This document contains a comparative analysis of the effectiveness of ML-based insider threat detection models across the 82 primary studies. The authors extracted standard performance indicators Using **NotebookLM** (processing 2–5 papers per prompt). These included Accuracy, Precision, Recall, and F1-Score. This was also done alongside operational metrics such as detection latency and computational overhead. Each extracted metric was manually validated against the results sections of the primary studies to ensure the highest degree of data integrity. **When the authors have mixed results, we reported the highest numbers.**

Prompt 4: Evaluation Metrics & Performance (RQ4)

Task: You are a research assistant specializing in evaluating Machine Learning models. Analyze the uploaded papers listed below and fill the **Evaluation Metrics Table** based on the following criteria.

Instructions for Columns:

- **Authors (year):** Use the ID and Author/Year (e.g., Adun (2023)).
- **Algorithm(s):** List the primary models tested (e.g., SVM, LSTM, GNN).
- **Acc, Pre, Rec, F1, AUC, MCC, FPR:** Extract the highest numerical values (percentages or decimals) for Accuracy, Precision, Recall, F1-score, Area Under the Curve, Matthews Correlation Coefficient, and False Positive Rate.
- **Other:** List any secondary metrics such as "Response Time," "Error Rate," "Kappa," or "Training Time."
- **Baseline / Comparative Models:** Identify which models or existing studies the authors compared their results against (e.g., "Compared against Random Forest" or "Baseline: CMU CERT Leaderboard").

Output Format: Provide the results in a table with these headers: | Authors (year) | Algorithm(s) | Acc | Pre | Rec | F1 | AUC | FPR | Other | Baseline / Comparative Models |

Constraint: Use ONLY information found in these papers. If a metric is not reported, use a dash (—) or write "Not specified." Ensure numerical values are accurately transcribed. Include links to the text of the papers.

Paper ID	Algorithm (s)	Evaluation Metrics							Baseline / Comparative Models Used
		Acc	Pre	Rec	F1	AUC	FPR	Other	
P01 Adun et al (2023)	SVM, ANFIS	92 %SVM 91%ANFIS	93%	-	-	-	-	Error 9 %	Not explicitly stated
P02 Alabdulkareem (2022)	MWF-IDLDC (LSTM, GRU, SAE, ALO)	99.10%	98.61%	98.60%	99.10%	99.10%	-	-	LR, DT, RF, HM, DNN, NBN
P03 Ahmadi et al (2025)	Random Forest, Gradient Boosting, K-means	92%	-	-	-	-	6.3%	Response Time: within seconds	Traditional static rule-based system
Ahmed 2025	Random Cut Forest (RCF)	-	70 %-97%	-	72 %-99%	-	-	TPR: 0.95 Detection Time: 1-20 mins	Not explicitly stated
Alhammadi et al 2021	Adaptive Boosting, Random Forest, 2D CNN, 1D CNN, KNN	97%	-	-	-	-	-	McNemar's test P-value: 0.00007 (2D CNN vs 1D CNN), 0.0013 (RF vs KNN)	2D CNN, 1D CNN, Adaptive Boosting, Random Forest, KNN (compared against each other)
ali-et-al-2025	BERT, BERTopic, Ensemble Model (BERT + BERTopic + multi-class logistic regression)	96%	-	-	-	-	-	Detection Accuracy Rate (DAR)	Internal comparison only Model 1 (baseline model), Model 2 (focused model)
ALmihqani et al 2021	AD-DNN (ADASYN + Deep Neural Network)	96%	-	-	95%	95%	4%	FNR: 5%	SVM, DNN, LSTM, OCSVM based on DBN, LSTM Autoencoder
Almusawi 2024	ML + Expert Policies	**99 %**	100%	94%	97%	-	-	-	Logistic Regression, Decision Tree, Random Forest, XGBoost, AdaBoost, Naive Bayes, SVM, KNN,

									MLP, Linear SVC, Voting Classifier
AL-Mihqani 2022 A new intelligent multilayer	HITD (Random Forest + K-Nearest Neighbors)	96%	74.2%	84%	95%	95%	2.88 %	Computation Time: 3.663 minutes	LOF, KNN, HMM, various DNNs
Alshehari 2023 (IF)	Isolation Forest (IF)	98%	-	98%	**99%**	-	-	-	Logistic Regression, Decision Trees, Random Forest
Al-shehari & Alsawail 2023	XGBoost, RF, DT, KNN	-	84%	67%	67%	94%	-	-	
Al-shehari (CNN) (2024)	CNN+ ADASYN					96%			
AL-SHEHARI (LOF) (2024)	DBLOF	-	-	-	99%	-	-	Detection Rate 98%	XGBoost, RF , DT , KNN
Alshehri (2022)	Rel-RNN (LSTM units with Direct Graphs)	—	99.12%	67.12%	0.80	0.99	—	Mean Squared Error (MSE)	SVM, HMM, and shallow Neural Network (NN)
Amiri-Zarandi (2023)	Deep Autoencoder (Federated Learning)	—	—	0.97	—	0.93	0.20	Investigation Budget (20%); Loss (MSE)	Individual local models; Centralized solutions
Al-Shehari (2021)	DT + SMOTE; RF + SMOTE	—	0.99	1.00	0.99	1.00	-	-	Label Encoding; One-hot Encoding
Amuda (2022)	Hybrid CNN-GRU	97.39%	99.96%	97.37% (Sensitivity)	—	0.90	—	Model Loss: 0.12	Single LSTM, CNN, and GRU models
Anju (2024)	**Stacked CNN-Attentional BiGRU**	92.52%	98%	95%	96%	0.95	96% (Reported as FAR)	Training Time: 0.2 s; Prediction Time: 1.5–2.3 s	FCVM, LSTM, ML, DL-BERT, CBSigIDS, TLDANN
Anakath (2022)	**Deep Belief Neural Network (DBN)**	99%	—	—	98% (F-Measure)	—	—	Evaluated with mouse/keystroke data blocks	SVM and LSTM

