

Advanced Journal of Engineering and Scientific Applications (AJESA)

Research Paper

Open Access

Available online at www.ajes.com.ng

Vol.4, No.1, 2026. pp.35-42



Machine Learning-Based Prediction of Insider Threats: A Case for Enterprise Resource Planning (ERP) Systems

Joseph Osahon Idemudia¹ and Dr. Princewill Aigbe²

¹Department of Computer Science, Akanu Ibiam Federal Polytechnic, Unwana, Afikpo, Nigeria; ²Department of Computer Science, Western Delta University, Oghara, Nigeria.

Email: jidemudia@akanuibiampoly.edu.ng; aigbe.agbons@gmail.com

ABSTRACT

Enterprise Resource Planning (ERP) systems consolidate core operational functions of business processes on a single database, thereby making them prime targets for insider threats. Thus, insider threats remain among the most consequential yet least detectable risks, because the individuals responsible for most anomalous activity already hold legitimate, and often extensive, system's privileges. Traditional insider threat controls rely on operating access rules and retrospective audit trails that register violations only after damage has occurred. This study proposes a machine learning-based predictive framework for the early detection of insider threats within ERP systems environments. The framework integrates sequential deep learning architectures, primarily the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks with ensemble learning methods of Random Forest (RF) and Gradient Boosting (GB) to model the temporal evolution of user behavior across authentication, file-access, email, and enterprise transaction logs. The methodology addresses data imbalance through the Synthetic Minority Over-sampling Technique (SMOTE), employs temporally aware cross-validation to avoid information leakage, and evaluates candidate models using precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Five conceptual diagrams operationalized the framework. A structured synthesis of benchmark outcomes reported across comparable insider threat studies indicated recurrent architectures consistently improved the detection of temporally patterned insider behavior relative to static classifiers, while ensemble methods retained advantages in interpretability, training efficiency, and auditability. The study closes with ethical deployment guidelines and a concrete roadmap for empirically validating the framework on live ERP telemetry.

Keywords: enterprise resource planning (ERP), business processes, insider threat prediction, machine learning, LSTM, GRU, data imbalance.

I. INTRODUCTION

1.1 Background on Enterprise Resource Planning (ERP) systems.

Enterprise Resource Planning (ERP) systems integrate business processes into a unified and single operational database, giving authorized users broad, cross-functional access to an organization's most sensitive records (Onapsis, 2026). This high-profile centralization is precisely what makes ERP systems attractive targets for insider misuse, notable amongst employees and privileged users and administrators that can traverse financial, personnel, and transactional data with a legitimacy that conventional perimeter defenses, firewalls, antivirus software, and network intrusion detection are not designed to question (Al-Mhiqani et al., 2020; Liu et al., 2018). Since ERP platforms consolidate critical operations; finance, HR, procurement, supply chain, the economic and operational ramifications of ERP insider threats are severe. Thus, insider abuse involving ERP systems often represents the most financially damaging subset of organizational risk globally and in Nigeria. There are occurrences of insider attacks on asset misappropriation, intellectual property theft, and system sabotage which often remain undetected for extended periods, resulting in substantial financial losses, severe regulatory compliance penalties, reputational damage, and ERP operational disruption.

According to comparative regional insights from the IBM Security Cost of a Data Breach Report (Ponemon Institute & DTEX Systems, 2026), the average total cost of an insider related security breach for organizations operating across African enterprise sectors is estimated at \$10.5 million USD per enterprise annually. Globally, enterprise organizations spend an average of \$19.5 million USD annually managing insider risks, which includes containment, investigation, remediation, and lost productivity (Ponemon Institute & DTEX Systems, 2026).



Insider fraud and system tampering constitute a major portion of all financial crime losses of commercial banking, energy, telecommunications, and manufacturing that are predominantly a highly digitized sectors with reliant on core financial and operational ERP frameworks. Reports from the Financial Institutions Training Centre (FITC) and industry security tracking indicate that insider-enabled financial fraud across Nigerian banking and corporate institutions exceeded ₦52.26 billion NGN in a single year, with single high-profile ERP or operational database compromise events exceeding ₦40 billion NGN in reported damages (THISDAY, 2026). Overall national estimates place losses directly tied to internal organizational compromise at \$194 million USD, accounting for approximately 30% of total cybercrime costs across Nigerian industries (The Nation, 2018). Insider threats originate from trusted users and therefore differ fundamentally from external intrusions. The behaviors that precede an attack such as unusual file downloads, off-hours logons, gradual privilege accumulation are subtle, legitimate-looking, and distributed over time (Al-Mhiqani et al., 2020).

