This document contains the classification of machine learning paradigms and architectural designs utilized to model normal versus malicious insider behavior. To ensure a precise taxonomy of Deep Learning, Ensemble, and Classical architectures, the authors employed **NotebookLM** (processing 2–5 papers per prompt) to parse the 82 primary studies for specific algorithmic implementations. The data was cross-checked manually to ensure the accuracy of the AI-extracted information against the original study.

Prompt 2: Detection Models & Paradigms (RQ2)

Task: You are a research assistant specializing in Machine Learning and Cybersecurity. Analyze the uploaded papers and fill the **Detection Models and Paradigms Table** based on the following criteria.

Instructions for Columns:

- **Model Category:** Classify the type of model used. Look for: *Classical ML (e.g., SVM, RF), Deep Learning (e.g., LSTM, CNN), Hybrid (ML + DL), or Ensemble (e.g., XGBoost, Bagging).*
- **Specific Architecture:** List the exact names of the models or neural networks used (e.g., Bi-LSTM, Graph Neural Network (GNN), Autoencoder, Transformer).
- **Learning Paradigm:** Identify how the model was trained. Look for: *Supervised (uses labels), Unsupervised (anomaly detection/no labels), Semi-supervised, or Self-supervised.*
- **Modeling Approach / Novelty:** In 1-2 short sentences, explain what makes this model unique or how it represents user behavior (e.g., "Models daily sessions as image grids" or "Uses attention mechanisms to focus on rare file actions").
- **Limitations Mentioned:** Identify the specific weaknesses of the model mentioned by the authors (e.g., high computational cost, vanishing gradients, inability to detect multi-day attacks).

Output Format: Provide the results in a single-row table with these columns: | Authors (year)| Model Category | Specific Architecture | Learning Paradigm | Modeling Approach / Novelty | Limitations Mentioned |

Constraint: Use ONLY information from this paper. If a model detail is missing, write "Not Specified."

Paper ID	**Model Category**	**Specific Architecture**	**Learning Paradigm**	**Modeling Approach / Novelty**	**Limitations Mentioned**
Adun (2023)	Hybrid Supervised ML	SVM, ANFIS	Supervised & Hybrid	Combines SVM classification with ANFIS prediction to leverage hybrid model strengths.	Potential overfitting shown by irregular training curves; small dataset (350 samples).
Alabdulkareem (2022)	Ensemble Deep Learning	LSTM, GRU, SAE, ALO	Deep Learning (Mixed)	MWF-IDLDC model fusing LSTM, GRU, and SAE weights; uses ALO for hyperparameter tuning.	No explicit model limitations mentioned. No mention of source of data
Ahmadi (2025)	Shallow ML	RF, Gradient Boosting, K-means	Supervised & Unsupervised	ZTA framework using RF, GB, and K-means for real-time risk scoring and dynamic access control.	Privacy risks; potential for false positives; high computational demands affecting scalability.
Ahmed (2025)	lassical ML, Anomaly Detection	Random Cut Forest (RCF)	Unsupervised	Integrates the RCF algorithm within a SIEM (Wazuh) to detect internal user behavior anomalies for insider threats, aligning with industry frameworks.	Some complex patterns require further refinement; Detection Time could be improved for certain scenarios.
Al Hammadi et al. (2021)	Deep Learning, Ensemble, Classical ML	2D CNN, 1D CNN, Multi-Layer Perceptron (MLP) with CNN, Adaptive Boosting (AdaBoost), Random Forest, KNN	Supervised	Uses low-cost EEG devices and a combination of deep learning and ensemble models with explainable AI to detect industrial insider threats based on emotional instability.	Small sample size; Demographic scope.
Ali et al. (2025)	Deep Learning, Ensemble	BERT, BERTopic, Ensemble Model (BERT + BERTopic outputs via multi-class logistic regression)	Supervised	Integrates advanced NLP techniques (BERT, BERTopic) and ensemble learning to classify insider threat levels from narrative text data, emphasizing a human-machine partnership.	Data quality; Model tuning; Ethical concerns (bias, privacy); Baseline BERT tended to underestimate threat levels (especially medium and low); Low threat occurrence rates; Large case backlogs.
ALmihqani 2021	Hybrid (Deep Learning + Sampling Technique)	Deep Neural Network (DNN) with Adaptive Synthetic Sampling (ADASYN)	Supervised	The model integrates ADASYN to oversample low-frequency insider threat samples, addressing data imbalance, and then uses a Deep Neural Network for classification to improve	none

				detection performance for minority classes	
Almusawi 2024	Hybrid (Machine Learning + Expert Policies) / Ensemble	XGBoost, AdaBoost, SVM, Naive Bayes (NB), Expert Policies Algorithm	Supervised (for ML algorithms), Rule-based (for expert policies)	The model combines expert rules with multiple machine learning classification algorithms (XGBoost, AdaBoost, SVM, NB) and then intersects their results to enhance accuracy and reduce false positives .	The current expert policies might be limited in scope or coverage, the current feature extraction might not be comprehensive enough or capable of tracking complex behavioral changes, and the current set of ML algorithms or their parallelization could be improved to reduce errors
AL-Mihqani 2022	Hybrid (Multilayer Machine Learning Framework)	Random Forest (RF) for Misuse Insider Threat Detection (MITD), K-Nearest Neighbors (KNN) for Anomaly Insider Threat Detection (AITD)	Supervised (for MITD), Anomaly Detection (for AITD)	A multilayer framework where the first layer uses an integrated Entropy-VIKOR method to select the best ML classification models, and the second layer combines a Misuse Insider Threat Detection (RF) and an Anomaly Insider Threat Detection (KNN) model to form a hybrid system.	The current selection process for ML models and evaluation criteria might not be exhaustive, and the proposed framework does not fully address the imbalanced dataset problem, suggesting future work could incorporate techniques like SMOTE or ADASYN
Alshehari 2023 (IF)	Classical ML, Anomaly Detection	Isolation Forest	Unsupervised	Emphasizes anomaly detection using the Isolation Forest algorithm to identify unusual behaviors without labeled data, addressing data imbalance.	High false positives (for some ML approaches), challenges with temporal data.
Al-shehari & Alsawail 2023	Classical ML	XGB, RF, DT, KNN	Supervised	Leverages various random sampling techniques (under-sampling, over-sampling, hybrid sampling) with well-known machine learning algorithms to address extremely imbalanced datasets for insider threat detection	Random under-sampling can lead to decreased classification accuracy due to discarded beneficial observations; random over-sampling may cause overfitting and increased computational cost
Al-shehari (CNN) (2024)	Hybrids Deep Learning + ADYSN	Convolutional Neural Networks (CNN)	Supervised	Integrates CNN with data imbalance addressing techniques to enhance insider threat detection accuracy and robustness in imbalanced datasets. CNN analyzes data with a grid-like	SMOTE is sensitive to noise and outliers in the minority class, and randomness can lead to result variations . ADASYN can generate poor-quality synthetic samples if the minority class contains noise/outliers,