Asha S (2023)	Double-layer architecture using NM-2 sampling and OCSVM,	82.46%,	64.92% ,	100%,,	78.72%,,	—	—	Model Loss: 0.12; Simulation Parameters: RBF kernel, Nu 0.02	Adaboost, LightGBM, SOM, RF, CGAN, LSTM, CNN, GCN
Cai X (2024)	LAN (Learning Adaptive Neighbors) using LSTM and GCN,	—	—	0.9478 (reported as DR)	—	0.9607	0.0865	Inference time: 0.30ms; retrieval threshold ϵ: 0.5,	DeepLog, Transformer, RWKV, TIRESIAS, DIEN, BST, FMLP, log2vec
Dong J (2025)	DDCC (Denoising Diffusion Probabilistic Models + Curriculum Learning)	—	—	—	—	0.9823	—	EER (Equal Error Rate); Inference Latency: 37.8ms; Training Time: 8.4h	RF, LR, XGBoost, LADOHD, UIBD_ARAE, TranAD, LAnoBERT, TCF-Trans
P01: Eshmawi (2026)	SVM, RF, KNN, DNN, NB	99.9% (KNN)	1.00 (100%)	1.00 (100%)	1.00 (100%)	—	0.18–1%*	p-value: 0.042; t-statistic: 2.11	Adaboost, LightGBM, SOM, RF, CGAN, LSTM, CNN, GCN
P02: Feng (2025)	MG-UABD (Random Forest)	99.99%	99.99%	99.99%	99.99%	—	9.69%	Complexity: O(D×logD×M×T)	S-LSTM, ITDBERT, SPYRAPTOR, SeqA-ITD, RAP-Net, LSTM-Autoencoder
P03: Ferraro (2025)	GABM (LLaMA-3.1 8B)	95.85%	35.82%	100.00%	52.74%	—	5.95%	Chain-of-Thought (CoT) reasoning logs	DGCNN, GAT, GATV2, GCN, GIN, GatedGraphConv, Node2Vec
Gayathri (2025) Adversarial Training for Backdoor Attacks	SNN-MLP, SNN-1DCNN, TabNet, XGBoost (XGB), RF, LGB	-	**1.000**	**0.973**	**0.916**	—	—	**Kappa:** 0.963; **MCC:** 0.963; **ASR:** 100%	RF, XGB, LGB, ROS, SMOTE, VAE, CGAN, ACGAN, CWGAN-GP, FGSM, DeepFool, CW, JSMA
Gayathri, B. (2025) ITD-	Jordan Neural Network	**97%**	**95%**	**96%**	**97%**	—	**3%**	**Training Time:** 0.823s;	LSTM, DBN, DNN, CNN, CNN-BiLSTM,

GMJN: Insider Thread Detection	(JNN), LS-AE, LSTM, DBN, DNN							**Execution Time:** 0.8298s; **Miss Rate:** 8%	ORC_NN, Osprey Optimization (OA), Black Widow (BWO)
Gayathri (2024) Hybrid Deep Learning Model	SPCAGAN, Hybrid Bayesian Neural Network (BNN), MLP, 1DCNN	-	**0.8956**	**0.9902**	**0.9198**	—	—	**Kappa:** 0.8727; **MCC:** 0.8729; **SPCA Similarity:** 3.868	ROS, SMOTE, Random Noise (RN), GMM, CGAN, CWGAN-GP, ACGAN, FGSM, DeepFool
Gonzales (2025)	C3P, C3P-SMART	0.9	0.90	0.90	0.90	0.96	—	**Runtime:** 205.24s; **Complexity:** $O(N \cdot L_2)$	Isolation Forest, One-Class SVM, Local Outlier Factor
Gupta (2024)	FedMUP (AFed, CFed, DFed)	96.73%	96.84%	96.73%	96.70%	—	—	**Loss:** 0.0659; **Time:** 367.728; **Memory:** 18.92	ABBAC, GAM, DT-ILIS, MLPAM, QM-MUP, XGBoost, SeCoM
Hafizu Rhman (2022)	NARX (Nonlinear Autoregressive Exogenous)	99.12%	99.05%	99.05%	—	—	21.13%	R_2: 0.97; **RMSE:** 1.0478; **MAE:** 0.97529	ARIMA, LSTM, Regression, ANN, IFSM; Mazzarolo et al.
Haq (2022)	**XGBoost** (Best), Word2vec-LSTM, GLoVe-LSTM, AdaBoost, RF, KNN, LR	0.92	0.92	1.00	0.95	—	—	**Cross Entropy Loss:** 1.156; **Loading Time:** 30.718s (Word2Vec)	Word2vec-LSTM, GLoVe-LSTM, and 25 models from prior literature (e.g., SVM, HMM)
He (2022)	**L-XGB** (Layer 1), **BiLA-ITD** (Layer 2)	98.35%	98.58%	97.89%	98.24%	0.9832	1.41%	**BiLA-ITD Precision:** 96.4%; **BiLA-ITD AUC:** 0.721	GNB, SVM, DTC, LSTM, LSTM+Attention, Bi-LSTM
He (2024)	**L-XGB** (Layer 1), **BiLA-ITD** (Layer 2)	98.35%	98.59%	97.89%	98.24%	0.721	1.41%	**Processing Time:** 56.58ms (for 4550 samples); **BiLA-ITD FPR:** 3.5%	GNB, SVM, DTC, Peng's NLP-ML model, and LSTM variations
Huang (2025)	**DenseAttDNN** (Dense Connection + Attention)	99.12%	0.9867	0.9754	0.9810	—	0.0125 (FAR)	RAM usage; Latency; Detection Time	SEL, DeepFed, SKP, DualNet-CV, SVM, LR
Jaiswal (2024)	**TSITD Models** (SMOTEENN, SMOTE-TL, ADASYN,	0.9994	0.95	0.9897	0.99	—	—	Balanced Acc (BAcc): 0.9949; Balanced Error Rate (BER):	Baseline models (unsampled data with identical 8 classifiers)

