

AI Powered User and Entity Behavior Analytics Platform for Intelligent Threat Detection and Insider Risk Management

K. Mahesh Raj¹, A. Sreekar², B. Akhil³, Ch. Vijay⁴, Ch. Shiva Sai Raj⁵, D. Sai Ram⁶

¹Faculty Guide, ²³⁴⁵⁶ B.Tech CSE Students, Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, India

Abstract - User and Entity Behavior Analytics is an important approach for identifying suspicious activity that may not be visible through fixed security rules alone. This paper presents a practical UEBA platform designed to model normal user behavior, identify deviations, and provide contextual risk information for security analysts. The proposed system uses the CERT Insider Threat Dataset r6.2 as its primary user activity source. The workflow includes data preprocessing, behavioral feature engineering, individual user baselines, peer group comparison, Isolation Forest based anomaly detection, contextual risk scoring, explainable alert generation, and event correlation. A simple analyst dashboard is planned to present high risk users, anomaly scores, behavioral deviations, and correlated alerts. The project is intentionally designed as a manageable Python based system rather than a large distributed security platform. Evaluation will compare a rule based baseline, Isolation Forest, Isolation Forest with behavioral baselines, and the complete contextual workflow using measurable detection metrics. At the current development stage, implementation and experiments are still being completed, so no unverified performance values are reported. The final system will be used to study whether combining individual behavior, peer context, anomaly detection, and explainable risk information can provide more useful security alerts.

Key Words: User and Entity Behavior Analytics, Insider Threat Detection, Anomaly Detection, Isolation Forest, Behavioral Baseline, Risk Scoring, Explainable Security Alerts

1. INTRODUCTION

Modern organizations generate large volumes of authentication, file, email, web, endpoint, and network activity. Traditional security monitoring commonly depends on predefined signatures and rules. These mechanisms remain useful, but they may struggle when an account is used in a way that is unusual for that particular user or department. User and Entity Behavior Analytics addresses this problem by learning behavioral patterns and identifying deviations from expected activity.

UEBA is particularly relevant to insider threat and compromised account scenarios because suspicious activity may use valid credentials. Instead of relying only

on whether an event is known to be malicious, a UEBA system can consider when an activity occurred, how frequently it occurred, whether it is normal for the user, and whether it differs from similar users. This provides additional context for analysts investigating security events.

The proposed project develops a practical UEBA workflow using the CERT Insider Threat Dataset r6.2. The system converts raw activity into behavioral features, establishes individual and peer group baselines, applies Isolation Forest for unsupervised anomaly detection, combines multiple signals into a configurable risk score, and produces explainable alerts. The project focuses on a clear and reproducible implementation that can be evaluated experimentally.

2. PROBLEM STATEMENT

Security monitoring systems can generate large numbers of events while providing limited context about whether an event is normal for a particular user. A login outside working hours, an increase in file activity, or an unusual sequence of events may be legitimate in one context and suspicious in another. The problem addressed in this project is the design of a practical system that can transform user activity into behavioral features, learn normal patterns, detect anomalies, consider peer context, assign a transparent risk score, and provide reasons that help an analyst investigate an alert.

3. OBJECTIVES

- To preprocess user activity data and convert heterogeneous events into consistent behavioral records.
- To extract user level features covering authentication, file, email, web, removable media, and general activity where the dataset supports them.
- To establish an individual behavioral baseline using historical activity.
- To compare user behavior with available peer group information such as department or role.
- To apply Isolation Forest as the primary unsupervised anomaly detection method.

- To calculate a configurable contextual risk score on a zero to one hundred scale.
- To generate explainable alerts containing the behavioral factors that contributed to the alert.
- To correlate related events within a configurable time window and present the information through an analyst dashboard.
- To evaluate the complete workflow against simpler detection baselines using reproducible experiments.

4. PROPOSED METHODOLOGY

The proposed methodology follows a sequential pipeline from telemetry collection to analyst investigation. The design intentionally keeps the core implementation compact so that each stage can be tested independently.



