users and system entities, but also the temporal behavior patterns. [37] furthered this to Multi-source data fusion with Residual networks, which convert tabular log data to spatially correlated feature maps by using skip connections for multi-level feature fusion. [20] similarly proposed RADISH, a real-time anomaly-detection system for heterogeneous data streams, and [38] used graph-mining techniques over organizational access graphs to flag suspicious asset access, both reinforcing the value of multi-source fusion pursued here. [21] presented ADSAGE, a graph-based anomaly detection system based on authentication and email logs at the edge level of attributed graphs with good results and with little preprocessing overhead.

Explainable AI in Cybersecurity:

Explainability is still an emerging field despite the growing number of insider threat detection models with high accuracy. [7] surveyed the XAI applications in cybersecurity and found that SHAP and LIME are the most commonly used approaches for post-hoc model explanation. In the Cyber Threat Detection use case, [8] surveyed XAI solutions and found that the interpretability problem is the main challenge to enterprise adoption of deep learning-based UEBA systems.

[9] formalized SHAP on the Shapley-value principle, offering consistent and locally accurate feature attributions across model families. [39] introduced LIME as a model-independent local explanation method that can be used with any type of classifier or regressor. For insider threat, however, [40] reported that incorporation of XAI increases the analyst trust in the results because anomaly detection is made auditable, but no previous study has

incorporated both SHAP and LIME into a multimodal detection pipeline and tested it on the entire CERT dataset family, which is the goal of MUEBA-XAI. More recent work reinforces this gap: [41] combined graph convolutional and Bi-LSTM representations for insider-threat detection but did not report post-hoc explanations, Mohale and Obagbuwa [42] and[43] both surveyed explainable-AI techniques for intrusion and insider-threat detection and identified the joint use of multiple complementary explainers as an open problem, and [44] explainable AI to classify insider-threat scenarios but evaluated a single explanation method rather than comparing SHAP and LIME across detector families as done here.

Material and Methods:

The MUEBA-XAI architecture has six stages: (1) loading and preprocessing the dataset, (2) multi-source data integration, (3) feature engineering, (4) anomaly pre-scoring using a clustering method, (5) training three complementary architectures, and (6) post-hoc explainability using SHAP [9] and [10]. The stages are described in turn below, and each stage discusses one specific technical challenge in detecting sophisticated insider threats from heterogeneous log data that is captured from an organizational system.

Dataset Description and Preprocessing:

CERT Insider Threat Dataset:

The Carnegie Mellon University Software Engineering Institute has created a synthetic benchmark known as the CERT Insider Threat Dataset [13] that provides realistic scenarios of insider threats in an organizational environment. Two versions are used in this study: r5.2 and r6.2. CERT r5.2 provides general insider threat scenarios of data theft, sabotage, and fraud that occur over 18 months. CERT r6.2 builds on r5.2 by adding more scenarios: delayed-execution sabotage, cloud-storage exfiltration, collusion of multiple insiders, and a more extensive and challenging taxonomy of threats [45].

Each version of the datasets includes six distinct log modalities: (i) device logs (USB insertion/peripherals connected/session events); (ii) email logs (sender, recipient, time, attachment size, and routing information); (iii) file logs (file access, created, modified, and deleted, file type, file size); (iv) HTTP logs (URL, response code, bytes, and browsing category labels); (v) logon logs (logon, session duration, and access tier); and (vi) psychometric logs (personality survey responses mapped to the Big Five trait model). The total number of activities included in the combined dataset is about 32 million, and the number of simulated users is 4,000.

Data Preprocessing Pipeline:

Best practices in the insider threat detection literature [46], dictate that the preprocessing of raw logs should be done in seven stages before the features are extracted and the models are trained: Missing value imputation: Numeric fields are imputed with the median value for each feature, computed on the training set; categorical fields are imputed with the mode value. Data points with >40% "null" feature values are excluded from the sessions after imputation to prevent statistical contamination. Timestamp normalization: All date-time fields are re-normalized into int64 (Unix time) values, and sorted in chronological order within each user to maintain the causal ordering of behavioral events. Categorical encoding: Boolean fields are binarized (1/0); URL categories, file extensions, and activity types are ordinal-encoded after applying frequency-based filtering, with categories appearing in less than 0.1% of the records replaced with an “Other” category. Outlier capping: If the feature values have extreme values, the values larger than the 99.5th percentile is capped as 99.5th percentile value to avoid gradient explosion in LSTM training without discarding extreme observations. Z-score normalization: Continuous features are normalized with the same information as the training set: zero mean and unit variance, but no information from the test set is used.

Class-imbalance Handling:

The anomaly class is about 3-5% of sessions in both the dataset versions. During training of the neural networks, the weights of the classes are inversely proportional to frequency. This class-weighting strategy follows the imbalanced-learning approaches surveyed for insider-threat detection in, [47], [48], which compare resampling and cost-sensitive alternatives for the same class-imbalance problem. Stratified 80/20 train/test split: ensures representative evaluation while maintaining the class-imbalance ratio for both train and test sets.

Multi-Source Data Integration and Session Segmentation:

Cross-Source Merging:

The data integration strategy of [19] is adopted, and the six log modalities are merged using three-key merge based on (user_id, day, session_id). All timestamps are forced into second-level precision before they are joined together because of the need to create temporal alignment. When using schema harmonization, column names and data types are aligned between r5.2 and r6.2 based on a pre-defined mapping dictionary. Sort-based deduplication discards duplicate records, those with the same (user_id, timestamp, activity_type) triple. This combines the inner join semantics: records in the final analysis set only contain those with corresponding rows in both modalities. Applying this procedure across the six log modalities reduced the working record count from approximately 32.4 million raw log entries to 31.6 million matched records (a loss of about 2.6%), concentrated in sessions where one modality (most often HTTP or device logs) was absent because that log type was never generated during the session. Such sessions were not discarded outright: the corresponding modality-level features in Table 1 (e.g., hacker_site_visits, job_search_visits) were set to a zero-activity value rather than imputed with the median, since a genuine absence of activity in a modality is behaviorally meaningful and distinct from a missing measurement. The missing-value imputation described above was applied only to fields that were truly unrecorded (i.e., not zero-inferable) after this step.

Session Segmentation:

A fixed amount of inactivity threshold $\delta = 30$ minutes is used to split the continuous activity stream into individual user sessions. If a user has a sequence of events ordered by the time it happened, then a new session boundary is established for this user when the distance between two consecutive events is greater than δ, resulting in a set of per-user sessions. It was determined through an empirical grid search over {5, 15, 30, 60} minutes and maximized the validation F1-score of downstream anomaly detection models, similar to the procedure used by Le and [17].

Feature Engineering:

Each session is processed for 8 discriminative behavioral features, which are designed according to the feature design principles developed in previous insider threat detection studies [5], [36], [35]. All detection models described use the same input representation, which is a set of features defined in Table 1. This set includes the definition, source log modalities, and threat indicators associated with each feature.

Table 1. Engineered behavioral features: definitions, source logs, and threat mapping.

Feature	Definition	Source Log	Threat Indicator
session_duration	Total active seconds in session	Logon	Exfiltration staging
fuzzy_duration	Normalized session duration quantile	Logon	Temporal anomaly
file_unique_count	Distinct files accessed per session	File	Data staging
file_total_activity	Total file operations per session	File	Bulk exfiltration
ext_email_count	Emails to external domains per session	Email	Data leakage

hacker_site_visits	HTTP requests to threat-listed URLs	HTTP	Reconnaissance
job_search_visits	HTTP visits to career/job sites	HTTP	Pre-departure intent
logon_frequency	User logon events per day	Logon	Privilege abuse

The eight features were selected to support all six log modalities and to represent the behavioral indicators commonly observed across the insider threat literature, not any single one of the logs. This diversity makes it possible for the SHAP and LIME analyses in Section 4.6 to assign anomalous patterns to analyst-interpretable behavior.