				structure to detect local patterns or anomalies	and k-nearest neighbor search can be computationally expensive . The model needs to be robust to threat pattern changes over time and its true test comes from real-world deployment and generalization to unseen data
AL-SHEHARI (LOF) (2024)	Classical ML	Density-Based Local Outlier Factor (DBLOF) algorithm	Unsupervised (Anomaly Detection)	Employs the DBLOF algorithm, fine-tuned to specifically tackle challenges posed by imbalanced datasets, by focusing on the local density deviation of data points to identify outliers indicative of potential insider threats [12]. It addresses imbalance at the algorithm level	High sensitivity to the 'contamination rate' hyper-parameter. Concerns about generalizability and risk of overfitting. Issues related to interpretability and scalability . Lacks a comprehensive cost-benefit analysis . The model's ability to generalize to other data is not yet verified, and its scalability in high-dimensional or real-time scenarios is untested
alshehri Abdullah (2022)	**Deep Learning**	**RNN** with **LSTM** units and **Direct Graph** structure	**Supervised** (Anomaly Detection)	Represents user actions as **multivariate time series** and uses **direct graphs** to learn **latent inter-relations** between different activity sequences.	Increased **time complexity** with larger batch sizes and lack of **Spatio-temporal dependencies** for remote access scenarios.
Amiri-Zarandi (2023)	**Deep Learning** Federated Learning	**Deep Autoencoder** integrated with **SHAP**	**Unsupervised** (Anomaly Detection)	Utilizes a **Federated Learning** framework for privacy and leverages **SHAP** to identify and explain the specific features contributing to an anomaly score.	Limited availability of **benchmark data** generated in real **IoT environments**; primarily relies on synthetic behavior data
Al-Shehari Tahir (2021)	**Classical ML** and **Ensemble**	**LR, DT, RF, NB, KNN,** and **Kernel SVM**	**Supervised**	Employs **One-Hot Encoding** and **SMOTE** to address encoding bias and extreme class imbalance in **pre-departure data leakage** scenarios.	Relies on a **synthetic dataset** (CERT), which may not fully reflect the **scalability** or complexity of real-world organizational data
Amuda (2022)	Hybrid	**CNN-GRU** (Convolutional Neural Network and Gated Recurrent Unit)	Supervised	Learns from **sequences of user actions** in time and frequency domains, incorporating **user roles** to achieve role-based behavior threat detection.	Not Specified.

Anju (2024)	Hybrid (DL + Optimization)	**Stacked CNN** (VGG19 + Xception) and **Attentional BiGRU**	Supervised	Extracts temporal representations from multivariate time-series logs using an **attention mechanism** to capture long-range user behavior without increasing network parameters.	Requires future exploration of **automatic feature extraction** and dimensionality reduction for newer datasets.
Anakath (2022)	Deep Learning	**Deep Belief Neural Network (DBN)** designed using **Restricted Boltzmann Machines (RBM)**	Supervised	Identifies insiders by monitoring unique **user interaction behavior patterns**, specifically biometric-style **mouse movements/clicks** and **keystroke dynamics**.	Training the model is difficult and potentially **insufficient** because cloud service providers do not provide adequate **real-time datasets**.
Asha S (2023)	Hybrid (Sampling + ML)	Double-layer architecture using **One-Class SVM (OCSVM)**, Robust Covariance, and Isolation Forest,	Anomaly Detection (Unsupervised/Semi-supervised),	Employs a **double-layer system** that first uses **Nearmiss-2 (NM-2)** undersampling to solve class imbalance before applying anomaly detection,,.	Perceiving **new threats** is difficult in classification techniques; high false detection rates remain a challenge in imbalanced data,.
Cai X (2024)	Deep Learning (Graph-based)	**LSTM**, Multi-head Attentive Pooling, and **GNN (GCN/GAT)**,	Hybrid (Supervised and Self-supervised),	Conducts the **first activity-level real-time detection** by adaptively learning a user activity graph to capture dependencies across sequences,,.	Increasing the number of neighbors in the graph can **introduce noise**; performance is sensitive to the retrieval threshold.
Dong J (2025)	Deep Learning (Generative)	**DDPM** (Denoising Diffusion Probabilistic Models) with **TCN** and **Curriculum Learning (CL)**,	Unsupervised (Anomaly detection via reconstruction),	Synergizes **diffusion models** with a curriculum-based strategy to **gradually expose the model** to increasingly complex behavioral patterns,.	Higher **computational training costs** and memory requirements compared to some recurrent baselines,.
Eshmawi (2026)	Classical ML and Deep Learning	SVM, Random Forest (RF), K-Nearest Neighbor (KNN), Deep Neural Network (DNN), and Naive Bayes (NB)	Supervised (Multi-class classification)	Models user behavior at four granularity levels (day, week, session, and sub-session) to capture both high-level patterns and fine-grained anomalies.	NB and KNN are sensitive to unbalanced data; DNN requires high computation power and cannot process past temporal perspectives.
Feng (2025)	Classical ML	Multi-Granularity User Anomalous Behavior Detection (MG-UABD) using Random Forest	Supervised (Classification)	Combines a coarse-grained module to screen all behaviors with a fine-grained module that builds individual models to learn the specific behavior patterns of each flagged user.	Increased computational time and space resources due to building separate models for each user; requires labeled data for initial training.