Fig -1: Proposed UEBA processing architecture

4.1 Data Collection and Preprocessing

The CERT Insider Threat Dataset r6.2 is selected as the primary dataset because it contains user oriented activity associated with insider threat scenarios. Relevant sources include authentication activity, file activity, removable device activity, email activity, web activity, and organizational information. The preprocessing stage reads the available files, inspects their actual schema, parses timestamps, normalizes identifiers and event types, handles missing values, and removes obvious duplicates where appropriate. The processed records are stored in a consistent format for feature generation.

4.2 Behavioral Feature Extraction

Raw events are aggregated into user level behavioral features. Authentication features can include login frequency, login timing, failed logins when available, active hours, weekend activity, and workstation diversity. File features can include file access volume, unique files, file types, and removable media activity. Email and web features can include message volume, recipient diversity, external activity where identifiable, domain diversity, and overall event frequency. Only features that can be derived from the actual dataset schema are included.

4.3 Individual Behavioral Baseline

For each user, historical activity is used to estimate normal behavior. Median, mean, standard deviation, percentile ranges, and rolling statistics may be used depending on the feature. Current activity is compared with the user's historical pattern to generate deviation measures. This allows the system to distinguish a high activity level that is normal for one user from the same activity level that is unusual for another.

4.4 Peer and Contextual Analysis

A user can also be compared with a peer group when organizational information is available. Peer groups may be formed from department, role, team, or similar attributes present in the dataset. Peer deviation features measure whether a user's login timing, file activity, email volume, or overall activity differs substantially from comparable users. Peer comparison is treated as contextual evidence rather than proof of malicious behavior.

4.5 Isolation Forest Anomaly Detection

Isolation Forest is used as the primary unsupervised anomaly detection method. The algorithm isolates observations by randomly partitioning feature space, making unusual observations easier to isolate. A training and testing strategy based on time ordering is preferred to reduce leakage from future activity. The model output is converted into an anomaly measure that can be combined with behavioral and contextual evidence.

4.6 Contextual Risk Scoring

The risk scoring layer combines the anomaly signal with individual behavioral deviation, peer deviation, unusual timing, activity volume, and other supported contextual indicators. The resulting score is normalized to a zero to one hundred scale. Initial severity bands are configurable as low, medium, high, and critical. The weights and thresholds are treated as engineering parameters and must be validated experimentally rather than presented as universally optimal values.

4.7 Explainable Alerts and Event Correlation

Each generated alert contains the user identifier, timestamp, anomaly information, risk score, severity, and the main behavioral factors that contributed to the alert. Related suspicious events can be grouped within a configurable time window. For example, an unusual login followed by abnormal file activity and removable media activity can be presented as one correlated incident. The system reports suspicious behavior rather than automatically assigning malicious intent.

5. IMPLEMENTATION ENVIRONMENT

The implementation is designed around Python and commonly available data science libraries. Pandas and NumPy are used for data processing, scikit learn is used for Isolation Forest and evaluation utilities, and Streamlit is planned for the analyst dashboard. Matplotlib can be used to generate research figures. The project is organized into separate modules for ingestion, preprocessing, feature engineering, baselines, peer analysis, anomaly detection, risk scoring, alerting, correlation, and visualization.

Implementation summary: CERT r6.2 is the primary dataset; CSE-CIC-IDS2018 is optional and separate. The core stack uses Python, Pandas, NumPy, scikit learn, Matplotlib, and Streamlit. The implementation contains preprocessing, feature engineering, individual behavioral baselines, peer analysis, Isolation Forest anomaly detection, configurable contextual risk scoring, explainable alerts, rule based event correlation, and an analyst dashboard.

6. SYSTEM MODELING

The system can be viewed as five logical layers. The data layer contains raw and processed telemetry. The feature layer converts events into user level behavioral observations. The analytics layer performs baseline comparison and anomaly detection. The decision layer combines the signals into contextual risk scores and explainable alerts. The presentation layer provides the analyst dashboard and investigation views. This separation makes individual modules easier to test and replace.