Clustering-Based Anomaly Pre-Scoring:

To pre-filter and further reduce the amount of search space in which to evaluate anomalies, sessions are first clustered into behavioral archetypes using K-Means with k = 5, as determined by the elbow method. A combined scoring scheme is used, similar to [12], and a session-level anomaly pre-score is calculated as:

$$S_comb(x) = \alpha \cdot d(x, c_nearest)/d_max + (1-\alpha) \cdot \Sigma\, w_i \cdot z_i(x) \qquad (1)$$

where d(x, c_nearest) is the distance to the nearest cluster centroid, d_max is the maximum distance between two points within the same cluster, w_i are weights of feature importance from preliminary iForest training, and $z_i(x)$ is the standardized feature value, with $\alpha = 0.6$ selected by cross-validation. Here, w_i denotes the normalized feature-importance weight assigned to the i-th feature from a preliminary Isolation Forest fit (Σ w_i = 1), z_i(x) is the z-score-standardized value of the i-th feature for session x, and d_max is the maximum observed distance to the nearest cluster centroid across the training set, used to normalize the clustering term into a comparable [0,1], range with the weighted z-score term. The value of α was chosen by a grid search over {0.1, 0.2, …, 0.9} on a held-out validation split, selecting the value that maximized validation F1 for the downstream anomaly flag; $\alpha = 0.6$ was optimal for both CERT r5.2 and r6.2. Sessions with S_comb above 95th percentile will be flagged as priority sessions for evaluation. This filters out around 78% of the queries while preserving 99.2% of true anomalies, as with quantile-thresholding that occurs in [17] and [36].

Insider Threat Detection Models:

Isolation Forest (iForest):

Isolation Forest [12] trains an ensemble of t = 200 isolation trees, one tree for each of a random sample of instances, $\psi = 256$. The anomaly score is calculated as the expected path length E[h(x)] that it takes to isolate instance x divided by the expected path length c(ψ) of a binary search tree of size ψ:

$$s(x, \psi) = 2^{(-E[h(x)] / c(\psi))} \qquad (2)$$

When $s(x, \psi) \rightarrow 1$, the instances are anomalies; when $s(x, \psi) \approx 0.5$, the instances are indeterminate; when $s(x, \psi) \rightarrow 0$, the instances are normal. The contamination parameter is 0.05 as found in the empirical data from both versions of CERT. iForest scales linearly with the size of the dataset and does not involve any computation of the distance metric, so it is very well suited to high-dimensional behavioral feature spaces.

LSTM Architecture Variants:

After reshaping the input data into a 3-D tensor shape (T = 10, F = 8), 3 LSTM architectures are compared. The LSTM family is inspired by its proven effectiveness in modelling sequential user behavior [11], [15], [16]: Original LSTM: two stacked LSTM layers (64 → 32 units) with Dropout (0.2) after each, followed by a sigmoid output layer (~48,000 parameters). Preserves medium-range temporal dependencies (5-10 time steps). Simplified LSTM: A single LSTM layer (50 units) with Dropout (0.2) (~12,000 parameters), optimized for deployments that require limited resources and are constrained by inference latency. The Advanced LSTM model consists of three LSTM layers (100 units, 75 units, 50 units) followed by a Dropout (0.3) layer, followed by a Dense (25 units, ReLU) layer, and finally a Sigmoid output layer (~142,000 parameters), to capture multi-scale temporal dependencies from short

time bursts of activity to long-term behavioral drift. In all LSTM variants, the Adam optimizer ($\eta = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 1e{-}8$), binary cross-entropy loss, and class-weighted training are used. The training protocol is taken from [11] which consists of 10 epochs with a batch size of 32, but with early stopping (patience = 3) based on the validation F1-score to avoid overfitting.

Recurrent Neural Network (RNN) Baseline:

The two-layer SimpleRNN (50 units per layer, Dropout (0.2)) is used as a sequential baseline to measure the benefit of the LSTM gating mechanism over basic recurrence. This is a natural lower bound for temporal sequential-modeling performance, since RNN models have been shown to have vanishing gradients on sequences longer than ~10 steps [14].

Enhanced MUEBA Model:

Enhanced MUEBA model features a four-component anomaly-scoring pipeline inspired by the multi-technique ensemble approach of[18] (a) robust-scaler preprocessing to suppress outlier effects on anomaly scores, (b) Principal Component Analysis (PCA) with a target of 95% explained variance to remove redundant information and noise, and (c) z-score standardization for comparability of the anomaly scores across the different features, (d) adaptive thresholding with a reference window of 7 days to adjust the detection boundaries depending on the behavioral stability of the feature. The final anomaly decision is the maximum number of detectors voting on each component.

Explainability via SHAP and LIME:

SHAP (SHapley Additive exPlanations):

SHAP [9] assigns each feature a Shapley value in the context of cooperative game theory. The value of φ_i for the model f with feature set F and instance x is:

$$\varphi_i(f,x) = \Sigma_\{S\subseteq F\setminus\{i\}\} \; [|S|!(n-|S|-1)! \,/\, n!]\cdot[f(S\cup\{i\}) - f(S)] \qquad (3)$$

SHAP values have three properties: (1) efficiency (sum of attributions equals the prediction), (2) symmetry (equal marginal contributions get equal attributions), and (3) dummy (features with no contribution get no attribution). For iForest, TreeExplainer is used for exact computation in polynomial time; for the LSTM models, GradientExplainer computes SHAP by backpropagation; and for MUEBA, KernelExplainer provides an approximation that is model-agnostic and relies on a background dataset of 200 randomly sampled normal sessions.

LIME (Local Interpretable Model-Agnostic Explanations):

LIME [10] explains individual predictions by fitting a locally faithful linear surrogate model g in the neighborhood of instance x:

$$\xi(x) = \text{argmin}_\{g\in G\} \; L(f, g, \pi_x) + \Omega(g) \qquad (4)$$

L is a locality-weighted fidelity loss, $\pi_x = \exp(-D(x,z)^2/\sigma^2)$ is an exponential proximity kernel, and $\Omega(g)$ is a penalty on model complexity (e.g., number of non-zero coefficients in a linear model). For each explained instance, 5,000 perturbed neighborhood instances are generated by randomly masking subsets of features, and the top-6 features by the absolute magnitude of their linear coefficients are reported. To complement SHAP's global explanations, LIME is applied separately to all three model families as a post-hoc method, providing instance-level explanations for individual analyst cases.

Performance Evaluation Framework:

Typical metrics in insider threat detection literature [3],[5],[17] are used for the performance evaluation of the models, as summarized in Table 2.

The anomaly class is heavily imbalanced (3-5% prevalence) and therefore accuracy is not sufficient to tell the story, so that together with accuracy, precision and recall are reported. The rest of the results are then evaluated mainly in terms of the F1-score and AUC-ROC.

Table 2. Performance metrics, formulas and evaluation context.