Ferraro (2025)	Generative Agent-Based Modeling (GABM)	Hierarchical multi-agent framework using LLaMA-3.1 8B	Prompt-based Reasoning/Inference (utilizes pre-trained LLMs without modification)	Employs Specialized Agents to analyze domain-specific logs using Chain-of-Thought reasoning, which are then synthesized by a Supervisor Agent for final classification.	Prioritizes high recall over precision, leading to a higher number of false positives; reasoning speed is limited by computational resource availability
Gayathri (2025) [Adversarial]	Deep Learning	SNN-MLP, SNN-1DCNN, and TabNet	Supervised (Multi-class classification)	Uses self-normalizing neural networks (SNN) with SeLU activation for tabular stability; investigates how XAI (SHAP) helps insiders identify critical features to create stealthy backdoor triggers.	SNN-MLP exhibits significant performance loss (up to 28.9%) under specific poisoning; the trigger value selection method requires modification.
Gayathri (2025) [Cloud]	Hybrid (ML + DL)	Jordan Neural Network (JNN) and LS-AE (LSTM-Autoencoder)	Supervised	Fuses features from a manual domain extractor (standalone data) and an automated LS-AE (sequential data) into a JNN where context units receive feed from the output layer.	Low training speed compared to other networks; potential for overfitting.
Gayathri (2024) [Hybrid]	Hybrid (ML + DL)	SPCAGAN and Hybrid Bayesian Neural Network (BNN) with MLP and 1DCNN layers	Supervised (Multi-class classification)	Introduces SPCAGAN using linear manifold learning (SPCA) to maximize similarity between real and fake data; uses a hybrid BNN to express predictive uncertainty through probabilistic layers.	Bayesian Neural Networks are currently computationally expensive due to sampling or variational inference phases; requires time-cost optimization.
Gonzales (2025)	**Hybrid** (Pattern Mining + ABC Model)	**C3P** (Contiguous, Contextual, and Classifying Pipeline) and **C3P-SMART**	**Unsupervised**	Combines contiguous pattern mining with the **ABC (Antecedent-Behavior-Consequence)** framework and **n-gram probability distributions** (bigrams/trigrams) to contextually score and classify user behavior sequences.	Computational cost grows **quadratically** with the average sequence length (*O*(*N*·*L*2)), making it challenging for very large sequences.
Gupta (2024)	**Hybrid** (DL/ML + Federated Learning)	**FedMUP** (utilizing **AFed, CFed, and DFed** architectures)	**Supervised**	Employs a **Federated Learning (FedL)** environment to train local models on sensitive user data, sharing only **learned weights**	Current models may have **limits on data sharing**; future work is needed to develop more **adaptive learning**

				(parameters) rather than raw data to proactively predict malicious users via a User Behavior Evaluation (UBE) unit.	supervised privacy-preserving approaches.
Hafizu Rhman (2022)	**Deep Learning** (Recurrent Dynamic ANN)	**NARX** (Nonlinear Autoregressive Exogenous) neural network	**Supervised**	Utilizes a **time-series** approach to model unintentional insider threats, conceptualizing behavior through dimensions such as **Access, Motivation, Action, and Intent** to assess an organization's readiness for human error or policy violations.	**Oversight** in analyzing various levels of access based on specific jobs/behaviors; predictive ability exhibits **variations due to autocorrelation**
Haq (2022)	**Hybrid (DL + ML) / Ensemble**	Word2vec-LSTM, GLoVe-LSTM, XGBoost, AdaBoost, Random Forest (RF), KNN, and Logistic Regression (LR).	**Supervised**.	Utilizes transfer learning with pre-trained NLP word embeddings (Word2vec and GloVe) to extract semantic and syntactic context from email text and financial data to identify "persons of interest".	Huge data volume requires high computation; lacks real-time detection and a standard evaluation framework.
He (2022)	**Hybrid (ML + DL)**	**Layer 1:** LSTM + XGBoost (L-XGB). **Layer 2:** Bi-LSTM + Attention (BiLA-ITD).	**Supervised**.	Employs a double-layer framework to link external phishing email detection with internal group security via Bi-LSTM and Attention to capture long-range behavior dependencies.	Standard LSTM architectures suffer from sequence loss/forgetting; slow classification speed in character/word-level RCNN models.
He (2024)	**Hybrid (ML + DL)**	**Layer 1:** LSTM/Bi-LSTM + XGBoost (L-XGB). **Layer 2:** Bi-LSTM + Attention (BiLA-ITD).	**Supervised**.	Integrates the Multi-Domain Feature Time Series (MDFTS) algorithm to process heterogeneous logs and addresses blockchain security by mitigating credential theft initiated via social engineering.	Cannot detect image links, QR codes, or dynamic images in phishing emails, which may bypass the first defense layer.
Huang (2025)	**Deep Learning**	DenseAttDNN (Dense Connection + Attention Mechanism)	**Supervised**	Integrates IDS with UEBA to monitor the transition from external access to internal masquerader attacks; uses dense connections for feature reuse and	Lacks an automated mechanism for the dynamic recalibration of model weights across different cyber threat categories.