	SVMSMOTE, ENN + 8 Classifiers)							0.0051 [Table 4]; TOPSIS Ranking	
Janjua (2021)	**K-Means + Decision Tree (DT)**	99.96%	-	99.8%	—	0.995	—	Accuracy on raw dataset (72%); Confusion Matrix [Table 1, 2]	User centric framework (85%); BPNN (98%); Ensemble & DT (94%)
Kamatchi (2025)	**FAWO HMAC-SHA** (Bidirectional LSTM)	98.85%	91.52%	92.40%	—	—	0.0035	Throughput: 490 bps; Time period: 9s; Latency: ~3s; RAM: 30MB	Centralized ML, Federated Average (FedAvg), Federated Proximal, GRU-LSTM, TEE
Kong (2025)	**DPI-ITDD** (FMLP) and **DPI-ITDM** (XGB)	—	97.39%	95.81%	96.59%	—	0.0052	Runtime: 0.17s; Memory: 192 MB; Embedding: 20-D vector	DeepLog, DITD, ITDBERT, OITP, CATE
Kotb (2025)	**DS-IID** (Binary Deep Learning) Copula GAN	97.31%	0.79	0.86	0.82	0.99	0.019	Cohen’s kappa: 0.81; FAR: 0.14; TPR: 0.86; Trainable Params: 147,341	Topic Modeling, GRU, J48 DT, SVM, AE, VAE, ADASYN, DFS+PCA, Isolation Forest, CTGAN
Lavanya (2024)	**EBiGAN + DNN-PI** (Deep Neural Network with Bayesian Optimization)	**0.988**	**0.978**	**0.967**	—	—	**0.03**	FNR: 0.04; Training Loss: 0.04; Epochs: 200	ITDBERT, ADASYN+DNN, BiLSTM+HG+CAE, SPCAGAN, RNN, ResHybnet
Lavanya (2025)	**HOGPNN-ITD** (ABWGAN-EHI + L2-SP regularized Pretrained AGCN)	**0.989**	**0.979**	**0.973**	—	—	**0.02**	FNR: 0.03; Throughput: 5125 samples/sec; Inference time: 0.20 ms	CH-GLM, STGCN, TGCN-DA, MEWRGNN, EPRSO+XGBOOST, BRITD
Le & Zincir-Heywood (2021)	**Unsupervised Ensembles** (AE, IF, LODA, LOF with VOTE/AVG schemes)	—	—	**100%** (at 5% IB)	—	**0.981** (UAUC)	**0.1%** (at 0.1% IB)	Detection Delay: Scenario-based; UAUC (ACM2278): 0.9996	HMM, OneClass-SVM, LSTM Ensembles (Matterer), ITDBERT (Liu et al.), log2vec
Li et al. (2024)	**GMFITD** (Graph Meta-Learning Framework)	**86.97%**	—	—	—	—	—	Standard deviation: ±0.72; Training epochs:	LODA, LOF, RNN, LSTM, GRU, GCN, GAT, GCN-FA, GUIDE, CONAD,

								200; Update step: 10	Meta-GNN (MeGN), Matching Network, HGNN, Meta-GPS
Li et al. (2023)	**DD-GCN** (Dual-Domain Graph Convolutional Network)	**98.65%**	—	—	**93.04%**	—	—	**FLOPs** (Efficiency metric); **Latency** (Running time percent)	SVM, RF, LR, LODA, LOF, AE, LSTM, CNN, GCN, GCN-FA
Liu et al. (2025)	**Ripple2Detect** (BERT-based Semantic Similarity + Knowledge Graph)	**0.9640**	**0.9669**	**0.9609**	**0.99**	**ROC Fig. 3**	**0.033**	**HNSW Retrieval Time**: 3.63s; Training epochs: 200	IForest, AutoEncoder, DeepLog, Log2vec, IdtBert, TextCNN, Sentence-Bert, CosNet, SimCSE
Mehmood et al. (2023)	**LightGBM** (Primary), XGBoost, AdaBoost, Random Forest	**97%**	**97**	**83**	**0.97**	—	**0.11**	**Learning Rate**: 0.1; **Max Depth**: 10	SVM, Naïve Bayes, Gradient Boosting, RF, AdaBoost, XGBoost
Medvedev et al. (2025)	**SNN** (Siamese Neural Network) with **CNN** & Triplet Loss	**0.895**	—	—	—	—	—	**EER (Equal Error Rate)**: 0.11; **Embedding Vector**: 256-D	Mean threshold, Mean + std threshold, Confidence level threshold, CMU and KeyRecs benchmarks
Mehnaz (2021)	Fine-Grained Profiling (FGP) with Finite State Automata (FSA) and Unsupervised Clustering	**98.7%**	**96.5%**	**99.46%**	**95.92%**	-	**1.53%**	Overhead: 2%; FNR: 0.53%; Profile storage: <1 MB	Access Control, File Level Profiling (FLP), G
Mladenovic et al. (2024)	**XG-HARFO** (Metaheuristic Hybrid Adaptive Red Fox	**100%** (File task); **98.5%** (Email task)	**1.000** (File); **0.987** (Email)	**1.000**	**1.000** (File); **0.985** (Email)	-	-	Cohen's Kappa: 0.916; Objective function: Error Rate; Features: 250	Plain XGBoost/ AdaBoost, CatBoost, Random Forest, DNN, CNN, RFO, GA, PSO,