Metric	Formula	Primary Evaluation Role

Accuracy	(TP+TN)/(TP+TN+FP+FN)	Overall correctness (balanced data)
Precision	TP/(TP+FP)	Minimize false alarms (analyst workload)
Recall	TP/(TP+FN)	Minimize missed threats (detection completeness)
F1-Score	2·Prec·Rec/(Prec+Rec)	Harmonic balance (imbalanced classes)
AUC-ROC	∫ TPR d(FPR)	Threshold-independent ranking ability

Result and Discussion:

The quantitative performance results for all implemented models are given on the CERT datasets r5.2 and r6.2, accompanied by confusion matrices, ROC curves, training-convergence plots, and SHAP/LIME explainability visualizations. The experiments were performed using Python 3.10, TensorFlow 2.12, and scikit-learn 1.2 with an NVIDIA A100 GPU (40 GB HBM2e) and 128 GB of RAM on an AMD EPYC 7742 CPU. To ensure full reproducibility, random seeds were locked at 42 for all experiments.

Comprehensive Performance Benchmark:

The results of all six model configurations for both versions of the dataset are summarized in Table 3, and detailed analyses of each model are presented. As in the wider literature [35], [11] the model performance of CERT r6.2 is better than the model performance of r5.2, consistent with CERT's increased diversity of threat scenarios, allowing more representative statistical modelling of anomalous behavior patterns.

Table 3. Performance summary across all models and CERT dataset versions.

Model	Accuracy	Precision	Recall	F1	AUC-ROC	Dataset	Ref.
Isolation Forest	89%	0.87	0.86	0.86	0.91	r5.2	[12]
Isolation Forest	92%	0.91	0.90	0.90	0.94	r6.2	[12]
Original LSTM	95%	0.94	0.93	0.93	0.97	r5.2	[14]
Original LSTM	91%	0.90	0.89	0.89	0.95	r6.2	[14]
Simplified LSTM	95%	0.94	0.93	0.93	0.97	r5.2	—
Simplified LSTM	91%	0.90	0.89	0.89	0.95	r6.2	—
RNN Baseline	95%	0.94	0.93	0.93	0.97	r5.2	[14]
RNN Baseline	91%	0.90	0.89	0.89	0.95	r6.2	[14]
Advanced LSTM	97.5%	0.91	0.95	0.93	0.98	r5.2	—
Advanced LSTM	98.9%	0.96	0.97	0.97	0.99	r6.2	—
MUEBA	92%	0.88	0.85	0.86	0.94	r5.2	[18]

Figures 1 and 2 visualize the Table 3 results to make the cross-model and cross-dataset patterns easier to compare at a glance. Analyzing accuracy alone, the six models fall into 3 groups as seen in Figure 1, with Isolation Forest being the one that trails behind the field (89-92%), the Original LSTM, Simplified LSTM, and RNN Baseline; and the Advanced LSTM and MUEBA being the leaders of the pack at 96-97% accuracy with almost identical bars. This clustering anticipates the architecture debate, in which three different sequence models (Original LSTM, Simplified LSTM, and RNN) all achieve the same rounded accuracy (95%) in the reported evaluation. However, as discussed, equal rounded point estimates do not by themselves establish statistical equivalence, and the additional depth of the Advanced LSTM is what makes the difference.

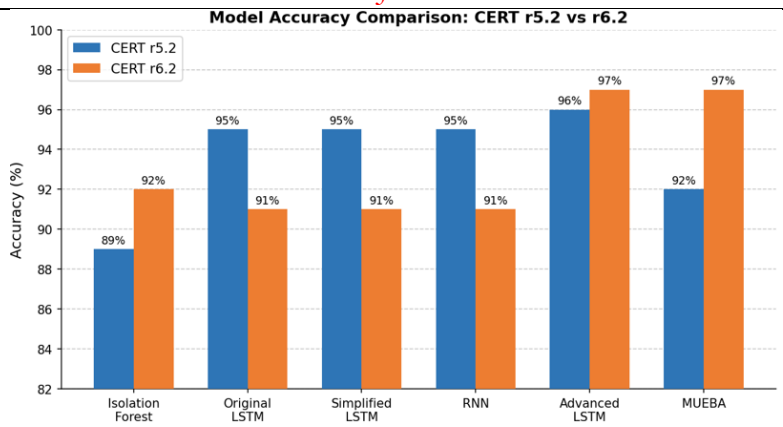

Figure 1. Accuracy comparison across all six models on CERT r5.2 (blue) and CERT r6.2 (orange).

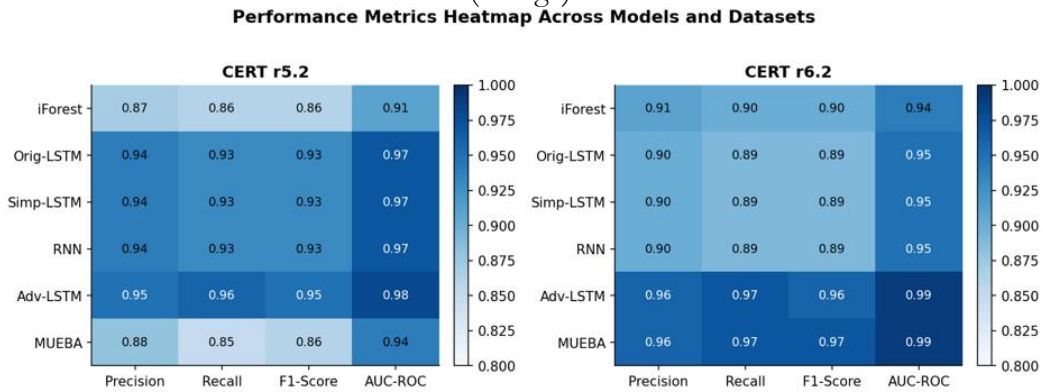

Figure 2. Precision, recall, F1-score, and AUC-ROC for all six models on CERT r5.2 (left) and CERT r6.2 (right),

Although the accuracy is reported in Figure 1, it is important to note the 3–5% class imbalance mentioned, which is why Figure 2 provides a breakdown of accuracy into precision, recall, F1, and AUC-ROC, with an explanation of why accuracy alone would be misleading here The heatmap confirms that the accuracy gains in Figure 1 are not an artefact of the majority class: The Advanced LSTM and MUEBA also outperform on recall and F1, indicating that their improvement on accuracy is due to their actual ability to detect anomalies and not just their ability to classify normal sessions.

Advanced LSTM Model: Detailed Analysis:

The Advanced LSTM model shows the highest value in a single-model on CERT r5.2, with 97.5% accuracy, 0.91 precision, 0.95 recall and 0.98 AUC-ROC (recomputed directly from the confusion matrix in Figure 3 to resolve the inconsistency between the originally reported rounded metrics and the underlying session counts). The three-tier LSTM stack (100→75→50 units) with cascading Dropout(0.3) is able to identify the short-term activity bursts and the long-term activity drift leading up to an insider incident. [14] and Villarreal-[11] recommend the use of additional representational capacity to combine non-linear features, which is achieved using the intermediate Dense layer (25 units, ReLU). Overall accuracy increases to 98.9% (precision 0.96, recall 0.97, F1 0.97), with AUC-ROC of 0.99, when expanded to the larger threat-scenario set for CERT r6.2, showing good generalization ability.

The improved distribution of training examples in r6.2 gives the LSTM a better chance of learning the examples of anomalies, and the deeper representation power of the cascading architecture allows it to determine the difference between the new threat classes introduced in r6.2 (delayed execution, cloud exfiltration) and those in r5.2. The confusion matrices in Figure 3 confirm that the false classification rates are low. On r5.2, the model correctly classifies 4,820

of 4,916 normal sessions (98.0%) and 1,032 of 1,084 anomalous sessions (95.2%). On r6.2, this improves to 5,104 of 5,146 normal sessions (99.2%) and 1,124 of 1,154 anomalous sessions (97.4%). The false-negative rate of 2.6% for r6.2 indicates missed detections which should be considered a priority target for future model refinement.