				attention to focus on rare attack classes,,.	
Jaiswal (2024)	**Classical ML & Deep Learning**	ANN, RVFL, ELM, RF, DT, XGB, kNN, and SVM	**Supervised**	Models user activity as day-long sequences to capture temporal context and activity frequency while handling extreme class imbalance through diverse sampling algorithms,.	Requires further optimization in cost-sensitive learning and reductions in computation time to improve scalability for large datasets.
Janjua (2021)	**Semi-supervised**	K-Means (Clustering) + Decision Tree (DT)	**Semi-supervised**	Uses unsupervised K-Means to label previously unlabeled email content based on TF-IDF clusters, identifying malicious content (e.g., "hou" and "ect" terms) for subsequent supervised classification,.	The Enron dataset provides only one attribute (emails), which is insufficient for a complete analysis of user behavior without psychological or web activity features.
Kamatchi (2025)	**Hybrid** (Deep Learning within Federated Learning)	**Bi-LSTM** (Bidirectional Long Short-Term Memory)	**Supervised**	Employs privacy-preserving **Federated Learning** to collaboratively train models on edge devices without centralizing raw data, using Bi-LSTMs to capture sequential behavior in both forward and backward directions.	Federated environments are susceptible to **attacks targeting deployment**, and **non-IID data** can cause model updates to become biased, slowing convergence and reducing accuracy.
Kong (2025)	**Hybrid** (evaluated ML-XGB and DL-FMLP)	**FMLP** (Filter-enhanced MLP) and **XGB** (eXtreme Gradient Boosting)	**Supervised**	Introduces **symbolic tagging** to filter redundant system noise and an **adaptive embedding mechanism** that dynamically adjusts context windows for rare but critical behavior fragments.	Fine-grained mapping can lead to **high memory overhead**; the framework currently lacks an **adaptive threshold mechanism** for evolving behavioral patterns.
Kotb (2025)	**Deep Learning**	**Fully connected binary DL** (10 hidden blocks with Batch Normalization and Leaky ReLU) Copula GAN	**Supervised**	Leverages **Deep Feature Synthesis (DFS)** to automatically generate complex tabular user profiles from raw logs, specifically designed to detect both traditional insiders and **AI-generated synthetic threats**.	Primarily evaluated on **synthetic datasets** and designed for **offline learning**, necessitating periodic manual retraining to stay current with new threat vectors.
Lavanya (2024)	**Hybrid** (ML + DL)	**EBiGAN** (Generative) and **DNN** (Detection),	**Supervised**,	Uses an **Enhanced Bidirectional GAN** with an additional	Faces significant **noise robustness** issues, **computational complexity**,

				discriminator to generate high-quality samples for balanced data, followed by a **DNN fine-tuned via Bayesian Optimization** (PI function),.	and a lack of awareness regarding external cyber attacks,.
Lavanya (2025)	**Hybrid** (DL + Graph)	**AGCN** (Attention Graph Convolutional Network) and **ABWGAN**,	**Supervised**,	Combines an **ABWGAN-EHI** (optimized for stability) with an **L2-SP regularized pretrained AGCN** that utilizes attention mechanisms to capture relational behavioral patterns,,.	Exhibits **complexity in diverse physical locations** and suffers from **"non-existent interpretation"** (lack of interpretability) in real-time environments.
Le & Zincir-Heywood (2021)	**Ensemble**	**AE, IF, LODA, and LOF**,	**Unsupervised**,	Employs an ensemble of four diverse algorithms combined with **percentile-based temporal representations** to identify transitions in behavior as anomalies relative to a user's recent history,.	**LOF** suffers from very long training and prediction times,; **Autoencoder (AE)** performance deteriorates significantly under **training data poisoning**
Li et al. (2024)	**Hybrid** (Graph Neural Network + Meta-Learning)	**GMFITD** (incorporating GAE, GCN, and MAML)	**Few-shot Learning** (Supervised within episodes)	Combines a **structural reconstruction mechanism** to infer latent **"soft" relationships** between users with **episodic meta-learning** to detect threats using limited labeled samples,,.	High costs and resource consumption; the **assumption that user behavior is static** over long periods is often inaccurate in real-world scenarios.
Li et al. (2023)	**Deep Learning**	**DD-GCN** (Dual-Domain Graph Convolutional Network)	**Semi-supervised**	Propagates node features across both **topology and feature domains** simultaneously, using an **attention mechanism** to adaptively weight and fuse the most relevant embedding information,,.	Higher **computational complexity and FLOPs** than standard GCN architectures due to the dual convolutional operations and attention processing.
Liu et al. (2025)	**Deep Learning / Knowledge Graph**	**BERT** architecture and **Attack Knowledge Graph**	**Self-supervised** (Pretraining) and **Supervised** (Contrastive learning)	Decomposes attack sequences into discrete **"motive" and "behavioral" evidence** components, using a preference propagation mechanism to predict malicious acts.	Requires a labeled dataset of malicious users to initial the evidence sequences; graph modeling can lead to **high computational costs** due to redundant information.

Mehmood et al. (2023)	**Ensemble**	**LightGBM, XGBoost, AdaBoost, and Random Forest**	**Supervised**	Implements multiple ensemble boosting algorithms to detect and classify **privilege escalation** anomalies in cloud-based activity logs.	Individual models can be **noisy**; traditional ML lacks the ability to automatically design features and has difficulty identifying tiny mutants of known attacks.
Medvedev et al. (2025)	**Deep Learning**	**Siamese Neural Networks (SNN)** with **CNN** branches	**Supervised** (Triplet Loss)	Transforms 1D keystroke time-series into **2D images (GAFMAT)** and uses **interpolation-based data fusion** to standardize variable password lengths.	Behavioral models exhibit high variability due to **physical or emotional states** (stress, fatigue); performance depends significantly on standardized input data.
Mehnaz (2021)	**Classical ML / Pattern Matching**	**Finite State Automata** and **Unsupervised Clustering**	**Unsupervised**	Models behavior at the **fine-grained block level** using "episodes" (serial or parallel task-based event sequences) to detect anomalous file access sizes and segments.	Cannot currently detect **collusion attacks** where groups of insiders collaborate to evade individual thresholds.
Mladenovic (2024)	**Ensemble**	**XGBoost** and **AdaBoost** tuned by **HARFO** (Hybrid Adaptive Red Fox Optimization)	**Supervised**	Analyzes the **sentiment and context** of communications using TF-IDF, focusing on behavioral indicators that are less prone to change than physical metrics like location.	Evaluated on **simulated data** due to real-world sample constraints; faced computational limits on the number of agents used for optimization.
Nasir (2021)	**Deep Learning**	**LSTM-Autoencoder**	**Unsupervised**	Utilizes a **flexible user-session time window** (Logon to Logoff) instead of fixed windows to capture behavioral sequences and organizational role data.	Limited by the **scarcity of diverse, publicly available** insider threat scenarios for robust system training.
Nikiforova et al. (2024)	**Classical ML**,	**K-means** refined with **Elbow method** and **Markov chains**,	**Unsupervised**,	Automatically **clusters users into similar groups** based on behavioral graphs, then identifies anomalies by comparing individuals to their group's model,.	**Scalability and resource requirements** for large organizations; inability to definitively confirm if atypical behavior represents real security incidents,.
Pal et al. (2023)	**Ensemble / Deep Learning**,	**Stacked-LSTM** and **Stacked-GRU** with **Attention Mechanisms**,	**Supervised**,	Employs an ensemble of deep sequential models to learn **nonlinear dependencies** in activity sequences, using attention	**High memory overhead** due to one-hot encoding; modeling is limited to single-day sequences, potentially **missing multi-day attack patterns**,.

