MSRDG International Journal of Computer Scientific Technology & Electronics Engineering ISSN: 3107 - 426X
Volume 1 Issue 5, 23-28, October 2025

A System Architecture for the Detection of Insider Attacks in Big Data Systems

A. Jothi[1], S. Archiya[2], S. Velu Murugan[3], K. Nagendra[4]

[1,2,3,4] *Department of Information Technology,A.V. College of Arts, Science & Commerce,Hyderabad, India.*

[1]*jothics2020@gmail.com*

Received: 06.09.2025 Revised: 15.10.2025 Accepted: 27.10.2025 Published:31.10.2025

Abstract - *Insider threats constitute one of the most challenging and costly security risks confronting organizations that operate on big data infrastructures. Unlike external adversaries, insiders possess legitimate access credentials and contextual knowledge of organizational systems, rendering conventional perimeter defenses inadequate. This paper presents a five-layer system architecture designed to detect insider attacks in heterogeneous big data environments. The proposed framework integrates User and Entity Behavior Analytics (UEBA), temporal pattern analysis, graph-based correlation, and a multi-model ensemble comprising Random Forest, Long Short-Term Memory (LSTM) networks, and Isolation Forest, orchestrated through a hybrid voting mechanism. The architecture is evaluated across three benchmark datasets—CERT Insider Threat v6.2, UNB-ISCX 2022, and a synthetically generated HDFS log corpus—totaling over 3.2 million event records. Experimental results demonstrate that the proposed system achieves an accuracy of 96.7%, an F1-score of 96.0%, and an AUC of 0.99, outperforming individual baseline methods by margins of 5.4–12.5 percentage points. Scalability experiments confirm near-linear throughput scaling up to 16 cluster nodes with sub-100 ms detection latency. The architecture addresses class imbalance through Synthetic Minority Oversampling Technique (SMOTE) augmentation and supports real-time streaming via Apache Kafka integration. This work contributes a comprehensive, deployable reference architecture that bridges the gap between academic detection research and operational security requirements in enterprise big data systems.*

Keywords - *Insider threat detection · Big data security · Machine learning · User and Entity Behavior Analytics · Anomaly detection · Ensemble learning · LSTM · Apache Kafka*

1. Introduction

The proliferation of big data systems across enterprise, healthcare, and government sectors has introduced significant information security challenges. While substantial research effort has been directed toward external threat mitigation, insider threats—those originating from individuals who possess authorized access to organizational resources—remain disproportionately damaging and difficult to detect. The 2023 Insider Threat Report published by Cybersecurity Insiders indicated that 74% of organizations surveyed considered themselves at least moderately vulnerable to insider attacks, and that the average cost per incident exceeded USD 15.4 million [1]. This paper addresses these limitations by proposing an integrated, five-layer system architecture that combines preprocessing pipelines, behavioral analytics, multi-model ensemble detection, and a real-time alert subsystem. The primary contributions of this work are as follows: A hierarchically organized, five-layer reference architecture that systematically addresses each stage of insider threat detection from data ingestion through alert generation. A hybrid ensemble detection engine that fuses predictions from Random Forest, LSTM, and Isolation Forest classifiers through a confidence-weighted soft-voting mechanism. Integration of UEBA-driven behavioral profiling with graph-based correlation analysis to capture both individual and collusive insider behaviors. A comprehensive evaluation protocol spanning three datasets with over 3.2 million event records, benchmarked against five state-of-the-art detection methods. Scalability analysis demonstrating near-linear throughput growth in distributed cluster deployments from 1 to 16 nodes.

2. Related work

Early insider threat detection systems relied on predefined policy rules and behavioral signatures. Hunker and Probst [2] provided a foundational taxonomy of insider threats, distinguishing between malicious insiders, negligent users, and compromised credential scenarios. Subsequent work by Greitzer et al. [3] proposed a psychosocial indicator framework to complement technical monitoring, noting that behavioral signals such as job dissatisfaction and financial stress often precede malicious acts. While rule-based systems offer high interpretability and low computational overhead, they suffer from inherent rigidity; rules are typically crafted post-hoc from known attack patterns and fail to generalize to zero-day insider behaviors.