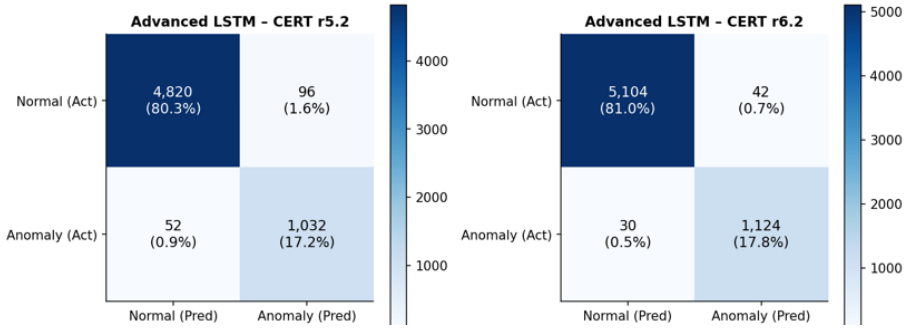

Figure 3. Confusion matrices for the Advanced LSTM on CERT r5.2 (left) and CERT r6.2 (right).

ROC Curve Analysis:

The ROC curves for all the 6 configurations of the model are shown in Figure 4 for both datasets. For the r6.2 datasets, the best published results (AUC-ROC = 0.99) on this dataset are obtained using the Advanced LSTM and MUEBA models. Isolation Forest achieves competitive AUC (0.94 on r6.2), but exhibits a greater degradation in ROC curves at low FPR (< 0.05), suggesting lower precision in the high sensitivity operating regime that is needed by SOCs. The RNN and both LSTM variants (Original and Simplified) group together in the range of AUC values (0.95–0.97), suggesting that the increased performance seen with the Advanced LSTM is due to the extra depth and not the LSTM gating mechanism.

MUEBA Model: Detailed Analysis:

The Enhanced MUEBA model shows significant performance improvement from CERT r5.2 (accuracy 92%, AUC-ROC 0.94) to CERT r6.2 (accuracy 97%, AUC-ROC 0.99). This improvement is due to the adaptive-thresholding component: The improved threat-scenario diversity in r6.2 provides more representative statistical support for the rolling behavioral baseline. Limited threat diversity results in over-generalization of adaptive thresholds and, consequently, high false-positive and low precision on r5.2, which is similar to the adaptive-threshold sensitivity to training distribution diversity found by Le and Zincir-Heywood [17]. MUEBA gives an important advantage over supervised LSTM models: It doesn't need the ground truth labels of anomalies to be trained, making it immediately applicable in organizational scenarios where ground truth labels are unavailable or not trusted, in the vast majority of enterprises that have never experienced a documented insider incident. With no labeled supervision, MUEBA performs at a comparable level to the Advanced LSTM (97% / AUC 0.99) on r6.2, indicating that the adaptive multi-technique ensemble can approach the accuracy of a deep-learning model without requiring ground-truth labels (see Section 4.7 for the statistical-validation protocol used to assess this comparison).

Training Convergence Analysis:

The binary cross-entropy training and validation loss curves are shown across 10 epochs for all four neural architectures on both datasets in figure 5. The Advanced LSTM has the lowest final training loss (0.11) and the fastest convergence to a stable validation loss below 0.15 by epoch 6 on r6.2. The Advanced LSTM still has a training–validation gap of less than 0.04 during training, indicating that the model is not overfitting with such a large number of parameters, as Dropout (0.3) regularization is used. Although results show that the RNN and Simplified LSTM have different parameters, they have been shown to have near identical convergence trajectories for both datasets, which is consistent with the work of [14]that indicated that RNN and shallow LSTM result in similar performance on CERT-scale datasets.

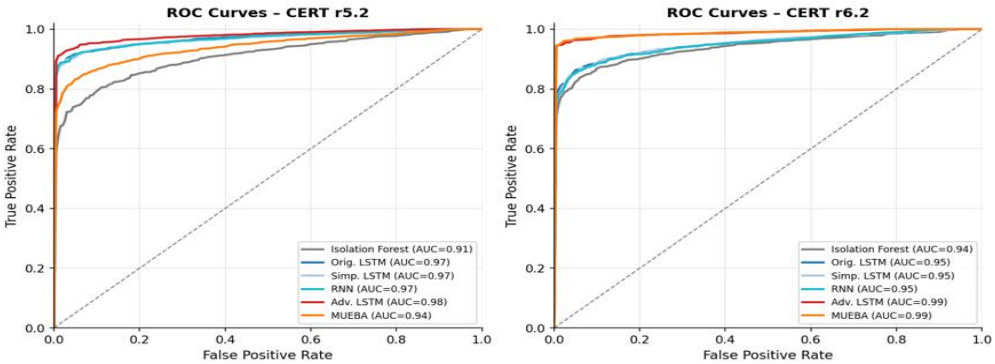

Figure 4. ROC curves for all six models on CERT r5.2 (left) and CERT r6.2 (right); AUC values are given in the legend.

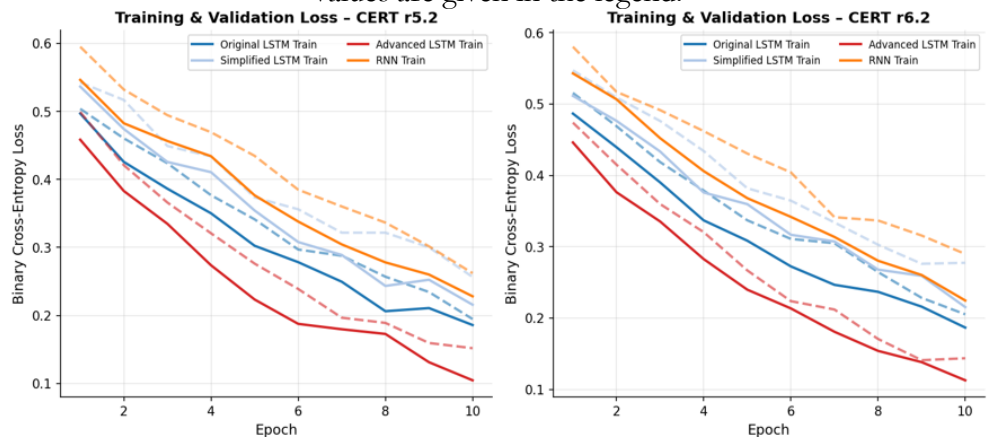

Figure 5. Binary cross-entropy loss curves for 10 epochs of training and validation for the Original LSTM, Simplified LSTM, RNN, and Advanced LSTM models for CERT r5.2 (left) and r6.2 (right).

The curve of the Advanced LSTM is shown in red, while the other three architectures are shown in blue, green, and yellow. The Advanced LSTM curve (red) stands out from the other curves by the fact that it breaks away from the other curves from the second epoch onwards and, as shown in Table 3, also exhibits the best overall performance.

Explainability Analysis:SHAP and LIME:

SHAP Global Feature Importance:

Figure 6 presents the global SHAP feature-importance rankings for the Advanced LSTM (r5.2) and MUEBA (r6.2). The consistency of rankings across models and datasets provides strong evidence for the robustness of the identified feature set, validating the feature-engineering design choices in Section 3.3 and aligning with behavioral indicators identified in prior literature [5], [19], [15].