				to **prioritize critical sections** of logs,.	
Patel & Iyer (2025)	**Deep Learning**	**SiaDNN** (incorporating **Siamese CNN** and **DNN**),	**Not Specified** (Mentions difficulty in obtaining ground truth labels)	Integrates Siamese networks with the **FP Growth algorithm** to analyze recurring sequences of user interaction patterns, such as administrative and authentication actions,.	**Struggles to distinguish data errors** from genuine anomalies; difficulty in generalization to other application domains.
Peccatiello et al. (2023)	**Classical ML**	**Isolation Forest (ISOF)**, **Elliptic Envelope**, and **Local Outlier Factor**,	**Semi-supervised** (with a supervised multi-class baseline for comparison),	Proposes a **stream data analysis framework** using one-class classification and **periodic retraining** to maintain resilience against concept drift,.	One-class algorithms can be **noisy**, leading to high false positives; difficulty in obtaining **contamination-free** training samples in real-world environments,.
Pennada et al. (2024)	**Ensemble**,	**Stacking** and **Voting** classifiers combining Random Forest, Adaboost, and Decision Tree,	**Supervised**,	Resolves class imbalance using a pipeline of **three oversampling and three undersampling techniques** before applying ensemble learners,.	Tested only on the **CERT dataset**; requires evaluation on more diversified insider threat datasets.
Pennada et al. (2025)	**Hybrid (ML + DL)**,,	**DAEs** and **VAEs** (Generative) combined with **Random Forest** and **XGBoost** (Discriminative),,	**Hybrid** (Unsupervised feature extraction and Supervised classification),,	Fuses **traditional behavioral features with latent representations** extracted from generative models to capture both overt and subtle malicious patterns,,.	Needs improvement in **real-time adaptability**; future work must address data privacy via federated learning,.
Perez-Miguel et al. (2025)	**Not Specified**	**Not Specified** (Focuses on dataset design rather than training specific models),	**Not Specified**	Proposes the **SPEDIA dataset** which integrates real user behavior from cyber exercises with simulated role-based activity,.	Non-malicious data is synthetic and **driven by models**; real human behavior is more chaotic and contains unexpected changes.
Qawasmeh & AlQahtani (2025)	**Ensemble / Classical ML**,	**XGBoost** (primary), Random Forest, SVM, and Logistic Regression,,	**Supervised**,,	Combines **real-time monitoring with a weighted risk scoring system** (17 activity types) to dynamically profile users into three risk levels,,.	Relies on **synthetic data**; focus on technical indicators may miss psychological factors like employee stress or job satisfaction.
Randive et al. (2023)	**Deep Learning**,	**Wavelet Convolutional Neural Network (WCNN)** based on VGG-19,	**Supervised**,	Transforms activity logs into **1D feature vectors represented as grayscale images**, utilizing spectral and spatial analysis to	Use of wavelet parameters is currently limited; only **Haar Wavelets** were employed in the model.

				prevent feature loss during pooling,,.	
Rauf et al. (2021)	**Hybrid** (ML + Policy Transition System)	**DBSCAN**, **Random Forest**, **SVM**, and **Z3 SMT solver**	**Unsupervised** and **Supervised**	Uses a bio-inspired framework mapping cellular regulation to an "Access DNA" model for autonomous, real-time policy regulation.	Accuracy degrades when trained over temporal windows larger than 3 weeks due to model over-approximation and increased false positives.
Roy & Chen (2024)	**Deep Learning** (GNN)	**GraphCH** framework using **CH-GLM** and **BiLSTM**	**Supervised** (multi-class classification)	Represents behavior as a Heterogeneous Cyber-Human Behavioral Graph (HetG-CH) that integrates host logs with psychological traits (impulsiveness and risk-taking).	Small study population (35 participants) and inability to assess new or temporary employees without established historical cyber-psychological profiles.
Bin Sarhan & Altwaijry (2023)	**Hybrid** (Classical ML + DL)	**SVM**, **Neural Network**, **AdaBoost**, **Random Forest**, **OCSVM**, and **iForest**	**Supervised** and **Unsupervised** (Anomaly Detection)	Utilizes the Deep Feature Synthesis (DFS) algorithm to automatically generate nearly 70,000 behavioral features from relational logs to capture human-intuition-driven patterns.	High feature counts from DFS can lead to overfitting; the system also requires significant retraining when log structures change or when new users are enrolled.
Senevirathna et al. (2025)	**Hybrid** (ML + DL)	CNN and Random Forest (for behavior); MobileNetV2 and LSTM (for physical security)	**Supervised**	Integrates human behavior analysis (logon/device logs) with real-time physical surveillance (CCTV) in a unified framework to detect cyber-physical threats.	Assumes constant availability of high-quality data, risks oversimplifying psychological factors in psychometric analysis, and relies on centralized processing which may cause delays in large-scale deployments.
Song et al. (2024)	**Deep Learning** (Autoencoder)	Stacked Bidirectional LSTM (BiLSTM) and Feedforward Neural Network (FNN)	**Semi-supervised**	Implicitly encodes absolute time into behavioral sequences and uses a covariance-based adaptive construction method to fit unique "user-day" behavioral rhythms.	The detection granularity is not fine enough, it fails to account for behavioral rhythms that change over time, and shows lower performance in specific threat scenarios (Scenario 2).
Tabassum et al. (2024)	**Hybrid** (Unsupervised + Supervised)	Isolation Forest (IForest) or Local Outlier Factor (LOF) combined with SVM, Decision Tree, or Random Forest	**Unsupervised** (for labeling) and **Supervised** (for modeling)	Identifies new contextual anomalies in Electronic Health Records (EHR) by utilizing cross-correlation for feature selection to reduce computational complexity and redundancy.	Legal constraints limit access to large volumes of sensitive EHR data, and evolved medical practices may flag benign actions as anomalies.