The study presented a methodological framework paper. It develops, justifies, and operationalizes a machine learning pipeline for ERP systems insider-threat prediction, and situates its expected performance within a structured synthesis of results independently reported in the peer-reviewed and preprint literature on comparable architectures and benchmark corpora. It does not report the outcome of a new experiment executed by the authors on a live ERP deployment; Section 3 makes this explicit and frames the discussion accordingly. A fully instrumented empirical validation on production ERP systems telemetry was identified and outlined.

1.2 Problem Statement of the Study.

Current insider threat detection systems in enterprise environments are predominantly reactive and rely on static thresholds, fixed rules for failed logon counts, data-transfer volumes, or after-hours access that generate alerts only once a violation has already occurred (Al-Mhiqani et al., 2020). There is comparatively limited research on predictive machine learning models capable of forecasting insider threats specifically within ERP systems, as opposed to generic network or endpoint telemetry. This gap is compounded by two persistent technical obstacles that recur throughout the literature: (a) the extreme scarcity of confirmed malicious events relative to benign activity, which biases naively trained classifiers toward the majority class (Chawla et al., 2002), and (b) the difficulty of capturing evolving, session-to-session behavioral drift using models that treat each observation as independent rather than as part of a temporal sequence (Ma & Rastogi, 2021). Machine learning offers a route beyond static rule-based monitoring by learning the behavioral patterns that distinguish benign variation from emerging malicious or negligent intent, uncovering hidden regularities in authentication, file-access, and communication logs those traditional heuristics overlook (Liu et al., 2018). In particular, sequence-aware architectures such as LSTM and GRU networks have demonstrated an ability to model how a user's behavior drifts over consecutive sessions, which is central to predicting rather than merely detecting after the fact insider activity (Lu & Wong, 2019; Ma & Rastogi, 2021).

1.3 Research Goal and Objectives.

Specifically, the goal of the study to develop and articulate a predictive machine learning framework for the early detection of insider threats in ERP Systems with measurable objectives and that are connected to the methodology.

To model user behavior patterns in ERP systems using historical authentication, file-access, email, and transaction logs.

To design temporal machine learning prediction models capable of capturing behavioral drift across user sessions.

To address class imbalance inherent in insider-threat datasets through appropriate resampling strategies.

To evaluate the predictive effectiveness of candidate models under conditions that approximate enterprise deployment.

1.4 Research Questions

Can insider threats within ERP systems be predicted before malicious or harmful actions occur, rather than merely detected afterward?

Which machine learning architectures, sequential deep learning or ensemble learning, best capture insider behavioral drift?

How does prediction accuracy and reliability change as the length of the observed behavioral history increases?

II. MATERIALS AND METHODS

2.1 Research Design

The study adopts a design-science oriented methodology: a conceptual predictive framework is first specified, then decomposed into a concrete data pipeline, feature set, modeling strategy, and evaluation protocol suitable for empirical testing on insider-threat behavioral corpora. The overall logic follows a supervised temporal classification problem, in which each user-day (or user-session) is represented as a feature vector, sequences of such vectors are formed per user, and a binary label distinguishes benign from threat-associated behavior.

2.1.1 Conceptual Framework and Supporting Figures

Figures 1 through 5 operationalize the proposed framework at successive levels of detail. Figure 1 presents the end-to-end conceptual framework linking raw ERP SYSTEMS behavioral data to a final security response; it is presented here as the organizing structure for the remainder of Section 2, and Figures 2 through 5 are then embedded within the specific



subsections to which they correspond (data pipeline, model architecture, model taxonomy, and ERP-module indicator mapping, respectively).