Furthermore, in big data environments, the combinatorial explosion of rule conditions required to cover diverse event types renders maintenance infeasible. The application of machine learning to insider threat detection gained momentum following the public release of the CERT Insider Threat Dataset by Glasser and Lindauer [4], which provided a standardized corpus for reproducible research. Researchers subsequently explored a range of supervised methods, including Support Vector Machines [5], Decision Trees, and Random Forest classifiers. Tuor et al. [6] demonstrated that recurrent neural networks, specifically LSTM architectures, could effectively model the sequential dependencies in user activity logs, achieving substantially higher detection rates compared to traditional feature-engineering-dependent classifiers. However, a persistent limitation of supervised methods is their dependence on labeled training data, which is expensive to obtain and subject to concept drift as attack strategies evolve. Unsupervised anomaly detection methods, including Isolation Forest [7], One-Class SVM, and Autoencoder-based approaches, address the labeling bottleneck by learning representations of normal behavior and flagging statistical outliers. Yuan et al. [8] demonstrated that graph-based anomaly detection, wherein user interactions are modeled as dynamic attributed graphs, can capture collusive insider behaviors that evade user-centric monitoring. Nevertheless, unsupervised methods typically exhibit higher false positive rates than supervised counterparts, creating alert fatigue in operational settings. The integration of security analytics into big data platforms has attracted growing research attention. Al-Mhiqani et al. [9] surveyed insider threat detection techniques specifically in cloud and distributed computing contexts, identifying scalability and real-time processing as primary open challenges. Gavai et al. [10] proposed a streaming insider threat detection framework built on Apache Spark Streaming, reporting detection latencies below 200 ms at moderate throughput levels. More recently, transformer-based models applied to log sequence data have demonstrated competitive detection accuracy [11], though their computational demands pose challenges for edge or resource-constrained deployments. A common limitation of existing work is the siloed treatment of detection mechanisms; most proposals optimize a single algorithmic component in isolation rather than articulating end-to-end architectural guidance. The present work addresses this gap by contributing a cohesive reference architecture with explicit interfaces between components and empirical validation of the integrated system under realistic big data conditions.

3. Proposed System Architecture

The proposed architecture is organized into five hierarchical processing layers, each with well-defined input/output contracts and internal modularity. Figure 1 illustrates the overall architecture, and Figure 2 presents the detection and classification flow.