Tian T et al. (2025) [ITDSTS]	**Deep Learning**	**Transformer Encoder** (Multi-head attention + Positional encoding)	**Supervised**	Dynamically updates user behavior vectors with semantic information (TF-IDF) to detect three specific threat scenarios: privilege abuse, identity theft, and data leakage.	Struggles to capture implicit relationships when the proportion of normal and abnormal behavior is seriously unbalanced.
Tian Z et al. (2024) [DSDLITD]	**Deep Learning**	**Attention-LSTM** combined with **Dempster-Shafer (D-S) fusion engine**	**Unsupervised** (Anomaly Detection)	Employs a multichannel framework where multi-head attention captures complex data abstractions and D-S theory fuses evidence from various classifiers to handle accidental and intentional threats.	Computation and memory overhead grows significantly with large datasets due to input-embedding matrix projections.
Villarreal-Vasquez et al. (2023)	**Deep Learning**	**LADOHD** (Three-layer **LSTM** encoder followed by a linear layer)	**Unsupervised** (Anomaly Detection)	Models high-dimensional system activity events as a "structured language" to identify anomalous execution patterns and long-term dependencies in sequences.	Requires further evaluation with different datasets and performance testing in live production environments.
Wall & Agrafiotis (2021)	**Classical ML**	**Bayesian Network** utilizing the **Bayes-Ball algorithm** for inference	**Unsupervised** (Anomaly Detection)	Learns causal relationships and dependence properties from behavior data to create a normal profile, allowing for the potential integration of prior psychological knowledge.	Difficulty in detecting "build-up" activities (subtle threats) that resemble normal behavior; inference is NP-Hard with exponential worst-case time.
Wang & El Saddik (2023)	**Deep Learning**	**Transformer** (Custom: DistilledTrans), **BERT**, and **RoBERTa**.	**Supervised**	Integrates **Digital Twin** technology with NLP self-attention models and uses **BERT/GPT-2** for context-aware data augmentation to fix extreme class imbalance.	Complex hybrid models can be too deep, leading to information loss; classic RNNs fail to learn long-term temporal patterns.
Wang Zhi et al. (2024) FedITD	**Deep Learning**	**LLMs** (**BERT, RoBERTa, XLNet, DistilBERT**) with **PETuning** (**LoRA, Adapter, BitFit**).	**Supervised** (Federated)	Combines **Federated Learning** with Parameter-Efficient Tuning to enable localized, privacy-preserving detection across distributed organizations without sharing raw data.	LLMs introduce significant communication and storage overhead; global models often struggle with **domain shift** on specific client data distributions.

Wang Jiarong et al. (2023) Deep Cluster	**Deep Learning**	**Encoder-Decoder** (RNN/GRU layers) with **Deep Clustering**.	**Unsupervised**	Automatically learns behavior representations through a multi-output decoder that predicts the **latent generating function** of future event entities to optimize clustering centroids.	Accuracy is highly sensitive to the predetermined number of clusters (*k*); existing deep clustering for images is not directly applicable to audit logs.
Wei Yichen et al. (2021)	**Deep Learning**	**Cascaded Autoencoders (CAEs)**, **BiLSTM**, and **Hypergraph** correction.	**Unsupervised**	Utilizes cascaded autoencoders for **"data purification"** to filter anomalies from unlabeled sets and a hypergraph module to correct false positives in proactive investigations.	Purification may drop normal data with abnormal appearances, raising false positive risks; the number of CAE filters (*K*) must be defined empirically.
Wei Zhiyuan et al. (2024)	**Hybrid** (ML + Statistical)	**Local Outlier Factor (LOF)** and **Information Gain** (Entropy-based).	**Unsupervised**	Employs a **"Divide & Conquer"** approach to create highly personalized behavioral profiles for individual employees rather than a single aggregated global model.	Requires a malicious sample **Impact Ratio (IR)** of at least 3% to avoid under-fitting; LOF algorithms suffer from high runtimes on very large datasets.
Wen et al. (2023)	**Network Analysis** (Statistical),.	**Singular Value Decomposition (SVD)** and **Eigenvector Centrality**,.	**Unsupervised**,.	First to use **SVD** to extract character patterns from **sentiment communication networks** to locate anomaly-related employee groups,.	The **VADER** model has a poor effect on capturing **minor emotions**, which can make the direction of the entire network confusing,.
Xiao Junchao et al. (2023)	**Deep Learning**,.	**MEWRGNN** (Relational GCN, GCN, and CAN-Graph Attention Networks),.	**Supervised**,.	Captures **contextual relationships** of user behaviors via a multi-edge weight graph representing **time intervals, Euclidean distance (similarity), and attention**,.	The hourly threat detection delay may still be slightly longer; **more real-time techniques** need further study,.
Xiao Fengrui et al. (2025)	**Deep Learning**,.	**SENTINEL** (ST-GNN using GCN + GRU), **EGAT**, and **HCA**,.	**Semi-supervised** (trains on benign data only),.	Devises a **Behavior Interaction Graph (BIG)** that treats interaction logs as **edges rather than nodes** to reduce computational overhead for large-scale graph learning,.	Prone to misjudging malicious actions that **closely resemble routine operations**, particularly for high-authority administrators,.
Xiao Haitao et al. (2024)	**Deep Learning**,.	**CATE** (Convolutional Attention and Transformer Encoder),.	**Supervised**,.	Integrates **statistical and sequential information** in parallel modules to learn patterns	Performance decreases when malicious activities represent a **very low proportion** of total daily