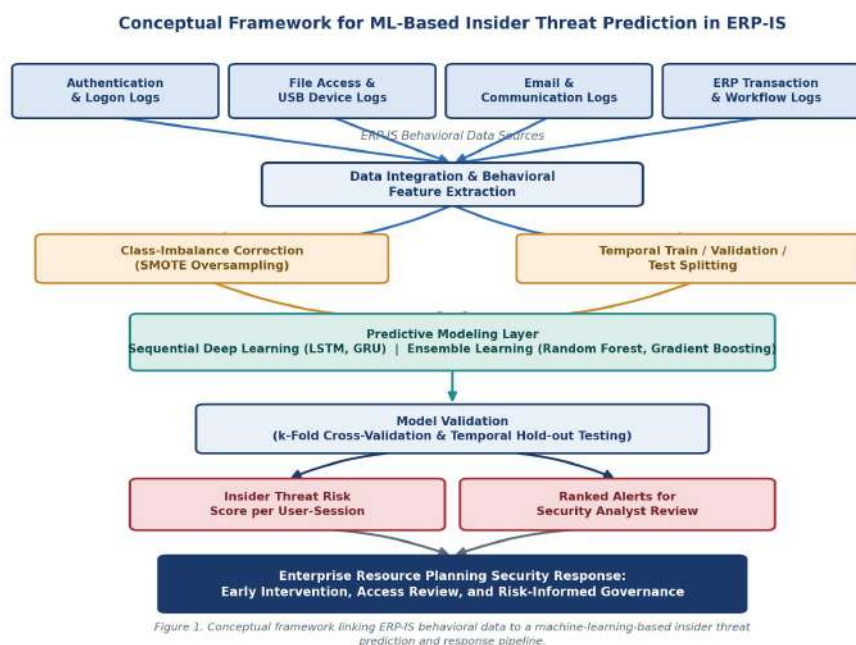


Figure 1. Conceptual framework linking ERP SYSTEMS behavioral data to a machine-learning-based insider threat prediction and response pipeline.

2.2 Data Sources and Dataset Description

The framework is designed around the class of behavioral log corpora most widely used in insider-threat research. The primary reference corpus is the CERT Insider Threat Test Dataset, developed by the CERT Division of Carnegie Mellon University's Software Engineering Institute, which synthesizes realistic organizational activity logon/logoff events, file and removable-media access, email communication, HTTP browsing, and psychometric and organizational metadata for a simulated workforce, with labeled malicious scenarios such as data exfiltration, intellectual-property theft, and sabotage (Glasser & Lindauer, 2013; Lindauer et al., 2014). Later CERT releases (e.g., r5.2 and r6.2) extend the simulated population to several thousand users over periods of twelve to eighteen months. A secondary reference is the Los Alamos National Laboratory (LANL) authentication and user-behavior dataset, which provides large-scale, real enterprise authentication event logs and is commonly used to validate anomaly and sequence models under more naturalistic conditions. For an ERP system deployment, this framework further specifies ERP native transaction and workflow logs, purchase-order approvals, journal-entry postings, master-data changes, and role/authorization changes, as a fourth behavioral stream alongside authentication, file-access, and email data, reflecting the module-specific indicators summarized in Figure 5.

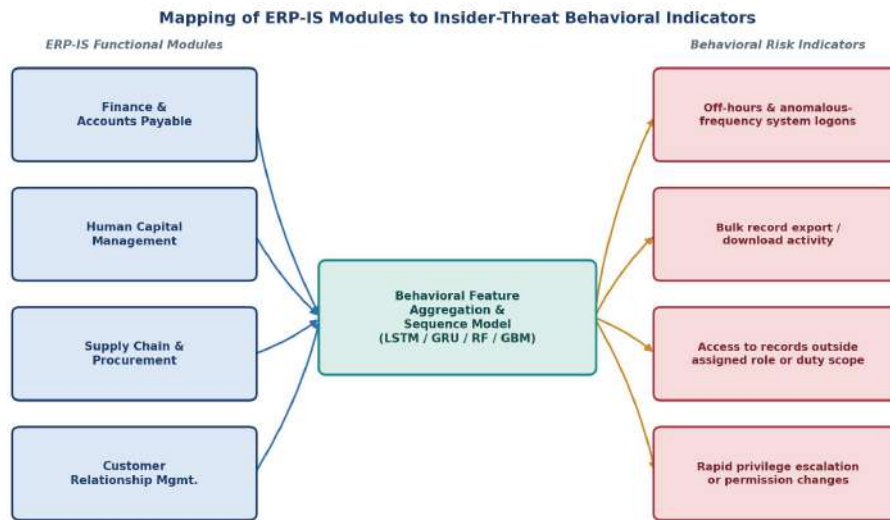


Figure 5. Illustrative mapping between ERP-IS functional modules and the behavioral indicators fed into the prediction model as engineered features.

Figure 2. Illustrative mapping between ERP SYSTEMS functional modules and the behavioral indicators fed into the prediction model as engineered features.