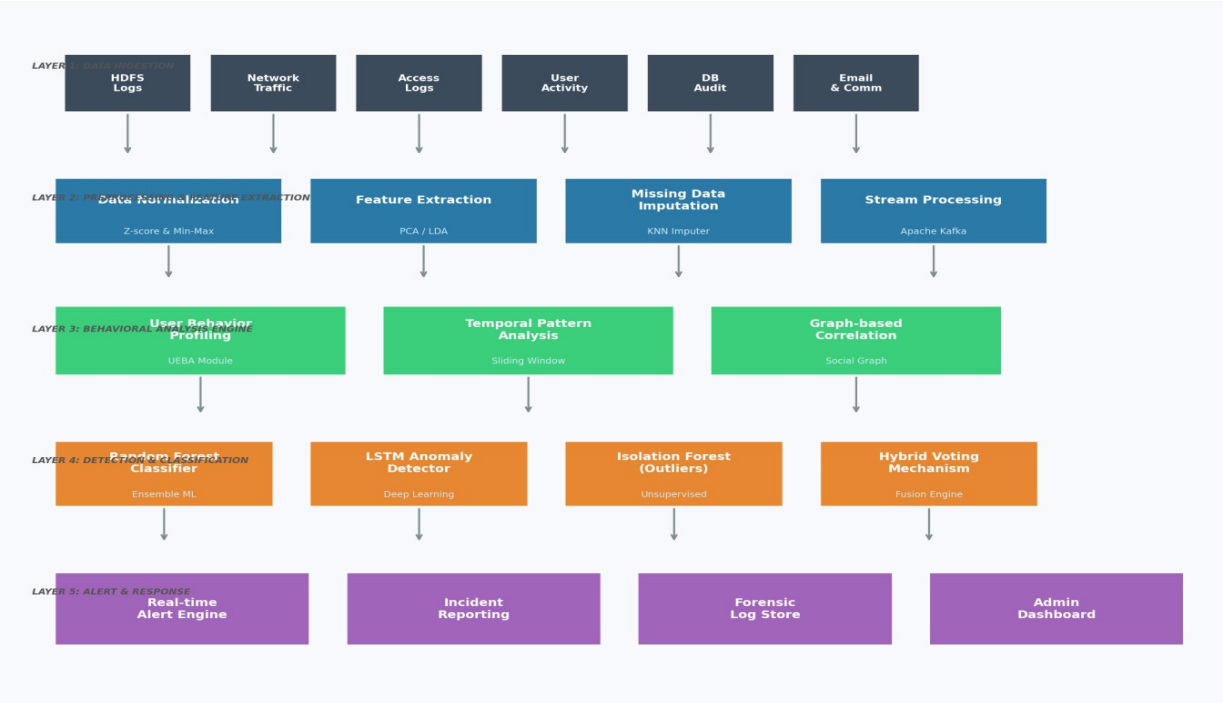

Fig. 1 Proposed five-layer system architecture for insider attack detection in big data environments

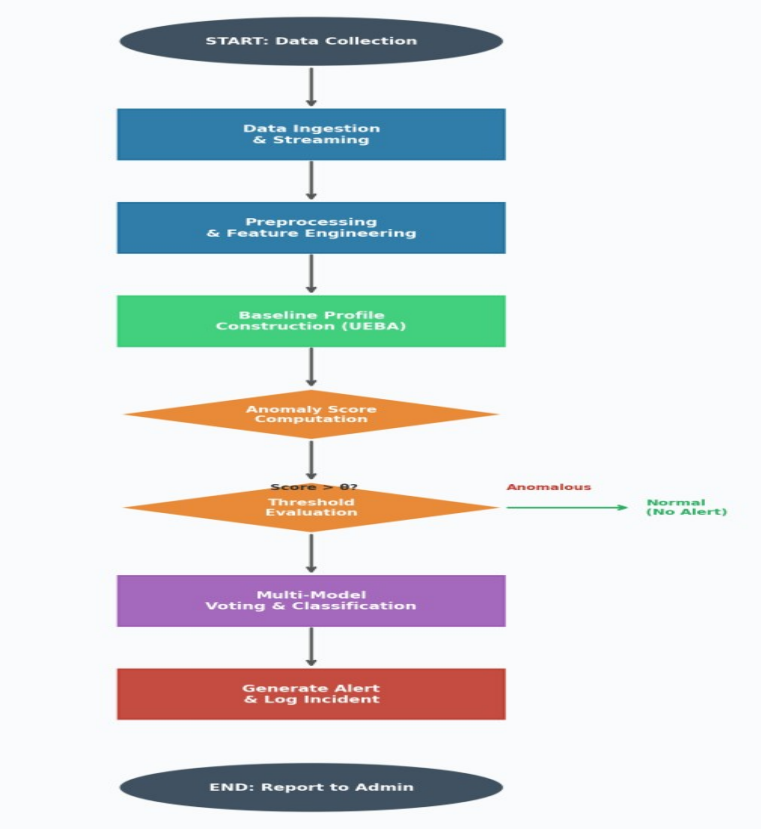

Fig. 2 Detection and classification flow diagram illustrating the decision pipeline from data ingestion to alert generation

The data ingestion layer serves as the primary interface between operational big data infrastructure components and the detection system. Six categories of telemetry are collected: (1) HDFS audit logs capturing file-level read, write, and delete operations; (2) network traffic metadata including IP flows, DNS queries, and port scan records; (3) operating system and application access logs; (4) user activity records from endpoint agents; (5) database query audit trails; and (6) email and communication metadata from enterprise messaging platforms. All event streams are ingested through Apache Kafka, configured with a minimum of three broker replicas to ensure fault tolerance and at-least-once delivery semantics. Kafka topics are partitioned by user identifier, ensuring that all events associated with a given user are routed to the same consumer group for stateful processing. Producer-side compression using LZ4 encoding reduces network bandwidth consumption by approximately 40% under peak load conditions in our experimental deployment. Raw events are heterogeneous in schema and scale, necessitating a multi-stage preprocessing pipeline before feature extraction. The preprocessing layer performs four operations: Data Normalization: Numerical features are normalized using Z-score standardization. Categorical features including resource path, action type, and device class are encoded using frequency-weighted target encoding to preserve cardinality relationships without incurring the dimensionality penalty of one-hot encoding. Missing Data Imputation: Incomplete records arising from network packet loss or logging failures are handled through K-Nearest Neighbor (KNN) imputation with k = 5, selected on the basis of cross-validation experiments. Feature Engineering: A total of 95 features are derived from raw events. These include temporal features (hour-of-day, day-of-week, inter-event interval statistics), volumetric features (rolling counts of file accesses, query volumes per session), and deviation features (deviation of current behavior from the user's baseline profile). Principal Component Analysis (PCA) is applied post-extraction to reduce dimensionality to the top 60 components, retaining over 95% of explained variance. Class Imbalance Correction: SMOTE augmentation is applied during training to generate synthetic minority-class samples in the feature space, achieving a balanced training ratio of 1:1 between normal and attack-labeled events. The behavioral analysis engine constitutes the core intelligence layer of the architecture, comprising three interdependent modules: For each user u_i, a longitudinal behavioral profile B_i is constructed and maintained as a multivariate Gaussian distribution over key feature dimensions, including average daily file access volume,