	Optimization + XGBoost)								ABC, FA, SCHO, ChOA
Nasir et al. (2021)	**LSTM-Autoencoder**	**90.60%**	**97%**	**92%**	**94%**	-	**9%**	Epochs: 200; Batch size: 64; Learning rate: 0.0001; Trainable parameters: 17,193	LSTM-CNN, Random Forest, LSTM-RNN, One Class SVM, Markov Chain, MSLSTM & CNN, GRU & Skip-gram
Nikiforova et al. (2024)	**K-means** refined with **Elbow method** and **Markov chains**,	—	—	—	—	—	—	**Suspiciousness coefficient** per session; Identifies additional malicious sessions for **158 users**	Manual role-based grouping; Traditional K-means focusing on action clusters
Pal et al. (2023)	**Ensemble** of **Stacked-LSTM** and **Stacked-GRU** with **Attention**,	—	—	**1.0** (v6.2)	—	**0.9968** (v5.2)	**0.0** (v6.2)	**EWRS** sampling technique; Input vector size: **282**	Yuan et al. (2018), Sharma et al. (2020), Huang et al. (2021); Meng et al. (2018)
Patel & Iyer (2025)	**SiaDNN** (Siamese CNN + DNN),	**91.373%**	—	**90.703%**	—	—	**7.300%**	**K-fold = 9**; FP Growth algorithm; Input vector size: **784**	PromptAD, Ensemble-based framework, QBDE, LSTM-Autoencoder, CNN-GRU, improved KNN
Nikiforova et al. (2024)	**K-means** refined with **Elbow method** and **Markov chains**,	—	—	—	—	—	—	**Suspiciousness coefficient** per session; Identifies additional malicious sessions for **158 users**	Manual role-based grouping; Traditional K-means focusing on action clusters

Pennada (2024)	Stacking Classifier, Voting Classifier, Random Forest (RF), Adaboost, Decision Tree (DT),	**99.2%**	**98.3%**	**98.3%**	**99%**	—	—	PCA applied; 830 features; SMOTE, ADASYN, and ENN sampling,	SVM (82.4%), XGBoost (92%), Isolation Forest (80%), Random Forest (98.1%)
Pennada (2025)	Hybrid Model (DAE/VAE + RF/XGBoost), Generative AI (DAE, VAE),	**94.1%**	**92.0%**	**91.2%**	**91.6%**	**0.96**	—	t-statistic: 5.2; p-value < 0.05; 16-D latent space extraction,	Vanilla ML/DL Models (87.2%), Generative AI Models (89.1%),
Perez-Miguel et al. (2025)	Not Specified (Dataset generation focus)	—	—	—	—	—	—	72,250 logs; 25 curated attributes; MITRE ATT&CK mapping,	Qualitative comparison with CERT r4.2, TWOS, LANL, and Enron
Qawasmeh & AlQahtani (2025)	XGBoost, Random Forest (RF), SVM, Logistic Regression (LR),	**1.00**	**1.00**	**1.00**	**1.00**	—	—	Detection Time: 0.014s; Classification Time: 0.071s (Logistic Regression)	Benchmarks from multiple studies (e.g., Alshuaibi et al., Shaver et al.)–
Randive et al. (2023)	Wavelet CNN (WCNN), DNN, MobileNet V2, ResNet-50, VGG-19,	**97.19%**	**95.00%**	**99.00%**	**97.00%**	**97.30%**	—	64×64 grayscale image representation; Haar wavelets; 25 epochs,,	LSTM-Autoencoder, Variational Autoencoders, IGT, MS-LSTM, GCN
Rauf et al. (2021)	**Random Forest**, **SVM**, **DBSCAN**, and **Z3 SMT Solver**	**98%**	—	—	—	—	—	**Policy Synthesis Time (≈0.15s)**; **Safety Verification Time (≈0.155s)**	**SIEM (Splunk)**; manual security analyst reporting time (≈15 mins)
Roy & Chen (2024)	**GraphCH** (GNN), **CH-GLM**, and **BiLSTM**	**97%**	**99%**	—	**99%**	**1.000**	**0.01%**	**False Negative Rate (0.05%)**	**metapath2vec**, **Log2vec**, **GraphSAGE**, **GAT**, **Logistic Regression**, **SVM**
Bin Sarhan & Altwaijry (2023)	**SVM**, **Neural Network (NN)**, **AdaBoost**, and **Random Forest**	**100%**	**1.00**	**1.00**	**1.00**	—	—	**Stratified 10-fold cross-validation**	**CNN**, **Autoencoder**, **DBN-OCSVM**, **DNN**, **GAN**, **Light Gradient Boosting**
Senevirathna et al. (2025)	**CNN-Random Forest Ensemble** (Cyber); **MobileNetV2 + LSTM** (Physical)	**98%** (Cyber Ensemble) ; **99.16%** (Physical)	**98%** (Cyber); **93%** (CNN-only)	**99%** (Cyber Class 0); **95%** (Physical)	**0.86** (CNN-only Malicious)	—	Not specified	**Processing speed:** 25-30 FPS; **Training:** 20 epochs	**Yi & Tian (2024)**, **Sharma et al. (2021)**, **Zhou et al. (2021)**, **Gavai et al. (2015)**, **Saaudi et al. (2019)**,,,,