A typical investigation begins with a user or time period selected by an analyst. The system retrieves the user's aggregated behavior, compares it with historical and peer patterns, displays the anomaly score and risk score, and lists the events that contributed to the alert. This workflow is intended to reduce the amount of raw event data an analyst must manually inspect.

7. DATASET AND EVALUATION PLAN

CERT r6.2 is the primary evaluation source because its synthetic enterprise environment provides user identities and activity relevant to insider threat research. The dataset contains multiple activity types that can be transformed into behavioral observations. CSE-CIC-IDS2018 is treated separately because it focuses on network traffic and does not provide the same user attribution required for direct UEBA evaluation.

The preferred evaluation strategy is chronological splitting so that historical activity is used to model

behavior and later activity is used for testing. Evaluation scenarios include abnormal login timing, unusual resource or file access, unusual activity volume, peer deviation, and other ground truth scenarios supported by the dataset.

7.1 Evaluation Metrics

Where ground truth labels support the calculation, the project will report precision, recall, F1 score, false positive rate, ROC AUC, and PR AUC. Operational measures such as alert volume, detection latency, and recall at a fixed alert budget may also be reported when they can be measured consistently. Metrics will be generated from actual experiments only.

7.2 Planned Comparison

The proposed workflow will be compared with simpler baselines. B1 uses rule or threshold based detection. B2 uses Isolation Forest without contextual behavioral scoring. B3 combines Isolation Forest with behavioral baselines. B4 is the complete workflow including individual baseline, peer analysis, anomaly detection, contextual risk scoring, explainable alerts, and event correlation. An ablation study can be added if sufficient experimental time is available.

Planned comparison: B1 rule based detection, B2 Isolation Forest, B3 Isolation Forest with behavioral baseline, and B4 the proposed contextual workflow. Precision, recall, F1 score, false positive rate, ROC AUC, PR AUC, alert volume, and detection latency will be reported only when supported by actual experiments.

4.8 Core Processing Pseudocode

OUTPUT: risk scores, explainable alerts, correlated events, evaluation files.

16. Present investigation results in the analyst dashboard.
15. Store alerts, scores, and evaluation outputs.
14. Correlate related alerts within the configured time window.
13. Generate an alert with contributing behavioral reasons.
12. Assign a configurable severity band.
11. Normalize contextual risk to the configured 0 to 100 scale.
10. Combine anomaly, baseline, peer, timing, and volume signals.
9. Generate anomaly scores for test observations.
8. Train Isolation Forest on the training feature matrix.
7. Compute peer deviation for supported behavioral features.
6. Build peer groups from available organizational attributes.

5. Compute individual deviation from the historical baseline.
4. Build historical baseline for each user using the training period.
3. Aggregate events into user level behavioral features.
2. Normalize user IDs, event types, timestamps, and missing values.
1. Load and inspect available event sources and timestamps.

INPUT: CERT activity records

8. RESULTS AND DISCUSSION

At the present development stage, the implementation and experimental evaluation are still being completed. Therefore, performance values are intentionally not reported in this manuscript draft. Once the pipeline is executed on the selected dataset, the results section will include the measured comparison between the baseline methods and the proposed workflow. The discussion will focus on detection quality, false positive behavior, alert volume, contextual explanations, and the types of anomalies that are detected.

The final results should also include screenshots of the working dashboard, representative alert records, feature distributions, and evaluation plots generated from the actual experiment output. No synthetic or manually selected values should be presented as measured results.

9. LIMITATIONS

- CERT is a synthetic insider threat dataset and may not represent all real enterprise environments.
- The usefulness of each feature depends on the activity types and fields actually present in the dataset.
- Behavioral thresholds and risk weights require validation and may affect false positive rates.
- User behavior can change over time, creating concept drift that a static baseline may not capture.
- Rule based event correlation may miss complex multi stage behavior.
- The dashboard is intended for project and research demonstration and is not a replacement for a production SIEM or SOC platform.