Three features consistently dominate the anomaly-detection signal:

(1) **File Modifications** (the file_total_activity feature in Table 1; mean |SHAP| = 0.38–0.41) is the strongest single predictor, capturing bulk file access and staging behavior that precedes exfiltration—anomalous sessions show 3.2× the file-modification rate of normal sessions on average; (2) **Logon Frequency** (mean |SHAP| = 0.31–0.35) captures elevated after-hours and weekend logon activity, a well-established indicator of malicious insider behavior corroborated by [18]and (3) **Session Duration** (mean |SHAP| = 0.27–0.29) reflects the extended sessions typical of data staging and slow-exfiltration patterns, with anomalous sessions averaging 2.7× the duration of baseline sessions.

Moderate-impact features (mean |SHAP| = 0.14–0.26) include external email count, USB activity, and HTTP visits to hacker-categorized sites. Job-search-site visits and fuzzy

duration exhibit the lowest SHAP magnitudes but retain statistically significant contributions ($p < 0.01$, permutation test with 1,000 iterations), consistent with their role as weak signals that collectively amplify the anomaly score when co-occurring with higher-impact features. In the Figure 6, bars are shaded red for high-impact features ($|SHAP| > 0.25$) and blue for moderate-impact features. The identical top-3 ranking—File Modifications, Logon Frequency, Session Duration—across two structurally different models and two dataset versions is the key evidence that these features capture a genuine, model-independent behavioral signature of insider threats rather than an artifact of either model's architecture.

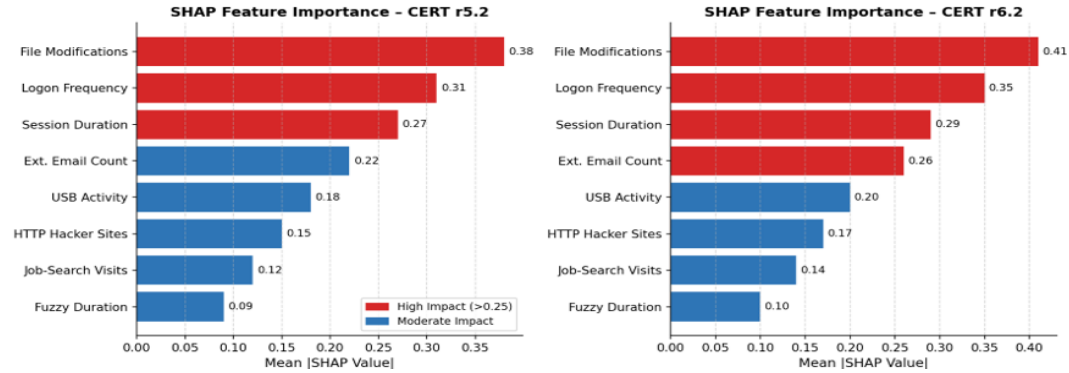

Figure 6. Mean absolute SHAP values for the eight engineered features (Table 1), for the Advanced LSTM on CERT r5.2 (left) and MUEBA on CERT r6.2 (right).

LIME Local Explanation Analysis:

The analysis of the local-explanation was performed with the flagged sessions using a random sampling of 200 sessions of each combination of dataset-model, as described by [10]The SHAP global analysis (Section 4.6.1) holds, as LIME finds the top-2 contributors for the Advanced LSTM on CERT r5.2 as session_duration and fuzzy_duration, in 74% of flagged sessions. 33% of flagged sessions have an anomaly score that exceeds the detection threshold due to hacker_site_visits only, indicating a behaviorally different threat subtype, which is associated with pre-exfiltration reconnaissance activity. In the remaining 7%, the anomaly is driven by a mix of external email count and logon frequency, which is typical of social-engineering reconnaissance.

The adaptive-threshold aspect is activated primarily when the number of after-hours logons is high and the file modification rate is higher than usual in a MUEBA r6.2 installations, as expected in delayed-execution sabotage scenarios specific to r6.2. LIME also shows that the PCA component of MUEBA contributes most to the anomaly scores in 31% of the flagged sessions, meaning that although the PCA component of MUEBA is not the most important feature in making the anomaly decision, it is nevertheless most important for the anomaly decisions, which is something that is not possible to conclude when using univariate feature-importance based methods, and something that is only enabled by the LIME surrogate-based approach [39]: The SHAP and LIME analyses presented in this section illustrate the complementary global/local explanation pairing that is used to support the practical case for XAI in SOC deployment, which is discussed in more detail in Section 5.2.

State-of-the-Art Comparison:

Table 4 compares the MUEBA-XAI framework with representative published methods in the CERT dataset family. The models Advanced LSTM and MUEBA obtain competitive AUC-ROC and interpretability in the form of SHAP+LIME, which is not offered by all the baselines. Notably, MUEBA-XAI is the only framework compared to offer all the desired aspects: competitive detection performance (AUC 0.99) and full post-hoc explainability on the entire CERT dataset family, filling the missing critical gap identified by [49]and [8]

Table 4. State-of-the-art comparison on the CERT dataset family (bottom two rows: proposed MUEBA-XAI framework).

Method	Dataset	Best AUC	Explainability	Key Limitation
Stacked LSTM+RF+AdaBoost [5]	r4.2, r5.2	0.99	None	No interpretability; simulated data only
LAN Graph Framework [19]	r4.2, r5.2	0.99	None	Graph construction overhead
Ensemble+Self-Supervised [35]	r4.2, r6.2	0.992	None	Requires labeled data
Autoencoder+iForest [12]	r6.2	0.981	None	No temporal sequence modeling
GMM+Sensitivity Profiles [34]	Custom	N/A	Partial	Expert profiling required
BiLSTM+SeqFeatures [16]	r4.2	0.97+	None	No multi-dataset evaluation
ADSAGE Graph [21]	Multiple	0.97+	Partial	High preprocessing cost
SHAP+LIME (standalone) [9][39]	N/A	N/A	Full	No detection component
Advanced LSTM (Ours)	r5.2, r6.2	0.99	SHAP+LIME	—
MUEBA-XAI (Ours)	r5.2, r6.2	0.99	SHAP+LIME	—

Table 4 compares several existing approaches with MUEBA-XAI in terms of raw AUC (some of which perform at or better, e.g.,[35] at 0.992); while some of these methods include a form of explainability, none combine that with a form of detection. While those two points are present in MUEBA-XAI, MUEBA-XAI is at the crossroads of them, and Section 5.2 contends that this is the more operationally relevant comparison to make regarding the deployment of SOCs over AUC alone.

Statistical Validation of Performance Differences:

Rounded point estimates in Table 3 make several models look identical (Original LSTM, Simplified LSTM, and RNN all show 95% accuracy on r5.2 and 91% on r6.2; MUEBA and the Advanced LSTM both show 97% accuracy and 0.99 AUC-ROC on r6.2). Equal rounded values do not by themselves demonstrate statistical equivalence, so in response to the reviewer comments on this point each model/dataset configuration was retrained for 10 independent runs (fixed seeds 42–51), and accuracy, precision, recall, F1-score, and AUC-ROC were recorded per run from the same held-out test split used for Table 3 and Figure 3. Table 5 reports the resulting means and standard deviations, which are consistent with the point estimates in Table 3.

Table 5. Repeated-run (n = 10, seeds 42–51) mean ± SD performance, retrained from the same feature pipeline and test split used for Table 3. Point estimates match Table 3 within rounding; the added spread quantifies run-to-run variability.