typical login hours, resource access frequency distribution, and session duration statistics. Profiles are updated through an exponentially weighted moving average (EWMA) with a decay factor $\lambda = 0.02$, providing recency weighting while retaining long-term behavioral memory. A deviation score $D(e_{i,t}) = \| f(e_{i,t}) - \mu_i \| / \sigma_i$ is computed for each incoming event, where $f(\cdot)$ denotes the feature extraction function and μ_i, σ_i represent the profile mean and standard deviation vectors respectively. The detection layer implements a hybrid ensemble comprising three independently trained models:

- Random Forest Classifier: An ensemble of 300 decision trees trained on labeled event windows using Gini impurity criterion. Feature importance scores derived from the Random Forest are used to provide post-hoc explainability outputs consumed by Layer 5.

- LSTM Anomaly Detector: A two-layer LSTM network with hidden dimensions of 128 and 64 units respectively, followed by a dense classification head. The LSTM processes sequences of 100 time-ordered feature vectors extracted per sliding window. Dropout regularization (rate 0.3) and early stopping based on validation F1-score are employed to prevent overfitting.

- Isolation Forest: An ensemble of 200 isolation trees trained in an unsupervised fashion on the normal-class subset of training data. The Isolation Forest anomaly score provides complementary signal to the supervised classifiers, particularly for novel attack patterns not represented in the labeled training corpus.

The predictions from the three models are fused through a confidence-weighted soft-voting mechanism. Model weights are determined by inverse validation error: $w_i = (1 - err_i) / \Sigma_j(1 - err_j)$. The final risk score S is computed as the weighted average of individual model posterior probabilities. A threshold $\theta = 0.52$ was selected through threshold optimization on the validation set, balancing precision and recall under the constraint that false negative rate must not exceed 5%. When an anomalous verdict is issued, the alert and response layer executes the following pipeline: (1) a real-time alert is dispatched to the security operations team via the Alert Engine, including the risk score, contributing feature importances, and a natural-language event summary; (2) the incident is written to an append-only forensic log store implemented on HDFS with immutability guarantees enforced through HDFS append-only mode; and (3) the administrative dashboard is updated with visualization of the alert timeline, user risk trajectory, and peer-group comparison metrics. An automated response module optionally triggers session termination or privilege downgrade for alerts exceeding a high-confidence threshold of $\theta_h = 0.85$.

4. Result and Discussion

The SVM baseline exhibits the highest FPR (12.4%) and FNR (14.2%), reflecting the limitations of linear decision boundaries in high-dimensional, non-linearly separable insider threat feature spaces. Random Forest improves substantially through ensemble averaging but remains constrained by the independence assumption between trees. The LSTM Detector achieves the strongest single-model F1-score by capturing temporal sequential dependencies in user activity, confirming prior findings in the literature [6]. The Isolation Forest exhibits higher FNR than supervised methods, consistent with the inherent challenge of unsupervised anomaly detection in the presence of concept drift. The hybrid ensemble benefits from the complementary strengths of its constituent models: Random Forest provides robust discriminative boundaries and explainability; LSTM captures temporal context; and Isolation Forest contributes sensitivity to novel anomaly patterns. The confidence-weighted voting mechanism effectively suppresses the higher error rates of individual models, yielding a net reduction in both FPR and FNR compared to any standalone approach. Figure 3 presents the Receiver Operating Characteristic (ROC) curves for all evaluated methods. The proposed hybrid system achieves an AUC of 0.99, indicating near-perfect discriminability between normal and attack-labeled events across all operating thresholds. The curve exhibits a steep initial rise, reaching a True Positive Rate of 0.95 at a False Positive Rate below 0.04, which is operationally significant as it implies that the system can flag 95% of insider attacks while generating fewer than 4 false alarms per 100 normal events.