10. FUTURE SCOPE

Future development can extend the project with streaming telemetry, adaptive baselines, concept drift detection, autoencoders or sequence models, graph based behavior modeling, supervised or ensemble classifiers, and explainability methods such as SHAP or LIME. Integration with SIEM and SOAR platforms can support automated investigation and response. Additional work

can evaluate the framework on real enterprise telemetry, privacy preserving or federated learning settings, and analyst in the loop feedback.

11. CONCLUSION

This paper presents a practical design for a UEBA platform that combines individual behavioral baselines, peer context, Isolation Forest anomaly detection, contextual risk scoring, explainable alerts, and event correlation. The approach is intended to provide security analysts with more context than a simple event or threshold based alert. The project uses CERT Insider Threat Dataset r6.2 as its primary user activity source and follows a modular Python based implementation. The final contribution will depend on the completed implementation and measured experiments. Performance claims will be added only after the evaluation pipeline produces reproducible results.

ACKNOWLEDGEMENT

The authors express their gratitude to the Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Hyderabad, and to the project guide for their guidance and support during the development of this work.

REFERENCES

- 1) Z. Wei, U. Rauf, and F. Mohsen, "E-Watcher: insider threat monitoring and detection for enhanced security," *Annals of Telecommunications*, vol. 79, pp. 819–831, 2024, doi: 10.1007/s12243-024-01023-7.
- 2) S. Song, N. Gao, Y. Zhang, and C. Ma, "BRITD: behavior rhythm insider threat detection with time awareness and user adaptation," *Cybersecurity*, vol. 7, 2024, Art. no. 2, doi: 10.1186/s42400-023-00190-9.
- 3) A. M. Sivakrishna, R. Mohan, and V. Rohini, "An efficient insider threat detection framework using Bayesian-optimized XGBoost," *Security and Privacy*, vol. 8, no. 6, 2025, Art. no. e70122, doi: 10.1002/spy2.70122.
- 4) K. S. Alketbi and A. Mehmood, "A Comprehensive Survey of Explainable Artificial Intelligence Techniques for Malicious Insider Threat Detection," *IEEE Access*, vol. 13, 2025, pp. 121772–121798, doi: 10.1109/ACCESS.2025.3587114.
- 5) X. Tao, J. Liu, Y. Yu, H. Zhang, and Y. Huang, "An insider threat detection method based on improved Test-Time Training model," *High-Confidence Computing*, vol. 5, no. 1, 2025, Art. no. 100283, doi: 10.1016/j.hcc.2024.100283.
- 6) T. Tian, C. Zhang, B. Jiang, H. Feng, and Z. Lu, "Insider threat detection for specific threat

scenarios,” Cybersecurity, vol. 8, 2025, Art. no. 17, doi: 10.1186/s42400-024-00321-w.

- 7) T. Patel and S. S. Iyer, “SiaDNN: Siamese deep neural network for anomaly detection in user behavior,” Knowledge-Based Systems, vol. 324, 2025, Art. no. 113769, doi: 10.1016/j.knosys.2025.113769.
- 8) Y. Qi, C. Yan, Z. Wang, C. Zhang, S. Liu, Z. Lu, and B. Jiang, “ATHITD: Attention-based temporal heterogeneous graph neural network for insider threat detection,” Computers & Security, vol. 157, 2025, Art. no. 104587, doi: 10.1016/j.cose.2025.104587.
- 9) J. Childress and M. Min, “A Review of User Account Anomaly Detection and Insider Threat Detection Techniques,” in 2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC), 2026, doi: 10.1109/ICAIC67076.2026.11395680.
- 10) B. Bhoi, G. Jethava, and S. Jethava, “A Comprehensive Review of Anomaly Detection Techniques for Insider Threat Identification,” in Proceedings of the 2026 International Conference on AI-Driven Smart Systems and Ubiquitous Computing (ICAUC), 2026, pp. 1050–1056, doi: 10.1109/ICAUC68182.2026.11440999.