Song et al. (2024)	**BRITD** (Stacked Bidirectional LSTM + FNN)	—	**0.8072**	**1.0000**	**0.8540**	**0.9730**	—	**Test time:** < 3s; **Training time:** 78.49s (12-h granularity)	**IF**, **OCSVM**, **FNN**, **BiLSTM**, **HMM**, **Tuor et al. (2017)**, **Dr et al. (2022)**, **Wu and Li (2021)**,,,,
Tabassum et al. (2024)	**Isolation Forest (IForest) + SVM** (Top Performer)	**99.21%**	**0.9823**	**99.75%**	**98.72%**	—	**0.68%** (calculated from 99.32% specificity)	**Kappa:** 0.6823; **Silhouette Score:** 0.63; **Dunn index:** 0.45	**Baseline (LOF) models** (SVM, Decision Tree, Random Forest),,,
Tian T et al. (2025)	**ITDSTS** (Transformer Encoder / Multi-head Attention),	**0.99**	**0.96**	**0.94**	**0.95**	—	—	**Dropout:** 0.2; **Epochs:** 200	LSTM (Sharma 2020), GCN (Hong 2023), CNN, SVM, Random Forest,
Tian Z et al. (2024)	**DSDLITD** (Attention-LSTM + Dempster-Shafer Fusion),	**95.47%**	**~0.95**	**95.79%**	—	—	**4.67%**	**CPU Usage:** ~80%; **Memory Usage:** ~560MB	GRNN, PNN, RBNN, KNN, SVM, Bayesian,
Villarreal-Vasquez et al. (2023)	**LADOHD** (Long Short-Term Memory - LSTM),	—	**53.04%**	**97.29%**	**0.85**	—	**0.38%**	**Vocabulary size:** 175 events; **BPTT:** 64	Enterprise EDR system, HMM, Lu's Method (2019),
Wall & Agrafiotis (2021)	**Bayesian Network** (Bayes-Ball algorithm),	—	—	—	—	**0.9912**	**"Very low"**	**Training time:** 11–41 seconds; **Inference:** Linear time	HMM, PCA, RNN, LSTM
Wang & El Saddik (2023)	**DistilledTrans**; BERT+FL; RoBERTa+FL; Transformer	**99.82%**	**100.00%**	**95.38%**	**96.55%**	**99.63%**	—	**Training time:** 10x faster than LSTM-AutoEncoder; **Epochs:** 20	One-Class SVM, HMM, Isolation Forest, Deeplog, LSTM-AutoEncoder, LSTM-CNN
Wang Zhi et al. (2024) FedITD	**FedITD** (XLNet + BitFit + TL); LoRA; Adapter; LLMs	**99.54%**	—	**96.94%**	**95.17%** (Macro)	**97.81%**	—	**Comm. Cost:** Reduced 98–99%; **Memory:** 0.04MB	Federated AutoEncoder, DeepMIT, DD-GCN, log2vec++, LSTM-RNN
Wang Jiarong et al. (2023) Deep Cluster	**Deep Clustering Network** (RNN/GRU Encoder-Decoder)	—	—	**99.84%**	—	**98%**	—	**Avg Recall:** 95.11%; **Embedding Dim:** 10-D	BAIT (SVM/NB), Isolation Forest, Scenario-Based (RF, Deep Autoencoder)
Wei Yichen et al. (2021)	**CPJOS** (Cascaded Autoencoders + BiLSTM + Hypergraph)	—	—	**0.925**	—	**93.2%**	**0.051**	**Purification:** K=5 AEs; **Epochs:** 50	One-class SVM, Unsupervised DNN, One-class AE, DAGMM, MAIDF
Wei Zhiyuan et al. (2024)	**E-Watcher** (Hybrid: LOF + Information Gain + Random Forest)	**98.48%**	**100%**	**98.48%**	**99.23%**	**1.00**	—	**Impact Ratio (IR):** ≥0.03 required; **Noise Resilience:** up to 10%	Gavai (2015), Aldairi (2019), Koutsouvelis (2020), Rastogi (2020), Rauf (2021), Le (2021)
Wen et al. (2023)	**SVD** and **Eigenvector Centrality**.	—	—	—	—	—	—	Identified **18/20** abnormality-related employees.	**ASEP** (Decision Tree + Sentiment).