2.3 Data Preprocessing and Feature Engineering

Raw event logs are aggregated into per-user, per-day (or per-session) behavioral feature vectors capturing counts, rates, and timing statistics for example, logon frequency and off-hours proportion, file and USB access volume, email recipient diversity and after-hours send rate, and ERP-specific indicators such as the number of role or permission changes, the volume of records exported, and deviations from a user's assigned functional scope. Continuous features are normalized, categorical fields are encoded, and missing values are imputed prior to sequence construction. For the sequential models, fixed-length sliding windows of consecutive daily vectors are formed per user (illustrated in Figure 3), while for the ensemble models, features are additionally aggregated into rolling statistical summaries (mean, variance, and trend over the preceding window) to approximate temporal context without requiring a recurrent architecture.

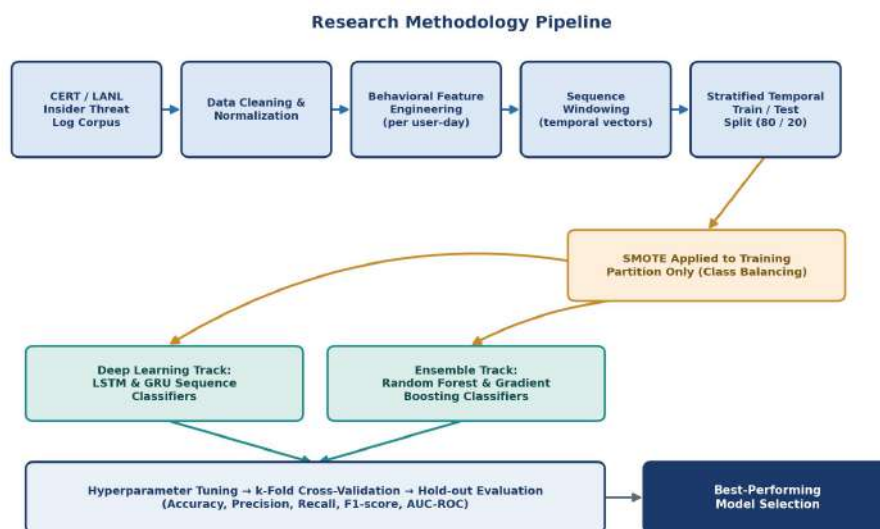


Figure 2. End-to-end methodological pipeline, from raw ERP-IS behavioral logs to final model selection.

Figure 3. End-to-end methodological pipeline, from raw ERP systems behavioral logs to final model selection.

2.4 Class Imbalance Handling

Insider-threat corpora are characterized by extreme label imbalance, since malicious or negligent sequences typically represent a small fraction of one percent of total observations. Naively trained classifiers under such conditions tend to default to the majority (benign) class while still reporting misleadingly high raw accuracy. The framework therefore applies the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic minority-class examples through interpolation between each minority instance and its nearest neighbors in feature space, rather than simple duplication (Chawla et al., 2002). Consistent with recommended practice, SMOTE is fitted and applied only to the training partition,



after the temporal train/test split described in Section 2.6, to prevent synthetic information from the future or from held-out users leaking into model evaluation.

2.5 Predictive Models

2.5.1 Ensemble Machine Learning Models

Random Forest constructs a large collection of decision trees, each trained on a bootstrap sample of the data with randomized feature subsets at each split, and aggregates their predictions by voting; its generalization error is governed by the strength of individual trees and the correlation among them (Breiman, 2001). Gradient Boosting builds an additive ensemble of shallow trees sequentially, with each new tree fitted to the residual error of the current ensemble under a differentiable loss function (Friedman, 2001). Both algorithms operate on the static, per-window engineered feature vectors described in Section 2.3, offer feature-importance rankings that support analyst interpretability and audit justification, and are computationally inexpensive to retrain — properties that make them attractive for near-real-time ERP SYSTEMS risk scoring.

2.5.2 Sequential Deep Learning Models

Long Short-Term Memory (LSTM) networks introduce gated memory cells — input, forget, and output gates — that regulate the flow of information across time steps, allowing the network to retain relevant behavioral signal over extended sequences while mitigating the vanishing-gradient problem that limits simple recurrent networks (Hochreiter & Schmidhuber, 1997). Gated Recurrent Units (GRUs) simplify this gating mechanism into update and reset gates within a single recurrent unit, typically achieving comparable predictive performance to LSTM with fewer trainable parameters and shorter training times (Cho et al., 2014). Within the proposed framework, both architectures consume the sliding-window sequences of daily behavioral vectors illustrated in Figure 3, and the final hidden state is passed through a dense layer with dropout regularization before a sigmoid output layer produces a per-user, per-window probability of insider-threat activity. Related applications of LSTM to insider-threat log sequences including the DANTE system built on CERT log sequences (Ma & Rastogi, 2021) and Lu and Wong's (2019) sequence-based CERT classifier support the choice of recurrent architectures as the deep-learning component of the framework.