8.1 Representative System Output

The UEBA platform is designed to convert model and behavioral signals into an analyst oriented output. A representative alert contains the user identifier, contextual risk score, severity, anomaly signal, behavioral reasons, and correlated events. The values shown below are output fields rather than measured experimental results. Actual values will be populated after the completed implementation is executed on the selected dataset.

Contextual risk score output scale

LOW	MEDIUM	HIGH	CRITICAL
-----	--------	------	----------

0

100

Configurable severity bands; final thresholds must be validated experimentally.

Fig -2: Contextual risk score output scale used by the proposed system

UEBA ALERT – REPRESENTATIVE OUTPUT FORMAT	
User:	U-XXXX
Risk Score:	[actual score after evaluation]
Severity:	[LOW / MEDIUM / HIGH / CRITICAL]
Anomaly Signal:	[Isolation Forest output]
Behavioral Reason 1:	[deviation from individual baseline]
Behavioral Reason 2:	[deviation from peer group]
Behavioral Reason 3:	[unusual activity/timing]
Correlated Events:	[actual correlated events]
<i>Illustrative output layout only – values are intentionally not fabricated.</i>	

Fig -3: Representative explainable alert output format

8.2 Output Summary Table

Table 3 describes the principal outputs expected from the implemented pipeline. It is included to define the result structure without introducing unverified performance measurements.

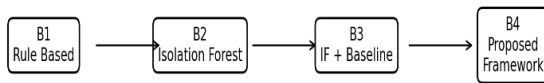
Table -3: Representative UEBA output and evaluation fields

Output	Example / Format	Purpose
User Risk Score	0–100	Prioritize investigation using contextual evidence
Severity	LOW / MEDIUM / HIGH / CRITICAL	Communicate the urgency of an alert
Anomaly Signal	Isolation Forest score	Represent unsupervised behavioral abnormality
Behavioral Reasons	Baseline / peer deviations	Explain why the activity is unusual
Correlated Events	Timestamped event group	Provide temporal investigation context
Evaluation Metrics	Precision, Recall, F1, FPR, ROC AUC, PR AUC	Measure detection performance after testing

8.3 Experimental Result Presentation

The final experimental results will compare the four detection configurations defined in the evaluation plan. The comparison is intended to show how behavioral baselines and contextual signals affect detection quality and alert generation. The final paper will replace the current output framework with measured values obtained from the same test split and evaluation procedure. This approach avoids reporting performance numbers that have not yet been verified.

Evaluation output structure



Compare using actual precision, recall, F1, FPR, ROC AUC and PR AUC after experiments.

Fig -4: Planned comparative evaluation output

Table -4: Comparative output structure before measured evaluation

Configuration	Expected Output	Measured Result
B1 Rule Based	Rule or threshold alerts	To be measured
B2 Isolation Forest	Unsupervised anomaly alerts	To be measured
B3 IF + Baseline	Anomaly plus individual deviation	To be measured
B4 Proposed Framework	Anomaly, baseline, peer context and explanation	To be measured

8.4 Illustrative Result Output Screens

The following figures show the intended form of the implemented outputs. They are illustrative execution designs generated for the manuscript and are not measured experimental results. Actual screenshots and values should replace them after the project pipeline is executed on CERT r6.2.