				from two distinct dimensions of user behavior,.	activities; struggles with highly concealed scenarios like **IPT**,.
Ye Xiaoyun et al. (2025)	**Deep Learning** (Federated),.	**SqueezeNet** (lightweight CNN) within a Federated Learning framework,.	**Supervised**,.	Employs **DeepInsight** to convert numerical logs into **grayscale image formats**, allowing CNNs to capture complex feature relationships while preserving privacy,.	Lacks **real-time detection** capability and assumes participants are "honest but curious" without defenses against **adversarial attacks** in the FL environment,.
Yildirim & Anarim (2022)	**Ensemble**,.	**XGBoost** (Gradient Boosted Decision Trees),.	**Supervised**,.	Assigns **legality scores** to individual mouse actions based on time and frequency domain features, then aggregates them to assign a probability to the session,.	Difficulty capturing **higher-level relationships**, such as the specific sequence of emergence of successive movement types,.
Zhu et al. (2024)	**Deep Learning**,.	**TL-AAE** (TCN + LSTM Adversarial Autoencoder),.	**Unsupervised**,.	Introduces **Generative Adversarial Theory** (discriminator) to align the encoder's latent distribution with a Gaussian prior to reduce **reconstruction uncertainty**,.	Utilizes time information in a **relatively simple manner**; needs better integration of time and event frequency features,.

The following is resulting table after grouping the studies into 5 groups according to their model family

Model Family	**Specific Architectures**	**Count (N)**	**Studies**
1. Deep Sequential, Attention & CNNs	**Architectures:** LSTM, Bi-LSTM, GRU, CNN (1D/2D), Transformers (BERT, RoBERTa), LLMs (LLaMA), Siamese Networks. **Focus:** Capturing temporal dependencies, semantic context, and visual patterns using deep neural networks.	**25**	Ali (2025), ALmihqani (2021), Al-Shehari (2024, CNN), Alshehri (2022), Amuda (2022), Anakath (2022), Anju (2024), Ferraro (2025), Gayathri (2025, Adv), Gayathri (2025, Cloud), Hafizu Rhman (2022), Huang (2025), Kamatchi (2025), Liu (2025), Medvedev (2025), Pal (2023), Patel & Iyer (2025), Randive (2023), Tian T (2025), Tian Z (2024), Villarreal-Vasquez (2023), Wang & El Saddik (2023), Wang Zhi (2024), Xiao Haitao (2024), Ye Xiaoyun (2025)

2. Classical & Statistical ML	**Architectures:** Isolation Forest, SVM, Random Forest, Naive Bayes, K-Means, Bayesian Networks, Finite State Automata, SVD. **Focus:** Establishing baselines using interpretable, computationally efficient algorithms on tabular data.	**20**	Ahmadi (2025), Ahmed (2025), Al-Shehari (2021), Al-Shehari (2023, IF), AL-SHEHARI (2024, LOF), Asha S (2023), Eshmawi (2026), Feng (2025), Gonzales (2025), Jaiswal (2024), Janjua (2021), Mehnaz (2021), Nikiforova (2024), Peccatiello (2023), Perez-Miguel (2025), Rauf (2021), Tabassum (2024), Wall & Agrafiotis (2021), Wei Zhiyuan (2024), Wen (2023)
3. Ensemble & Hybrid Frameworks	**Architectures:** Boosting (XGBoost, LightGBM, AdaBoost), Stacking, Voting, Hybrid (ML + DL fusion), Federated Aggregation. **Focus:** Reducing variance and improving robustness by combining multiple "weak learners" or fusing deep and shallow models.	**20**	Adun (2023), Alabdulkareem (2022), Al Hammadi (2021), Almusawi (2024), AL-Mihqani (2022), Al-Shehari & Alsawail (2023), Gayathri (2024, Hybrid), Gupta (2024), Haq (2022), He (2022), He (2024), Kong (2025), Le & Zincir-Heywood (2021), Mehmood (2023), Mladenovic (2024), Pennada (2024), Qawasmeh (2025), Bin Sarhan (2023), Senevirathna (2025), Yildirim (2022)
4. Generative & Reconstruction Models	**Architectures:** Autoencoders (AE, VAE), GANs (SPCAGAN, CopulaGAN), Diffusion Models, Deep Clustering. **Focus:** Unsupervised learning of "normal" behavior distributions to detect anomalies via reconstruction error.	**10**	Amiri-Zarandi (2023), Dong J (2025), Kotb (2025), Lavanya (2024), Nasir (2021), Pennada (2025), Song (2024), Wang Jiarong (2023), Wei Yichen (2021), Zhu (2024)
5. Graph Neural Networks (GNNs)	**Architectures:** GCN, GAT, Heterogeneous Graphs, Knowledge Graphs. **Focus:** Modeling complex topological relationships between users, devices, and files.	**7**	Cai X (2024), Lavanya (2025), Li et al. (2023), Li et al. (2024), Roy & Chen (2024), Xiao Junchao (2023), Xiao Fengrui (2025)

Total		**82**	

We used Google Gemini and **manual** cross-referencing to ensure that the data and grouping are accurate.