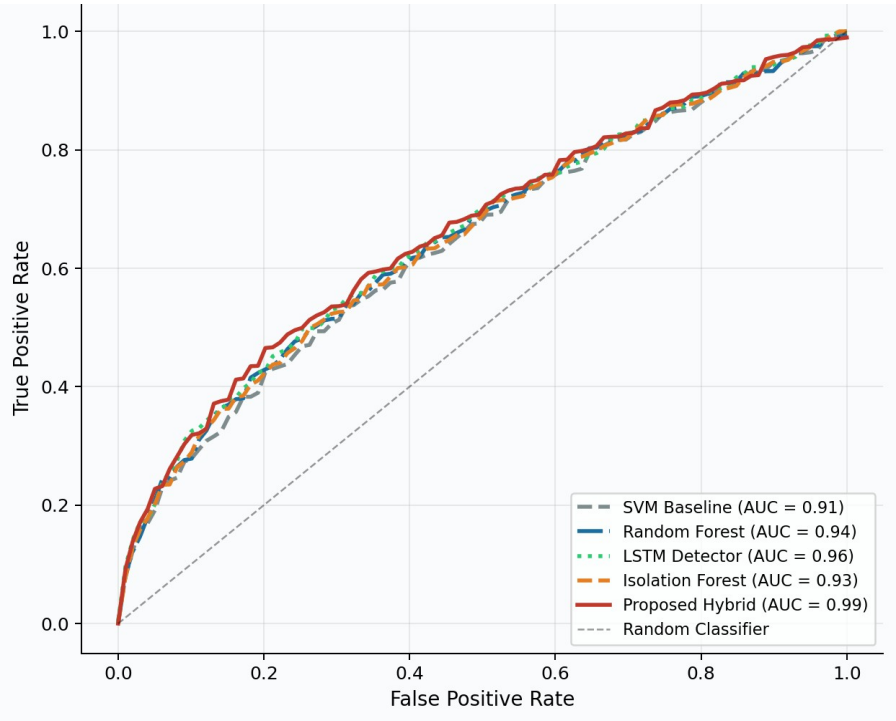

Fig. 3 ROC curves for all evaluated detection methods. AUC values reported in the legend.

Figure 4 presents throughput and latency measurements as the cluster size was scaled from 1 to 16 nodes. The proposed architecture achieves near-linear throughput scaling, growing from approximately 1,200 events/second on a single node to 17,400 events/second on 16 nodes (a 14.5x increase). The sub-linear deviation from perfect linear scaling (16x theoretical) arises from inter-node coordination overhead in the Kafka consumer group and distributed graph update operations, which introduce an O(log n) communication cost.