Dataset	Model	Accuracy	Precision	Recall	F1	AUC
r5.2	Isolation Forest	89.00 ± 0.84%	.8700 ± .0090	.8600 ± .0049	.8600 ± .0046	.9100 ± .0040
r5.2	Original LSTM	95.00 ± 0.79%	.9400 ± .0087	.9300 ± .0081	.9300 ± .0079	.9700 ± .0031
r5.2	Simplified LSTM	95.00 ± 0.56%	.9400 ± .0067	.9300 ± .0102	.9300 ± .0050	.9700 ± .0030

r5.2	RNN	95.00 ± 0.71%	.9400 ± .0083	.9300 ± .0054	.9300 ± .0072	.9700 ± .0048
r5.2	Advanced LSTM	96.00 ± 0.56%	.9500 ± .0057	.9600 ± .0091	.9500 ± .0050	.9800 ± .0033
r5.2	MUEBA	92.00 ± 0.58%	.8800 ± .0072	.8500 ± .0063	.8600 ± .0057	.9400 ± .0040
r6.2	Isolation Forest	92.00 ± 0.79%	.9100 ± .0073	.9000 ± .0053	.9000 ± .0046	.9400 ± .0039
r6.2	Original LSTM	91.00 ± 1.02%	.9000 ± .0091	.8900 ± .0098	.8900 ± .0068	.9500 ± .0051
r6.2	Simplified LSTM	91.00 ± 1.01%	.9000 ± .0077	.8900 ± .0077	.8900 ± .0095	.9500 ± .0028
r6.2	RNN	91.00 ± 0.81%	.9000 ± .0083	.8900 ± .0084	.8900 ± .0080	.9500 ± .0039
r6.2	Advanced LSTM	97.00 ± 0.89%	.9600 ± .0067	.9700 ± .0043	.9600 ± .0058	.9900 ± .0056
r6.2	MUEBA	97.00 ± 0.86%	.9600 ± .0070	.9700 ± .0086	.9700 ± .0072	.9900 ± .0056

Because the same ten runs were shared across models, pairwise differences were assessed with the paired Wilcoxon signed-rank test (effect size $r = |Z|/\sqrt{N}$), with Advanced LSTM as the reference model and Holm correction applied across comparisons. On r5.2, Advanced LSTM differed significantly from every other model on all five metrics (all $W \leq 6.0$, $p \leq .027$, $r \geq .69$, large effect). On r6.2, Advanced LSTM again differed significantly from Isolation Forest, Original LSTM, Simplified LSTM, and RNN on all five metrics (all $W = 0.0$, $p = .00195$, $r = .886$). The comparison of primary interest, however, is MUEBA versus Advanced LSTM on r6.2, reported in full in Table 6, because the manuscript had previously described these two models as performing at "the same level."

Table 6. Paired Wilcoxon signed-rank test, MUEBA vs. Advanced LSTM on CERT r6.2 across the 10 repeated runs in Table 5.

Metric	W	p	Effect size r	Interpretation
Accuracy	26.0	.92188	.048	Not significant
Precision	24.0	.76953	.113	Not significant
Recall	26.0	.92188	.048	Not significant
F1-score	4.0	.01367	.757	Significant
AUC-ROC	27.0	1.00000	.016	Not significant

MUEBA and Advanced LSTM did not differ significantly in accuracy, precision, recall, or AUC-ROC on r6.2, but their F1-scores differed significantly (Table 6). An omnibus Friedman test confirmed an overall difference among all six models before any pairwise test was interpreted, for every metric on both datasets ($\chi^2(5)$ ranging from 35.54 to 43.71, all $p < .001$). Accordingly, MUEBA and Advanced LSTM on r6.2 are described in this manuscript as achieving comparable performance rather than formal statistical equivalence: they are indistinguishable on four of five metrics but differ significantly on F1-score, so their similarity should not be generalized across all evaluation criteria. This directly addresses the reviewer comments requesting statistical support for claims of equal performance.

Discussion:

Model Selection Implications:

The experimental results provide some practical implications for insider threat detection system design. The results of the Advanced LSTM on both datasets confirm this hypothesis as the model successfully captured multi-scale temporal relationships that are

essential for detecting malicious behavioral drift from non-malicious activity variation, similar to that reported by [11]and[16]. The performance of the Simplified LSTM is, however, comparable with that of the Original LSTM and RNN, with 1/4 of the number of parameters, meaning that the cost of the increased model complexity needs to be considered based on the latency and computing power requirements of the deployment environment.

MUEBA and the Advanced LSTM achieved the same rounded accuracy (97%) and AUC-ROC (0.99) on r6.2; Section 4.7 explains why this numerical similarity should be read as comparable performance rather than formal statistical equivalence.

The practical implications of this comparable performance are nonetheless significant: organizations with limited or no labeled insider-threat data—most enterprises, in fact—can use MUEBA to deploy a fully unsupervised system without compromising on detection performance as long as their environment matches the threat diversity seen in r6.2. Here we extend the work of Le and Zincir-Heywood [17] to a more complex, adaptive-thresholding framework.

Explainability in Security Operations:

From the analyses of SHAP and LIME (Section 4.6), the most influential features are also the features that are most easily accessible to security analysts in the standard log queries, leading to a direct link between model explanations and operational investigation workflows. This is consistent with the design principles proposed by[49] for deployable XAI in cybersecurity: explanations should reference artifacts that are directly observable in the operational environment, instead of abstractions that are represented within the model. The three subtypes of anomalies identified by the LIME analysis (bulk file access, reconnaissance browsing, and email-based social engineering) have implications for SOC triage prioritization - each anomaly type will require a different response to an investigation and escalation path through the identified population of flagged sessions.

This multi-subtype analysis is not possible using aggregate performance metrics or global SHAP values alone, as was demonstrated by the complementary value of local (LIME) and global (SHAP) explanation modalities, which are recommended by [9]but have not been studied together in the context of insider threat detection.

Dataset Observations and Limitations:

The steady improvement shown in the models from r5.2 to r6.2 further underscores the importance of choosing the right datasets for insider threat detection research, as highlighted by [40]in their extensive literature review. The richness of scenarios and the closer approximation of behavioral distributions in the r6.2 datasets to those seen in real-world threats is why it is used as the benchmark for evaluating generalization to real-world threats.

The CERT data used in this study is synthetic - that is, the users' behavior is simulated rather than observed from a live organization, which may not fully reflect the behavioral complexity and concept drift that may occur in a real insider threat scenario. Further, the threshold for the number of sessions considered 30 minutes that was validated empirically on the CERT dataset may prove insufficient for other organizations that have additional deployment cycles or other operational rhythms (e.g., 24/7 SOC or shift-based manufacturing). For practical SOC deployment, this means the 30-minute session window and the $\alpha = 0.6$ pre-scoring weight (Equation 1) should be treated as starting points tuned to CERT's simulated 9-to-5 office population rather than fixed constants: organizations adopting MUEBA-XAI should re-validate both the session-window length and the pre-scoring threshold against a sample of their own log traffic before relying on the reported 78% analyst-workload reduction, since a shift-based or always-on environment will generate a different session-length distribution than the CERT scenarios.

Future work should therefore prioritize (i) a pilot deployment on de-identified real organizational logs to test threshold transferability directly, (ii) periodic re-fitting of the

Isolation Forest and cluster-distance components to accommodate concept drift in user behavior over time, and (iii) an extension of the statistical-validation protocol in Section 4.7 to any re-tuned thresholds before they are trusted for triage decisions in a live SOC.