Xiao Junchao et al. (2023)	**MEWRGNN** (R-GCN, GCN, CAN-GAT).	**99.18%**.	97.77%.	97.55%.	97.66%.	**0.996**.	—	**Detection Delay:** ~3 days for certain behaviors.	k-NN, Naive Bayes, Decision Trees, MLP, GCN.
Xiao Fengrui et al. (2025)	**SENTINEL** (ST-GNN: GCN+GRU, EGAT).	—	—	**98.0%** (TPR).	—	**0.980**.	—	**Training Time:** 488s (LANL dataset).	LODA, LOF, LSTM, AddGraph, Log2vec, LMTracker.
Xiao Haitao et al. (2024)	**CATE** (Convolutional Attention & Transformer).	—	95.93%.	96.42%.	96.18%.	—	**0.26%**.	**Optimal Sequence Length:** 320.	LR, KNN, DT, RF, GB, CNN, LSTM, Transformer.
Ye Xiaoyun et al. (2025)	**Personalized FL** (SqueezeNet + DeepInsight).	—	**99.95%** (Fed).	**99.96%** (Fed).	**0.9996** (Fed).	**0.99**.	—	**Model Size:** <1MB; **Local Epochs:** 5.	Random Forest, Isolation Forest, **FedAT**.
Yildirim & Anarim (2022)	**Ensemble Learning** (XGBoost/GBM).	—	—	—	—	96.47%.	—	**EER:** 7.46%; **Response Time:** <2ms.	Antal (2019), Chong (2019), Ahmed (2007).
Zhu et al. (2024)	**AUTH** (TL-AAE: TCN + LSTM + Adversarial).	—	—	—	—	0.9319.	—	**EER:** 0.1463; **Training Time:** 3.61 hours.	OCSVM, IF, DAGMM, VAE-LSTM, TCNAE, RCA.

The Studies were grouped into 3 categories as detailed in the SLR , the tables below detail these three groups.

3 Groups

Group A: The "Balanced Classification" Cluster (Focus on F1, Recall, Precision) (57 Studies)

Goal: To balance detection rates against false alarms in imbalanced data.

Paper ID	Algorithm	Accuracy	Precision	Recall (TPR)	F1-Score	Comparative Effectiveness / Insight
Adun (2023)	SVM, ANFIS	92%	-	93%	-	SVM outperformed ANFIS in raw accuracy (92% vs 91%).
Alabdulkareem (2022)	LSTM-GRU Ensemble	99.1%	98.6%	98.6%	99.1%	High consistency across all metrics indicates robust handling of class imbalance.
Almusawi (2024)	ML + Expert Policies	99%	100%	94%	97%	**Perfect Precision (100%)** proves expert rules effectively filter false positives.
AL-Mihqani (2022)	RF + KNN	96%	74.2%	84%	95%	**Precision Drop:** High F1 but low Precision (74%) suggests high False Alarm rate.

Paper ID	Algorithm	Accuracy	Precision	Recall (TPR)	F1-Score	Comparative Effectiveness / Insight
ALmihqani (2021)	ADASYN + DNN	96%	-	-	95%	ADASYN sampling stabilized the F1-score to 95%.
Al-Shehari (2021)	DT/RF + SMOTE	-	99%	100%	99%	SMOTE yielded perfect recall (100%), but likely overfitted on synthetic CERT data.
Al-Shehari (2024 CNN)	CNN + ADASYN	-	-	-	-	Reported 96% AUC (see Group B); F1 not primary.
Ali et al. (2025)	BERT Ensemble	96%	-	-	-	Metrics focused on text classification accuracy; lacks granular F1.
Alhammadi (2021)	CNN/RF (EEG)	97%	-	-	-	2D CNN statistically outperformed 1D CNN (p<0.001).
Alshehri (2022)	Rel-RNN	-	99%	67%	0.80	**Recall Gap:** High precision (99%) but poor recall (67%) shows missed threats.
Amuda (2022)	CNN-GRU	97.4%	99.9%	97.4%	-	Hybrid model achieved near-perfect precision (99.9%).
Anju (2024)	CNN-BiGRU	92.5%	98%	95%	96%	Balanced profile; attention mechanism maintained high recall.
Anakath (2022)	DBN	99%	-	-	98%	F-Measure of 98% confirms DBN effectiveness on mouse dynamics.
Asha S (2023)	OCSVM + Sampling	82.5%	64.9%	100%	78.7%	**Trade-off:** 100% Recall (caught everything) but poor Precision (64.9%).
Eshmawi (2026)	KNN/SVM/RF	99.9%	100%	100%	100%	**Suspicious Perfection:** 100% across all metrics suggests overfitting on CERT.
Feng (2025)	RF (Multi-Granularity)	99.9%	99.9%	99.9%	99.9%	Similarly suspicious perfect scores on synthetic data.
Ferraro (2025)	LLaMA-3.1 (LLM)	95.8%	35.8%	100%	52.7%	**The "Paranoid" Model:** Caught 100% of threats but had massive False Positives (35% Prec).
Gayathri (2025 Adv)	SNN-MLP	-	1.00	0.973	0.916	Maintained high metrics even under adversarial attack scenarios.
Gayathri (2025 Cloud)	JNN + LSTM-AE	97%	95%	96%	97%	JNN architecture provided balanced performance for cloud logs.
Gayathri (2024 Hybrid)	SPCAGAN + BNN	-	89.5%	99.0%	91.9%	GAN augmentation boosted Recall to 99%.
Gupta (2024)	Federated Learning	96.7%	96.8%	96.7%	96.7%	Proved Federated Learning matches centralized performance (loss < 0.07).
Haq (2022)	XGBoost + Word2Vec	0.92	0.92	1.00	0.95	**Perfect Recall (1.0)** using NLP features on Enron emails.
He (2022)	Bi-LSTM Attention	98.3%	98.6%	97.9%	98.2%	Attention mechanisms improved F1 to 0.982 compared to baselines.
He (2024)	BiLA-ITD	98.3%	98.6%	97.9%	98.2%	Consistent performance; noted FPR of 1.41%.
Huang (2025)	DenseAttDNN	99.1%	98.7%	97.5%	98.1%	Attention layer kept False Acceptance Rate (FAR) low (0.0125).
Jaiswal (2024)	Ensemble (SMOTE)	99.9%	95%	98.9%	99%	**Benchmark Study:** Used TOPSIS ranking to prove Ensemble superiority.
Janjua (2021)	K-Means + DT	99.9%	-	99.8%	-	High accuracy (99.9%) on unlabeled email clusters.
Kamatchi (2025)	Bi-LSTM (Fed)	98.8%	91.5%	92.4%	-	Good recall (92%) despite the constraints of Federated Learning.