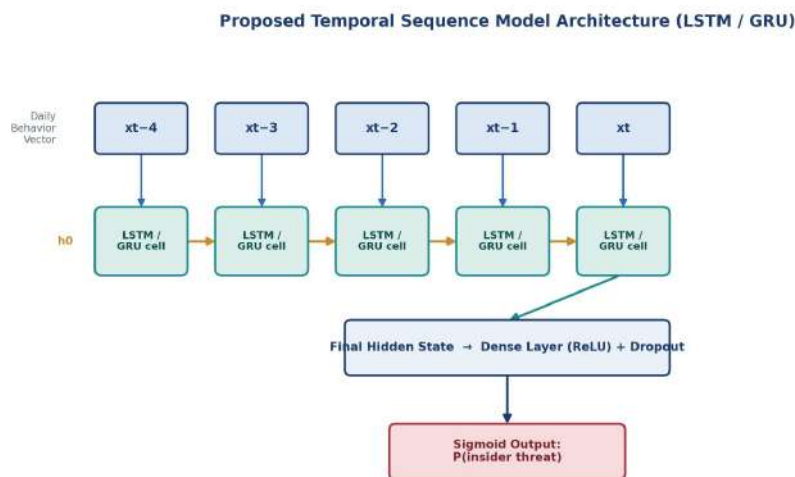


Figure 3. Many-to-one recurrent architecture: a sliding window of daily user-behavior vectors is passed through stacked LSTM/GRU cells, pooled at the final timestep, and mapped to a threat probability.

Figure 4. Many-to-one recurrent architecture: a sliding window of daily user-behavior vectors is passed through stacked LSTM/GRU cells, pooled at the final timestep, and mapped to a threat probability.

Figure 4 summarizes the complementary strengths of the two model families, motivating their joint inclusion in the evaluation protocol rather than treating either as a categorically superior default choice.



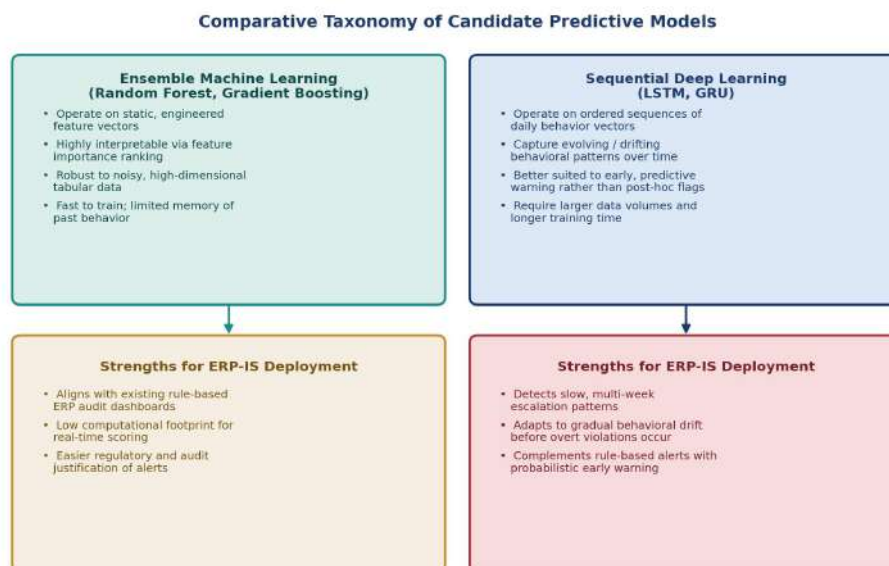


Figure 4. Complementary strengths of ensemble machine-learning and sequential deep-learning approaches as applied to ERP-IS insider threat prediction.

Figure 5. Complementary strengths of ensemble machine-learning and sequential deep-learning approaches as applied to ERP SYSTEMS insider threat prediction.