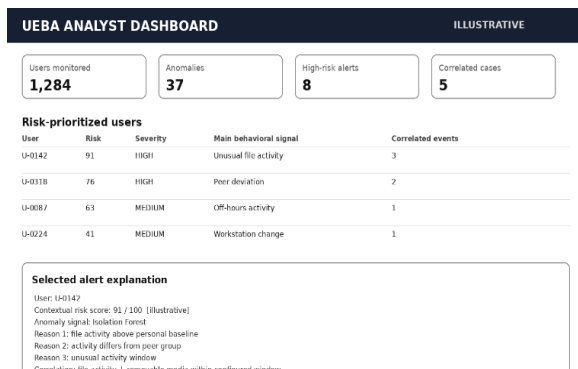


Fig -8: Illustrative UEBA analyst dashboard result output

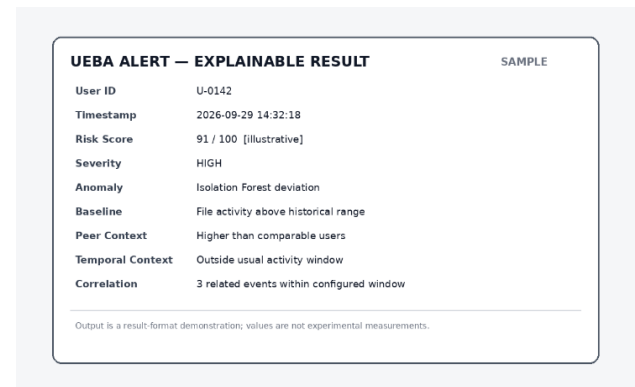


Fig -9: Illustrative explainable alert result output

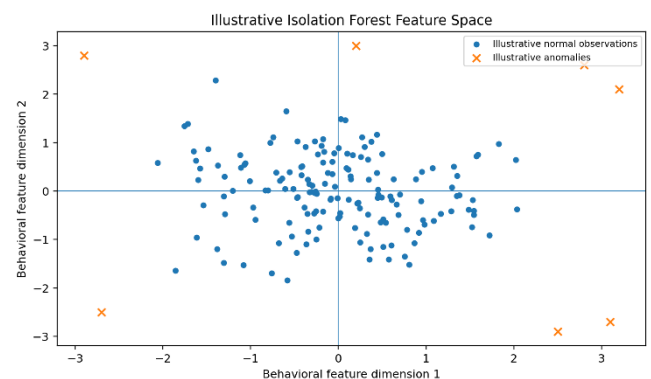


Fig -10: Illustrative anomaly distribution from behavioral feature space



Fig -11: Illustrative UEBA pipeline execution output

Table -6: Illustrative result fields generated by the proposed workflow

These outputs demonstrate the expected presentation format only. They must not be interpreted as evidence of detection performance until the actual implementation produces reproducible measurements.

APPENDIX A. REPRESENTATIVE UEBA RESULT OUTPUTS

This appendix records the form in which the completed UEBA implementation will present its outputs. The figures are intentionally presented as representative output designs rather than measured experimental results. This distinction is important because the implementation and evaluation are still being completed. Once the dataset pipeline has been executed, the same locations can contain actual screenshots and measured values without changing the surrounding manuscript structure.

A.1 User Risk Investigation

The analyst view is expected to begin with a list of users ordered by contextual risk. Selecting a user should display the user's recent activity, historical behavioral baseline, peer comparison, anomaly score, risk score, severity, and the events that contributed to the alert. This output is intended to support investigation rather than to automatically classify a person as malicious.

A.2 Alert Explanation

The explanation layer is a central part of the proposed system. Instead of returning only an anomaly label, the system records the behavioral evidence that contributed to the alert. Examples include activity outside the user's usual time range, unusually high file activity, deviation from a peer group, or a combination of related events. The final implementation will populate these fields directly from the feature and scoring modules.

UEBA ALERT – REPRESENTATIVE OUTPUT FORMAT	
User:	U-XXXX
Risk Score:	[actual score after evaluation]
Severity:	[LOW / MEDIUM / HIGH / CRITICAL]
Anomaly Signal:	[Isolation Forest output]
Behavioral Reason 1:	[deviation from individual baseline]
Behavioral Reason 2:	[deviation from peer group]
Behavioral Reason 3:	[unusual activity/timing]
Correlated Events:	[actual correlated events]
<i>Illustrative output layout only – values are intentionally not fabricated.</i>	

Fig -5: Representative UEBA alert output used for implementation validation

APPENDIX B. EXPERIMENTAL RESULT RECORDING

The experimental stage will use the same preprocessing and test split for each comparison method. This prevents changes in the input data from being confused with

changes caused by the detection method. The rule based baseline, Isolation Forest baseline, baseline enhanced Isolation Forest, and proposed contextual framework will therefore be evaluated under a common procedure.