Paper ID	Algorithm	Accuracy	Precision	Recall (TPR)	F1-Score	Comparative Effectiveness / Insight
Kong (2025)	FMLP / XGB	-	97.4%	95.8%	96.6%	Filter-enhanced MLP achieved high F1 (96.6%) with low memory.
Kotb (2025)	Copula GAN	97.3%	79%	86%	82%	**Realistic:** Lower F1 (0.82) reflects the difficulty of identifying AI-generated threats.
Lavanya (2024)	EBiGAN + DNN	98.8%	97.8%	96.7%	-	GAN generation improved DNN Precision to 97.8%.
Lavanya (2025)	HOGPNN-ITD (GNN)	98.9%	97.9%	97.3%	-	Graph features outperformed standard DNNs (see row above).
Mehmood (2023)	LightGBM	97%	97%	83%	97%	LightGBM had high precision but dropped significant Recall (83%).
Mehnaz (2021)	FGP (FSA)	98.7%	96.5%	99.5%	95.9%	Finite State Automata achieved massive recall (99.5%) on file logs.
Mladenovic (2024)	XG-HARFO	100%	1.00	1.00	1.00	**Overfitting Risk:** Perfect scores on optimized feature sets.
Nasir (2021)	LSTM-AE	90.6%	97%	92%	94%	Autoencoder approach balanced precision (97%) and recall (92%).
Pal (2023)	Stacked LSTM	-	-	1.00	-	**Perfect Recall (1.0)** on CERT v6.2 using Attention.
Patel & Iyer (2025)	SiaDNN	91.3%	-	90.7%	-	Siamese networks struggled with Precision (not reported), only Recall.
Pennada (2024)	Stacking Ensemble	99.2%	98.3%	98.3%	99%	Stacking multiple classifiers achieved 99% F1.
Pennada (2025)	Hybrid VAE	94.1%	92.0%	91.2%	91.6%	Generative features (VAE) achieved solid F1 (91.6%) on difficult data.
Qawasmeh (2025)	XGBoost	100%	1.00	1.00	1.00	**Overfitting:** 100% scores on synthetic data are likely unrealistic.
Randive (2023)	Wavelet CNN	97.2%	95%	99%	97%	Image-based CNN achieved very high Recall (99%).
Roy & Chen (2024)	GraphCH (GNN)	97%	99%	-	99%	Graph approach yielded 99% F1, proving structural modeling works.
Bin Sarhan (2023)	SVM/NN	100%	1.00	1.00	1.00	Another case of suspicious 100% metrics on synthetic data.
Senevirathna (2025)	CNN + RF	98%	98%	99%	-	Cyber-Physical fusion achieved 99% Recall on Class 0.
Song (2024)	Bi-LSTM	-	80.7%	100%	85.4%	**Recall Bias:** Perfect recall (100%) but low Precision (80%).
Tabassum (2024)	IForest + SVM	99.2%	98.2%	99.7%	98.7%	Hybrid model on Real Hospital data; 99.7% recall is impressive.
Tian T (2025)	Transformer	0.99	0.96	0.94	0.95	Transformer achieved 0.95 F1, balancing the metrics well.
Tian Z (2024)	Attention-LSTM	95.5%	0.95	95.8%	-	Dempster-Shafer fusion maintained ~95% across all metrics.
Villarreal (2023)	LADOHD (LSTM)	-	53%	97%	0.85	**Precision Issue:** Low precision (53%) on Real EDR data.
Wang & El Saddik	DistilledTrans	99.8%	100%	95.4%	96.5%	**Best in Class:** High F1 (96.5%) with 100% Precision.
Wang Zhi (2024)	FedITD (LLM)	99.5%	-	96.9%	95.2%	LLM achieved 95% F1 in a Federated setting.
Wang Jiarong (2023)	Deep Cluster	-	-	99.8%	-	Unsupervised clustering reached 99.8% Recall.
Wei Yichen (2021)	Autoencoder	-	-	0.925	-	Data "purification" strategy reached 92.5% Recall.
Wei Zhiyuan (2024)	LOF + RF	98.5%	100%	98.5%	99.2%	Personalized profiling achieved near-perfect F1 (99.2%).
Xiao Haitao (2024)	CATE	-	95.9%	96.4%	96.2%	Transformer-CNN hybrid balanced metrics at ~96%.
Xiao Fengrui (2025)	SENTINEL (GNN)	-	-	98.0%	-	Graph model focused on Recall (98%).
Xiao Junchao (2023)	MEWRGNN	99.2%	97.7%	97.5%	97.6%	Relational Graph achieved consistent 97%+ across all metrics.
Ye Xiaoyun (2025)	SqueezeNet (FL)	-	99.9%	99.9%	0.999	**Image-based FL:** 99.9% scores suggest SqueezeNet is highly effective.

Note: We purposely did not include all the results of the studies. We took a sample that represent the above group to highlight the tendency to focus on Focus on F1, Recall, Precision instead of Accuracy.

Group B: The "Anomaly & Ranking" Cluster (Focus on AUC, EER, FPR) 21 studies

Goal: To measure the quality of anomaly scoring and ranking, independent of thresholds.