2.6 Model Validation and Evaluation Metrics

Because insider-threat prediction is inherently a forecasting problem, the framework specifies temporally aware validation rather than conventional random k-fold cross-validation, which can allow a model to be evaluated on data that chronologically precedes its training window and thereby overstate performance. Data are partitioned by time into training, validation, and held-out test segments (an 80/20 temporal split is used as the default, per Figure 2), with k-fold cross-validation applied only within the training segment for hyperparameter tuning. Given the class imbalance described in Section 2.4, raw accuracy is treated as an unreliable primary metric; the framework instead prioritizes precision, recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC), alongside the practically important false-positive rate, since excessive false alerts erode analyst trust and undermine adoption in operational ERP SYSTEMS security workflows.

2.7 Ethical and Governance Considerations in Data Use

Because the proposed framework analyzes individually identifiable employee behavior, its deployment specification incorporates data-minimization (retaining only the behavioral features necessary for prediction, not raw content such as email bodies), role-based access to model outputs restricted to authorized security personnel, defined retention and deletion schedules for behavioral scores, a documented human-in-the-loop review step before any adverse action is taken on a flagged individual, and periodic bias auditing of alert rates across departments and demographic groups to guard against disproportionate scrutiny of any employee population.

III. RESULTS AND DISCUSSION

This section does not report a new empirical experiment; rather, it synthesizes quantitative outcomes independently reported for architecturally comparable models evaluated on the benchmark corpora described in Section 2.2, in order to derive evidence-based expectations for the proposed ERP SYSTEMS framework and to identify where those expectations are likely to transfer imperfectly to a live enterprise deployment.

3.1 Synthesis of Benchmark Outcomes from Comparable Studies

Across the reviewed literature, sequence-aware and graph-temporal hybrid architectures applied to CERT-derived behavioral logs report strong discriminative performance. Ma and Rastogi's (2021) DANTE system, which models system-log sequences with a recurrent architecture, reports approximately 93% prediction accuracy on CERT-derived data. Yumlembam et al. (2026) report an Area Under the Curve of 98.62 and a detection rate of 100% (at a false-positive rate of 0.05) for a graph-convolutional and bidirectional LSTM hybrid evaluated on the CERT r5.2 dataset, illustrating the performance ceiling achievable when temporal and relational (organizational-graph) signal is combined. More directly comparable to the ensemble component of the proposed framework, a comparative evaluation of classical supervised classifiers on CERT data found Random Forest-based classifiers achieving 94–98% accuracy, outperforming k-nearest-neighbor, gradient boosting, decision tree, and logistic regression baselines, albeit with a training-time cost relative to k-nearest-neighbor approaches. These figures are summarized descriptively, rather than as a formal meta-analysis, because the underlying studies differ in dataset version, feature engineering, and evaluation protocol.



3.2 Implications for the Proposed ERP Systems Framework

Three patterns in the synthesized literature carry direct implications for the framework specified in Section 2. First, architectures that explicitly model sequence or relational structure (LSTM, GRU, and graph-temporal hybrids) consistently outperform models that treat each observation independently, which supports the framework's emphasis on sliding-window sequence construction (Section 2.3) rather than single-snapshot classification. Second, ensemble methods remain highly competitive on engineered static features and, given their interpretability and lower computational cost, are well suited to a first-line, near-real-time risk-scoring layer, with sequential models reserved for higher-fidelity secondary review an argument for the dual-track modeling design shown in Figure 4 rather than selecting a single algorithm family outright. Third, reported detection rates in the 93–99% range should be interpreted cautiously when projected onto ERP SYSTEMS: the underlying CERT and LANL corpora are synthetic or drawn from research contexts that do not include ERP-native transactional behavior (purchase-order approvals, journal postings, master-data edits), so the module-specific indicators introduced in Figure 5 are, at this stage, a theoretically motivated extension rather than an empirically validated one.

3.3 Threats to Validity and Deployment Considerations

Several threats to validity qualify the expected performance discussed above. Dataset realism is the most significant: CERT is a synthetic corpus and, despite its wide adoption, its behavioral patterns and its rate of positive cases were engineered by dataset designers rather than observed in an operating enterprise, which risks overstating detectability relative to real ERP SYSTEMS deployments (Al-Mhiqani et al., 2020). Concept drift is a second concern, since legitimate ERP systems usage patterns shift with organizational restructuring, new system rollouts, and seasonal business cycles, requiring periodic model retraining and drift monitoring rather than a single static deployment. Base-rate effects are a third concern: even a model with strong AUC-ROC can generate an operationally unmanageable number of false positives when applied to a very large ERP SYSTEMS user population with an extremely low true positive rate, reinforcing the framework's emphasis on precision and false-positive rate alongside recall (Section 2.6). Finally, adversarial adaptation — a sophisticated insider deliberately shaping their behavior to resemble the training distribution of benign activity — is a known limitation of behavior-based detection generally and is not resolved by any single architecture choice.