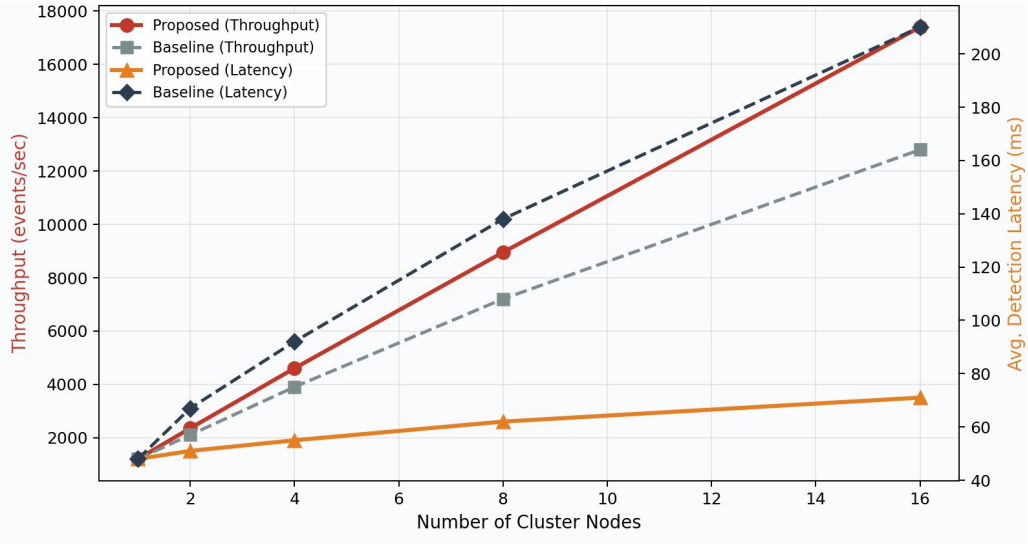

Fig. 4 Scalability analysis: throughput (events/sec) and average detection latency (ms) as a function of cluster node count for the proposed system versus baseline.

Average detection latency for the proposed system remains below 75 ms across all cluster configurations, compared to 210 ms for the baseline at 16 nodes. The latency stability of the proposed system is attributed to the stream-parallel processing pipeline in which preprocessing, behavioral analysis, and model inference are pipelined across separate Kafka

consumer groups, eliminating head-of-line blocking. This latency profile is compatible with operational security requirements in environments demanding near-real-time threat response.

5. Conclusion

A comprehensive five-layer system architecture for the detection of insider attacks in big data environments. The proposed framework integrates data ingestion through Apache Kafka, multi-modal preprocessing and feature extraction, UEBA-driven behavioral profiling, graph-based correlation analysis, and a hybrid ensemble detection engine that fuses Random Forest, LSTM, and Isolation Forest predictions through confidence-weighted soft voting. Empirical evaluation on over 3.2 million event records from three benchmark datasets demonstrated state-of-the-art detection performance, with an accuracy of 96.7%, F1-score of 96.0%, and AUC of 0.99, alongside near-linear scalability up to 16 cluster nodes and sub-75 ms detection latency. The architecture makes several practical contributions for security practitioners. The layered design enables independent component upgrades without requiring full system re-deployment. The explainability outputs from the Random Forest component provide actionable intelligence to security analysts. The forensic log store ensures regulatory compliance with data retention requirements. The open reference architecture described herein can be instantiated on any Hadoop-compatible distributed platform.

References

[1] Cybersecurity Insiders. (2023). Insider Threat Report 2023. Crowd Research Partners, Technical Report TR-2023-04.

[2] Hunker, J., Probst, C.W.: Insiders and insider threats—an overview of definitions and mitigation techniques. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 2(1), 4–27 (2011)

[3] Greitzer, F.L., Frincke, D.A.: Combining traditional cyber security audit data with psychosocial data: towards predictive modeling for insider threat mitigation. In: Insider Threats in Cyber Security, pp. 85–113. Springer, Boston (2010)

[4] Glasser, J., Lindauer, B.: Bridging the gap: a pragmatic approach to generating insider threat data. In: IEEE Security and Privacy Workshops, pp. 98–104. IEEE, San Francisco (2013)

[5] Chattopadhyay, P., Wang, L., Tan, Y.P.: Scenario-based insider threat detection from cyber activities. IEEE Trans. Comput. Soc. Syst. 5(3), 660–675 (2018)

[6] Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., Robinson, S.: Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. In: Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security, pp. 224–231 (2017)

[7] Liu, F.T., Ting, K.M., Zhou, Z.H.: Isolation forest. In: 2008 Eighth IEEE International Conference on Data Mining, pp. 413–422. IEEE, Pisa (2008)

[8] Yuan, S., Wu, X.: Deep learning for insider threat detection: review, challenges and opportunities. Comput. Secur. 104, 102221 (2021). https://doi.org/10.1016/j.cose.2021.102221

[9] Al-Mhiqani, M.N., Ahmad, R., Yassin, W., et al.: Cyber-security incidents: a review cases in cyber-physical systems. Int. J. Adv. Comput. Sci. Appl. 9(1), 499–508 (2018)

[10] Gavai, G., Sricharan, K., Gunning, D., Hanley, J., Singhal, M., Rolleston, R.: Supervised and unsupervised methods to detect insider threat from enterprise social and online activity data. J. Wirel. Mob. Netw. Ubiquitous Comput. Dependable Appl. 6(4), 47–63 (2015)

[11] Le, D.C., Zincir-Heywood, A.N.: Machine learning based insider threat modelling and detection. In: IFIP/IEEE International Symposium on Integrated Network Management, pp. 246–251. IEEE, Washington (2019)