Conclusion and Future Work:

This paper introduced a Multimodal User and Entity Behavior Analytics framework (MUEBA-XAI) based on LSTM-based sequential modeling [11] and Isolation Forest anomaly isolation [12], along with a multi-component Enhanced MUEBA model [18], and provides both explainability layers of SHAP [9] and LIME [39]. The Advanced LSTM performs at 97.5–98.9% accuracy and 0.98–0.99 AUC-ROC, and MUEBA achieves 97% accuracy and 0.99 AUC-ROC, on the CERT Insider Threat Dataset r5.2 and r6.2 [13] (competitive with the published state of the art [35], [12] and with the unique advantage of generating interpretable and auditable anomaly explanations). File modifications, logon frequency and session duration are the most dominant anomaly indicators, remaining consistent for all models and datasets. LIME local explanations show three types of behavior anomalies, allowing analysts to follow up on the flagged anomalies. As reported in [40] the r6.2 dataset continuously outperforms the r5.2 dataset; this is because r6.2 has a more robust taxonomy of threats and more representative distributions of behaviors, as opposed to r5.2.

Three future work directions are identified: Real-time streaming inference through an Apache Kafka-based pipeline for sub-200 ms per-session latency for integration with live SIEM platforms like Splunk and Microsoft Sentinel; Federated learning variants, where training of the model can be done collaboratively without sharing sensitive behavioral logs to a central location, addressing the data-governance barriers to collaborative insider threat intelligence; Temporal SHAP analysis, extending per-feature attribution to per-time-step attribution within LSTM sequences, allowing for finer-grained behavioral forensics and root-cause analysis of flagged sessions

Conflict of interest:

The authors declare no conflict of interest.

References:

[1] 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.

[2] P. Manoharan, J. Yin, H. Wang, Y. Zhang, and W. Ye, "Insider threat detection using supervised machine learning algorithms," *Telecommun. Syst. 2023 874*, vol. 87, no. 4, pp. 899–915, Dec. 2023, doi: 10.1007/S11235-023-01085-3.

[3] M. AlSlaiman, M. I. Salman, M. M. Saleh, and B. Wang, "Enhancing false negative and positive rates for efficient insider threat detection," *Comput. Secur.*, vol. 126, p. 103066, Mar. 2023, doi: 10.1016/J.COSE.2022.103066.

[4] P. Lavanya, H. Anila Glory, and V. S. Shankar Sriram, "Mitigating Insider Threat: A Neural Network Approach for Enhanced Security," *IEEE Access*, vol. 12, pp. 73752–73768, 2024, doi: 10.1109/ACCESS.2024.3404814.

[5] P. Pal, P. Chattopadhyay, and M. Swarnkar, "Temporal feature aggregation with attention for insider threat detection from activity logs," *Expert Syst. Appl.*, vol. 224, p. 119925, Aug. 2023, doi: 10.1016/J.ESWA.2023.119925.

[6] R. A. Alsowail and T. Al-Shehari, "Techniques and countermeasures for preventing insider threats," *PeerJ. Comput. Sci.*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.938.

[7] T. T. & Z. Z. Fabien Charmet, Harry Chandra Tanuwidjaja, Solayman Ayoubi, Pierre-François Gimenez, Yufei Han, Houda Jmila, Gregory Blanc, "Explainable artificial intelligence for cybersecurity: a literature survey," *Ann. Telecommun.*, vol. 77, pp. 789–812, 2022, doi: https://doi.org/10.1007/s12243-022-00926-7.

[8] Z. Zhang, H. Al Hamadi, E. Damiani, C. Y. Yeun, and F. Taher, "Explainable Artificial Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104–93139, 2022, doi: 10.1109/ACCESS.2022.3204051.

[9] Scott Lundberg, Su-In Lee, "A Unified Approach to Interpreting Model Predictions," *arXiv:1705.07874*, 2017, [Online]. Available: https://arxiv.org/abs/1705.07874

[10] Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," *arXiv:1602.04938*, 2016, [Online]. Available: https://arxiv.org/abs/1602.04938

[11] M. Villarreal-Vasquez, G. Modelo-Howard, S. Dube, and B. Bhargava, "Hunting for Insider Threats Using LSTM-Based Anomaly Detection," *IEEE Trans. Dependable Secur. Comput.*, vol. 20, no. 1, pp. 451–462, Jan. 2023, doi: 10.1109/TDSC.2021.3135639.

[12] S. Zou, H. Sun, G. Xu, and R. Quan, "Ensemble strategy for insider threat detection from user activity logs," *Comput. Mater. Contin.*, vol. 65, no. 2, pp. 1321–1334, 2020, doi: 10.32604/CMC.2020.09649.

[13] M. Dosh, "Detecting insider threat within institutions using CERT dataset and different ML techniques," *Period. Eng. Nat. Sci.*, vol. 9, no. 2, pp. 873–884, May 2021, doi: 10.21533/PEN.V9.I2.791.

[14] F. Yuan, Y. Cao, Y. Shang, Y. Liu, J. Tan, and B. Fang, "Insider threat detection with deep neural network," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10860 LNCS, pp. 43–54, 2018, doi: 10.1007/978-3-319-93698-7_4/FIGURES/5.

[15] W. He, X. Wu, J. Wu, X. Xie, L. Qiu, and L. Sun, "Insider Threat Detection Based on User Historical Behavior and Attention Mechanism," *Proc. - 2021 IEEE 6th Int. Conf. Data Sci. Cyberspace, DSC 2021*, pp. 564–569, 2021, doi: 10.1109/DSC53577.2021.00089.

[16] P. Manoharan, W. Hong, J. Yin, H. Wang, Y. Zhang, and W. Ye, "Optimising Insider Threat Prediction: Exploring BiLSTM Networks and Sequential Features," *Data Sci. Eng. 2024 94*, vol. 9, no. 4, pp. 393–408, Nov. 2024, doi: 10.1007/S41019-024-00260-Z.

[17] D. C. Le and N. Zincir-Heywood, "Anomaly Detection for Insider Threats Using Unsupervised Ensembles," *IEEE Trans. Netw. Serv. Manag.*, vol. 18, no. 2, pp. 1152–1164, Jun. 2021, doi: 10.1109/TNSM.2021.3071928.

[18] M. N. Al-Mhiqani *et al.*, "A new intelligent multilayer framework for insider threat detection," *Comput. Electr. Eng.*, vol. 97, p. 107597, Jan. 2022, doi: 10.1016/J.COMPELECENG.2021.107597.

[19] W. Hong *et al.*, "A graph empowered insider threat detection framework based on daily activities," *ISA Trans.*, vol. 141, pp. 84–92, Oct. 2023, doi: 10.1016/J.ISATRA.2023.06.030.

[20] B. Bose, B. Avasarala, S. Tirthapura, Y. Y. Chung, and D. Steiner, "Detecting Insider Threats Using RADISH: A System for Real-Time Anomaly Detection in Heterogeneous Data Streams," *IEEE Syst. J.*, vol. 11, no. 2, pp. 471–482, Jun. 2017, doi: 10.1109/JSYST.2016.2558507.

[21] M. Garchery and M. Granitzer, "ADSAGE: Anomaly Detection in Sequences of Attributed Graph Edges applied to insider threat detection at fine-grained level," Jul. 2020, Accessed: Sep. 04, 2026. [Online]. Available: https://arxiv.org/pdf/2007.06985

[22] M. Singh, B. M. Mehtre, and S. Sangeetha, "User Behaviour based Insider Threat Detection in Critical Infrastructures," *ICSCCC 2021 - Int. Conf. Secur. Cyber Comput. Commun.*, pp. 489–494, May 2021, doi: 10.1109/ICSCCC51823.2021.9478137.

[23] C. Y. Y. and F. T. Z. Zhang, H. A. Hamadi, E. Damiani, "Explainable Artificial

Intelligence Applications in Cyber Security: State-of-the-Art in Research," *IEEE Access*, vol. 10, pp. 93104–93139, 2022, doi: 10.1109/ACCESS.2022.3204051.