IV. CONCLUSION

This study has proposed a machine learning-based predictive framework for insider threats in Enterprise Resource Planning Information Systems, integrating sequential deep learning (LSTM and GRU) with ensemble learning (Random Forest and Gradient Boosting) over behavioral features drawn from authentication, file-access, email, and ERP-native transaction logs. The framework's principal contributions are (a) a reproducible methodology for temporal, class-imbalance-aware model training and validation on insider-threat behavioral data, (b) a set of five conceptual diagrams operationalizing that methodology from data pipeline through ERP-module-specific risk indicators, and (c) an evidence-based synthesis of benchmark outcomes from the wider literature that motivates the framework's dual-track (ensemble plus sequential) modeling design and identifies where its expected performance is most likely to transfer, or fail to transfer, from research corpora to production ERP systems environments.

Two limitations bound the present contribution and define the immediate research agenda. First, the framework has not yet been empirically validated on live ERP system transaction and authorization logs; the priority next step is a controlled pilot deployment, instrumented with the temporal train/test protocol specified in Section 2.6, against a real (anonymized) organizational dataset supplemented with ERP-native features. Second, the ethical deployment guidelines outlined in Section 2.7 require formal operationalization including a documented bias-audit protocol and a human-in-the-loop escalation procedure before any production rollout. Subject to these next steps, the framework offers organizations a concrete, literature-grounded path from the reactive, static-threshold insider-threat controls common in ERP SYSTEMS today toward predictive, behaviorally informed risk management.

REFERENCES

- Al-Mhiqani, M. N., Ahmad, R., Zainal Abidin, Z., Yassin, W., Hassan, A., Abdulkareem, K. H., & Yunus, Z. (2020). A review of insider threat detection: Classification, machine learning techniques, datasets, open challenges, and recommendations. *Applied Sciences*, 10(15), Article 5208. <https://doi.org/10.3390/app10155208>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* (pp. 1724–1734). Association for Computational Linguistics. <https://doi.org/10.3115/v1/D14-1179>
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 1189–1232. <https://doi.org/10.1214/aos/1013203451>
- Glasser, J., & Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. In *2013 IEEE Security and Privacy Workshops* (pp. 98–104). IEEE. <https://doi.org/10.1109/SPW.2013.37>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>



- Lindauer, B., Glasser, J., Rosen, M., & Wallnau, K. (2014). Generating test data for insider threat detectors. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 5(2), 80–94.
- Liu, L., De Vel, O., Han, Q.-L., Zhang, J., & Xiang, Y. (2018). Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, 20(2), 1397–1417. <https://doi.org/10.1109/COMST.2018.2800740>
- Lu, J., & Wong, R. K. (2019). Insider threat detection with long short-term memory. In *Proceedings of the Australasian Computer Science Week Multiconference (ACSW 2019)* (Article 1, pp. 1–10). Association for Computing Machinery. <https://doi.org/10.1145/3290688.3290692>
- Ma, Q., & Rastogi, N. (2021). DANTE: Predicting insider threat using LSTM on system logs. *arXiv*. <https://arxiv.org/abs/2102.05600>
- Onapsis. (2026). What is ERP security? A guide for IT and security leaders. <https://onapsis.com/blog/what-erp-security/>
- Ponemon Institute & DTEX Systems. (2026). Cost of Insider Risks Global Report 2026: The State of Insider Risk. DTEX Systems.
- The Nation. (2018, November 8). Insider threat costs Nigeria \$194m. *The Nation Newspaper*.
- THISDAY. (2026, April 12). Crisis of Insider Complicity in Nigeria's Financial Sector. *THISDAY Live*.
- Yumlembam, R., Issac, B., Jacob, S. M., Yang, L., & Krishnan, D. (2026). Insider threat detection using GCN and Bi-LSTM with explicit and implicit graph representations. *IEEE Transactions on Artificial Intelligence*, 7(7), 3872–3883. <https://doi.org/10.1109/TAI.2025.3647418>