Paper ID	Algorithm	AUC / ROC	FPR (False Positive Rate)	Other Metrics (EER / Detection Rate)
Ahmed (2025)	Random Cut Forest	-	-	TPR: 0.95; Detection Time: 1-20 mins.
Al-Shehari (2023)	Isolation Forest	0.99	-	Standard anomaly detection metric.
Al-Shehari (2024)	DBLOF	0.99	-	Detection Rate: 98%.
Amiri-Zarandi (2023)	Autoencoder (Fed)	0.93	0.20	Used Investigation Budget (20%) metric.
Cai X (2024)	GCN + LSTM	0.96	0.0865	Retrieval Threshold ϵ: 0.5.
Dong J (2025)	Diffusion (DDCC)	0.9823	-	**EER Focus:** Used Equal Error Rate for diffusion thresholding.
Gonzales (2025)	C3P (Pattern Mining)	0.96	-	Contextual scoring AUC.
Hafizu Rhman (2022)	NARX	-	21.13%	**High FPR:** 21% FPR indicates statistical model limitation.
Le & Zincir (2021)	Unsupervised Ensemble	0.981	0.1% (at 0.1% IB)	**Metric Innovation:** UAUC (Utility AUC) to measure budget vs. detection.
Li et al. (2024)	Meta-Learning	-	-	Focused on Few-Shot stability (Std Dev ±0.72).
Li et al. (2023)	DD-GCN	93.04%	-	Focused on Graph FLOPs (Efficiency).
Liu et al. (2025)	Knowledge Graph	0.99	0.033	ROC curve analysis (Fig 3 in paper).
Medvedev (2025)	Siamese NN	-	-	**EER: 0.11**. Standard for biometric authentication.
Nikiforova (2024)	K-Means	-	-	Qualitative: "Suspiciousness coefficient".
Peccatiello (2023)	IForest / LOF	-	-	Stream-based anomaly detection (metrics unclear).
Perez-Miguel (2025)	Dataset Focus	-	-	Qualitative comparison of datasets (SPEDIA vs CERT).
Rauf (2021)	SMT Solver	-	-	Verification Safety (Logic-based, not statistical).
Wall (2021)	Bayesian Network	0.9912	"Very low"	Bayesian inference probability.
Wen (2023)	SVD (Eigenvector)	-	-	Identified 18/20 malicious employees (Qualitative).
Yildirim (2022)	Mouse Dynamics	96.47%	-	**EER: 7.46%** for biometric authentication.
Zhu (2024)	Adversarial AE	0.9319	-	**EER: 0.1463** for reconstruction error.

Group C: The "Operational Efficiency" Cluster (Focus on Time, Cost, Complexity) 18 Studies

Goal: To assess practical deployment viability (Real-time vs. Offline).

Paper ID	**Algorithm**	**Computational Cost / Efficiency Metrics**	**Comparison / Insight**
Ahmadi (2025)	RF / GBM	**Response Time:** "Seconds"	Suitable for Zero Trust (Real-time).
Anju (2024)	Stacked CNN	**Training:** 0.2s; **Pred:** 1.5–2.3s	Fast training, slightly slower inference.
Gayathri B (2025)	Jordan NN	**Execution:** 0.8298s	Sub-second execution for cloud streams.
Gonzales (2025)	C3P	**Complexity:** $O(N \cdot L^2)$	Quadratic complexity hurts scalability.
Gupta (2024)	FedMUP	**Time:** 367s; **Memory:** 18.9MB	Low memory footprint for Federated Learning.
Haq (2022)	Word2Vec	**Loading Time:** 30.7s	NLP embedding loading is a bottleneck.
He (2024)	BiLA-ITD	**Proc Time:** 56.58ms	Ultra-fast (ms) processing for 4550 samples.
Huang (2025)	DenseAttDNN	**Latency:** Reported (unspecified)	Focus on RAM vs Latency trade-off.
Kamatchi (2025)	Bi-LSTM (IoT)	**RAM:** 30MB; **Latency:** 3s	**IoT Optimized:** Fits on edge devices.
Kong (2025)	FMLP	**Runtime:** 0.17s; **Mem:** 192MB	Efficient embedding vector (20-D).
Lavanya (2025)	GNN	**Throughput:** 5125 samples/sec	High throughput for Graph model.
Qawasmeh (2025)	XGBoost	**Detection:** 0.014s	Extremely fast tabular classification.
Tian Z (2024)	Attention-LSTM	**CPU:** 80%; **Mem:** 560MB	**High Resource:** Heavy load for LSTM attention.
Wang & El Saddik	DistilledTrans	**Training:** 10x faster than LSTM	Distillation massively sped up training.
Wang Zhi (2024)	FedITD (LLM)	**Comm Cost:** Reduced 99%	Federated Learning focused on bandwidth.
Xiao Junchao (2023)	MEWRGNN	**Delay:** ~3 Days (found later to be wrong) it is Hours not days	**Critical Flaw:** Graph construction is too slow.
Xiao Fengrui (2025)	SENTINEL	**Training:** 488s	Moderate training time for GNN.
Ye Xiaoyun (2025)	SqueezeNet	**Size:** <1MB	**Ultra-light:** SqueezeNet fits on any device.

Note: The duplicate entries across these three tables are due to the fact that many of the 82 unique studies contribute to more than one thematic area. Studies are listed in Table 1 if they report standard performance (e.g., Accuracy, Precision, and F1-Score), Table 2 if they focus on detection quality (e.g., AUC and False Positive Rates), and Table 3 if they provide technical details on computational efficiency (e.g., speed or memory usage). This multi-mapping approach ensures each table contains all relevant evidence from the 82-paper corpus, even if a study is cited in multiple categories.