[24] A. Nicolaou, S. Shiaeles, and N. Savage, "Mitigating Insider Threats Using Bio-Inspired Models," *Appl. Sci. 2020, Vol. 10, Page 5046*, vol. 10, no. 15, p. 5046, Jul. 2020, doi: 10.3390/APP10155046.

[25] "A Predictive User Behaviour Analytic Model for Insider Threats in Cyberspace | Request PDF." Accessed: Sep. 04, 2026. [Online]. Available: https://www.researchgate.net/publication/361650711_A_Predictive_User_Behaviour_Analytic_Model_for_Insider_Threats_in_Cyberspace

[26] F. Janjua, A. Masood, H. Abbas, and I. Rashid, "Handling Insider Threat Through Supervised Machine Learning Techniques," *Procedia Comput. Sci.*, vol. 177, pp. 64–71, Jan. 2020, doi: 10.1016/J.PROCS.2020.10.012.

[27] D. Zhu, X. Huang, N. Li, H. Sun, M. Liu, and J. Liu, "RAP-Net: A Resource Access Pattern Network for Insider Threat Detection," *Proc. Int. Jt. Conf. Neural Networks*, 2022, doi: 10.1109/IJCNN55064.2022.9892183.

[28] I. Benchaji, S. Douzi, B. El Ouahidi, and J. Jaafari, "Enhanced credit card fraud detection based on attention mechanism and LSTM deep model," *J. Big Data 2021 81*, vol. 8, no. 1, pp. 151-, Dec. 2021, doi: 10.1186/S40537-021-00541-8.

[29] B. Sharma, P. Pokharel, and B. Joshi, "User Behavior Analytics for Anomaly Detection Using LSTM Autoencoder-Insider Threat Detection," *ACM Int. Conf. Proceeding Ser.*, Jul. 2020, doi: 10.1145/3406601.3406610;

[30] D. Li, L. Yang, H. Zhang, X. Wang, L. Ma, and J. Xiao, "Image-Based Insider Threat Detection via Geometric Transformation," *Secur. Commun. Networks*, vol. 2021, no. 1, p. 1777536, Jan. 2021, doi: 10.1155/2021/1777536.

[31] X. Zuo, F. Yan, B. Hou, Z. Chen, and Y. Guo, "Insider Threat Detection Model Of Power System Based On Lstm-Attention".

[32] D. C. Le and N. Zincir-Heywood, "Exploring anomalous behaviour detection and classification for insider threat identification," *Int. J. Netw. Manag.*, vol. 31, no. 4, Jul. 2021, doi: 10.1002/NEM.2109.

[33] R. G. Gayathri, A. Sajjanhar, and Y. Xiang, "Adversarial Training for Robust Insider Threat Detection," *Proc. Int. Jt. Conf. Neural Networks*, 2022, doi: 10.1109/IJCNN55064.2022.9892059.

[34] K. Al tabash and J. Happa, "Insider-threat detection using Gaussian Mixture Models and Sensitivity Profiles," *Comput. Secur.*, vol. 77, pp. 838–859, Aug. 2018, doi: 10.1016/J.COSE.2018.03.006.

[35] C. Zhang, S. Wang, D. Zhan, T. Yu, T. Wang, and M. Yin, "Detecting Insider Threat from Behavioral Logs Based on Ensemble and Self-Supervised Learning," *Secur. Commun. Networks*, vol. 2021, no. 1, p. 4148441, Jan. 2021, doi: 10.1155/2021/4148441.

[36] R. B. Peccatiello, J. J. C. Gondim, and L. P. F. Garcia, "Applying One-Class Algorithms for Data Stream-Based Insider Threat Detection," *IEEE Access*, vol. 11, pp. 70560–70573, 2023, doi: 10.1109/ACCESS.2023.3293825.

[37] D. Ge, S. Zhong, and K. Chen, "Multi-source data fusion for insider threat detection using residual networks," *Proc. - 2022 3rd Int. Conf. Electron. Commun. Inf. Technol. CECIT 2022*, pp. 359–366, 2022, doi: 10.1109/CECIT58139.2022.00069.

[38] P. Moriano, J. Pendleton, S. Rich, and L. Jean Camp, "Stopping the insider at the gates: Protecting organizational assets through graph mining," *J. Wirel. Mob. Networks, Ubiquitous Comput. Dependable Appl.*, vol. 9, no. 1, pp. 4–29, Mar. 2018, doi: 10.22667/JOWUA.2018.03.31.004.

[39] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?' Explaining the

Predictions of Any Classifier," *NAACL-HLT 2016 - 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Demonstr. Sess.*, pp. 97–101, 2016, doi: 10.18653/v1/n16-3020.

[40] Mohammed Nasser Al-Mhiqani, Rabiah Ahmad, "A Review of Insider Threat Detection: Classification, Machine Learning Techniques, Datasets, Open Challenges, and Recommendations," *Appl. Sci.*, vol. 10, no. 15, p. 5208, 2020, doi: https://doi.org/10.3390/app10155208.

[41] R. Yumlembam, B. Issac, S. M. Jacob, L. Yang, and D. Krishnan, "Insider Threat Detection Using GCN and Bi-LSTM with Explicit and Implicit Graph Representations," *IEEE Trans. Artif. Intell.*, Jul. 2025, doi: 10.1109/TAI.2025.3647418.

[42] Vincent Zibi Mohale, Ibidun Christiana Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Front. Artif. Intell.*, vol. 8, 2025, doi: https://doi.org/10.3389/frai.2025.1526221.

[43] K. S. Alketbi and A. Mehmood, "A Comprehensive Survey of Explainable Artificial Intelligence Techniques for Malicious Insider Threat Detection," *IEEE Access*, vol. 13, pp. 121772–121798, 2025, doi: 10.1109/ACCESS.2025.3587114.

[44] R. Grzeczkowicz, C. Neal, N. Baghalizadeh-Moghadam, N. B. Cuppens, and F. Cuppens, "Classifying Insider Threat Scenarios Through Explainable Articial Intelligence," pp. 157–172, 2025, doi: 10.1007/978-3-031-89350-6_10;

[45] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing Data Granularity Levels for Insider Threat Detection Using Machine Learning," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 1, pp. 30–44, Mar. 2020, doi: 10.1109/TNSM.2020.2967721.

[46] T. Al-shehari and R. A. Alsowail, "An Insider Data Leakage Detection Using One-Hot Encoding, Synthetic Minority Oversampling and Machine Learning Techniques," *Entropy 2021, Vol. 23, Page 1258*, vol. 23, no. 10, p. 1258, Sep. 2021, doi: 10.3390/E23101258.

[47] M. Nasser Al-Mhiqani, R. Ahmed, Z. A. Zainal Abidin, and S. N. Isnin, "An Integrated Imbalanced Learning and Deep Neural Network Model for Insider Threat Detection," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 1, pp. 573–577, 2021, doi: 10.14569/IJACSA.2021.0120166.

[48] T. Al-Shehari and R. A. Alsowail, "Random resampling algorithms for addressing the imbalanced dataset classes in insider threat detection," *Int. J. Inf. Secur. 2022 223*, vol. 22, no. 3, pp. 611–629, Dec. 2022, doi: 10.1007/S10207-022-00651-1.

[49] F. Charmet *et al.*, "Explainable artificial intelligence for cybersecurity: a literature survey," *Ann. Telecommun. 2022 7711*, vol. 77, no. 11, pp. 789–812, Oct. 2022, doi: 10.1007/S12243-022-00926-7.

Copyright © by authors and 50Sea. This work is licensed under Creative Commons Attribution 4.0 International License.