B.1 Result Recording Table

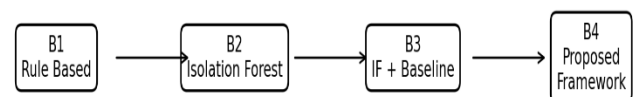
Table -5: Experimental result recording plan

Experiment	Configuration	Primary Output	Status
E1	Rule based baseline	Alert count and detection metrics	Pending execution
E2	Isolation Forest	Anomaly score and detection metrics	Pending execution
E3	IF + behavioral baseline	Anomaly plus deviation metrics	Pending execution
E4	Full UEBA framework	Risk score, explanations and detection metrics	Pending execution
E5	Ablation, if feasible	Contribution of contextual components	Optional

B.2 Evaluation Outputs

For each experiment, the evaluation script will save a machine readable metrics file and generated figures. The final paper can then report precision, recall, F1 score, false positive rate, ROC AUC, PR AUC, alert volume, and detection latency where the underlying labels and timestamps support those calculations. Confusion matrices, ROC curves, precision recall curves, anomaly distributions, and risk score distributions can be generated from the same output files.

Evaluation output structure



Compare using actual precision, recall, F1, FPR, ROC AUC and PR AUC after experiments.

Fig -6: Comparative evaluation output structure

APPENDIX C. IMPLEMENTATION VALIDATION AND REPRODUCIBILITY

The final UEBA system should be reproducible from the project repository without manually modifying generated results. The recommended execution order is dataset inspection, preprocessing, feature generation, baseline construction, model training, risk scoring, alert generation, evaluation, and dashboard execution. Each stage should produce an inspectable output so that errors can be isolated before the next stage is executed.

C.1 Validation Checklist

- Verify that all dataset files are read using their actual schema and timestamps are parsed correctly.
- Verify that feature generation does not use information from the future test period.
- Verify that user and peer baselines are built only from the intended historical period.
- Verify that Isolation Forest receives the intended feature matrix and that anomaly scores are stored.
- Verify that the contextual risk score remains within the configured zero to one hundred range.
- Verify that every high risk alert contains at least one explainable behavioral reason.
- Verify that event correlation groups events only within the configured time window.
- Verify that evaluation metrics are generated from actual predictions and ground truth labels.
- Verify that dashboard values match the stored model and alert outputs.
- Record the software versions, dataset version, feature list, split strategy, model parameters, and experiment date.

C.2 Final Result Replacement

The representative values and output formats in this appendix must not be interpreted as empirical performance. Before final submission, the project team should replace the pending result fields with values produced by the completed implementation, insert screenshots captured from the working dashboard, and update the Results and Discussion section with the corresponding interpretation. This keeps the manuscript aligned with the actual system and avoids unsupported performance claims.

Contextual risk score output scale

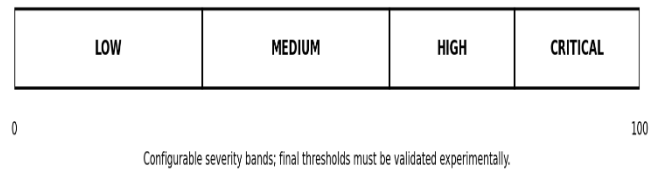


Fig -7: Risk score interpretation used for the contextual scoring module

Output	Example format	Purpose
Risk score	0 to 100	Contextual prioritization
Anomaly score	Isolation Forest output	Behavioral abnormality signal
Baseline deviation	Historical deviation	User-specific context
Peer deviation	Peer-group comparison	Contextual evidence
Alert explanation	Contributing factors	Analyst investigation
Correlation	Related events in time window	Incident-level context