Seq.	**Model Family (from Table IV)**	**Ref #**	**Author Name (Reference List)**
1	Deep Learning (Sequential & CNN)	**[8]**	Ferraro (2025)
2	Deep Learning (Sequential & CNN)	**[9]**	Gayathri (2025, Cloud)
3	Deep Learning (Sequential & CNN)	**[16]**	Gayathri (2025, Adv)
4	Deep Learning (Sequential & CNN)	**[26]**	Ali (2025)
5	Deep Learning (Sequential & CNN)	**[38]**	ALmihqani (2021)
6	Deep Learning (Sequential & CNN)	**[41]**	Amuda (2022)
7	Deep Learning (Sequential & CNN)	**[42]**	Anakath (2022)
8	Deep Learning (Sequential & CNN)	**[46]**	Huang (2025)
9	Deep Learning (Sequential & CNN)	**[47]**	Kamatchi (2025)
10	Deep Learning (Sequential & CNN)	**[51]**	Patel & Iyer (2025)
11	Deep Learning (Sequential & CNN)	**[57]**	Tian Z (2024)
12	Deep Learning (Sequential & CNN)	**[59]**	Xiao Haitao (2024)
13	Deep Learning (Sequential & CNN)	**[61]**	Al-Shehari (2024, CNN)
14	Deep Learning (Sequential & CNN)	**[67]**	Tian T (2025)
15	Deep Learning (Sequential & CNN)	**[72]**	Liu (2025)
16	Deep Learning (Sequential & CNN)	**[73]**	Wang & El Saddik (2023)
17	Deep Learning (Sequential & CNN)	**[74]**	Wang Zhi (2024)
18	Deep Learning (Sequential & CNN)	**[75]**	Alshehri (2022)
19	Deep Learning (Sequential & CNN)	**[76]**	Anju (2024)
20	Deep Learning (Sequential & CNN)	**[80]**	Hafizu Rhman (2022)
21	Deep Learning (Sequential & CNN)	**[83]**	Pal (2023)
22	Deep Learning (Sequential & CNN)	**[85]**	Villarreal-Vasquez (2023)
23	Deep Learning (Sequential & CNN)	**[94]**	Medvedev (2025)
24	Deep Learning (Sequential & CNN)	**[95]**	Randive (2023)
25	Deep Learning (Sequential & CNN)	**[96]**	Ye Xiaoyun (2025)
26	Classical Machine Learning	**[10]**	Peccatiello (2023)

27	Classical Machine Learning	**[13]**	Al-Shehari (2024, Isolation Forest)
28	Classical Machine Learning	**[22]**	Wei Zhiyuan (2024)
29	Classical Machine Learning	**[23]**	Feng (2025)
30	Classical Machine Learning	**[28]**	Rauf (2021)
31	Classical Machine Learning	**[29]**	Al-Shehari & Alsowail (2021)
32	Classical Machine Learning	**[36]**	Ahmed (2025)
33	Classical Machine Learning	**[39]**	Al-Shehari & Alsowail (2023)
34	Classical Machine Learning	**[43]**	Asha S (2023)
35	Classical Machine Learning	**[44]**	Eshmawi (2026)
36	Classical Machine Learning	**[50]**	Nikiforova (2024)
37	Classical Machine Learning	**[53]**	Perez-Miguel (2025)
38	Classical Machine Learning	**[56]**	Tabassum (2024)
39	Classical Machine Learning	**[58]**	Wall & Agrafiotis (2021)
40	Classical Machine Learning	**[65]**	Janjua (2021)
41	Classical Machine Learning	**[68]**	Wen (2023)
42	Classical Machine Learning	**[79]**	Gonzales (2025)
43	Classical Machine Learning	**[81]**	Jaiswal (2024)
44	Classical Machine Learning	**[82]**	Mehnaz (2021)
45	Classical Machine Learning	**[88]**	Ahmadi (2025)
46	Ensemble & Hybrid Frameworks	**[24]**	Le & Zincir-Heywood (2021)
47	Ensemble & Hybrid Frameworks	**[25]**	He (2024)
48	Ensemble & Hybrid Frameworks	**[27]**	Kong (2025)
49	Ensemble & Hybrid Frameworks	**[30]**	Mehmood (2023)
50	Ensemble & Hybrid Frameworks	**[32]**	Bin Sarhan (2023)
51	Ensemble & Hybrid Frameworks	**[35]**	Adun (2023)
52	Ensemble & Hybrid Frameworks	**[37]**	Alabdulkareem (2022)
53	Ensemble & Hybrid Frameworks	**[45]**	Gupta (2024)
54	Ensemble & Hybrid Frameworks	**[52]**	Pennada (2024)
55	Ensemble & Hybrid Frameworks	**[54]**	Qawasmeh (2025)
56	Ensemble & Hybrid Frameworks	**[55]**	Senevirathna (2025)
57	Ensemble & Hybrid Frameworks	**[60]**	Yildirim (2022)

58	Ensemble & Hybrid Frameworks	**[62]**	Al-Shehari (2023)
59	Ensemble & Hybrid Frameworks	**[63]**	AL-Mihqani (2022)
60	Ensemble & Hybrid Frameworks	**[64]**	Almusawi (2024)
61	Ensemble & Hybrid Frameworks	**[66]**	Mladenovic (2024)
62	Ensemble & Hybrid Frameworks	**[69]**	Gayathri (2024 Hybrid)
63	Ensemble & Hybrid Frameworks	**[70]**	Haq (2022)
64	Ensemble & Hybrid Frameworks	**[71]**	He (2022)
65	Ensemble & Hybrid Frameworks	**[93]**	Al Hammadi (2021)
66	Generative & Reconstruction Models	**[14]**	Pennada (2025)
67	Generative & Reconstruction Models	**[15]**	Zhu (2024)
68	Generative & Reconstruction Models	**[31]**	Kotb (2025)
69	Generative & Reconstruction Models	**[40]**	Amiri-Zarandi (2023)
70	Generative & Reconstruction Models	**[48]**	Lavanya (2024)
71	Generative & Reconstruction Models	**[49]**	Nasir (2021)
72	Generative & Reconstruction Models	**[78]**	Dong J (2025)
73	Generative & Reconstruction Models	**[84]**	Song (2024)
74	Generative & Reconstruction Models	**[86]**	Wang Jiarong (2023)
75	Generative & Reconstruction Models	**[87]**	Wei Yichen (2021)
76	Graph Neural Networks (GNNs)	**[33]**	Li et al. (2023, DD-GCN)
77	Graph Neural Networks (GNNs)	**[34]**	Lavanya (2025)
78	Graph Neural Networks (GNNs)	**[77]**	Cai X (2024)
79	Graph Neural Networks (GNNs)	**[89]**	Li et al. (2024, GMFITD)
80	Graph Neural Networks (GNNs)	**[90]**	Roy & Chen (2024)
81	Graph Neural Networks (GNNs)	**[91]**	Xiao Junchao (2023)
82	Graph Neural Networks (GNNs)	**[92]**	Xiao Fengrui (2025)