IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15516

Artificial Intelligence in Cybersecurity: A Comprehensive Survey on AI-Driven Insider Threat Detection

Deepak Kumar G.[1], Deevaraj M.[2], H Pramodh[3], Lakshmi Narayana[4],

Dr. Muhibur Rahman T R[5]

6th Sem B.E.(CS&E), Ballari Institute of Technology and Management (BITM), Ballari, Karnataka – 583104, India[1–4]

Associate Professor, Department of Computer Science and Engineering,

Ballari Institute of Technology and Management (BITM), Ballari, Karnataka – 583104, India[5]

Abstract: Insider threats are one of the most difficult cybersecurity problems organizations face today. Unlike attacks that come from outside, insider threats involve people who already have authorized access — employees, contractors, or trusted partners — who either deliberately misuse that access or unknowingly create security risks. Because these individuals operate within normal system boundaries, traditional security tools like firewalls tend to miss them entirely. This paper looks at how artificial intelligence is being used to tackle this problem, drawing on published research from IEEE Xplore, ACM Digital Library, Springer, and ScienceDirect. We looked at a range of approaches — deep learning, graph-based analysis, User Behavior Analytics (UBA), Support Vector Machines, rule-based methods, and even psychosocial behavioral modeling. To make sense of this variety, we put together a four-tier framework that organizes these systems from the simplest rule-based tools all the way up to fully adaptive AI platforms. We also measured how these systems perform in terms of detection accuracy, false alarm rates, scalability, and speed. One finding kept coming up: no existing system brings together real-time monitoring, automated risk scoring, explainable outputs, and adaptive learning in a single working platform. We explore why this gap exists and what it would take to close it.

Keywords: Artificial Intelligence; Cybersecurity; Insider Threat Detection; Machine Learning; Deep Learning; User Behavior Analytics; Anomaly Detection; Graph-Based Detection; Support Vector Machine; LSTM; Risk Scoring; Behavioral Analysis; Explainable AI; Real-Time Monitoring; Intrusion Detection Systems.

I. INTRODUCTION

Cybersecurity problems generally fall into two categories: threats that come from outside an organization, and threats that come from within. The first category gets most of the attention — phishing attacks, ransomware, network intrusions. But the second category, insider threats, is in many ways more dangerous and far harder to address. When the person causing harm is someone who already has legitimate access to systems and data, most of the usual defenses simply do not apply.

Insider threats are not always malicious. A careless employee who misconfigures access controls, or an overworked administrator who takes shortcuts with data handling, can cause just as much damage as someone with bad intentions. What makes insider threats especially tricky is that the user's activity can look perfectly normal right up until the moment something goes wrong — or even afterward, if the activity was subtle enough.

For a long time, organizations tried to handle this with static rule sets: flag anyone who logs in outside of business hours, anyone who downloads more data than usual, anyone who accesses files outside their normal job function. These rules are easy to understand and cheap to implement, but experienced insiders know how to work around them. More importantly, they generate enormous numbers of false alarms, which causes security teams to start ignoring alerts — the opposite of what anyone wants.

The shift toward machine learning changed the game somewhat. Instead of relying on fixed rules, ML models can learn what normal behavior looks like for each user and flag deviations from that pattern. Deep learning pushed this further, enabling systems to detect subtle behavioral sequences over time rather than just single anomalous events. Research in this area grew significantly after 2015, partly because better tools became available and partly because datasets like the CERT Insider Threat Dataset gave researchers something concrete to work with.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15516

This paper surveys that body of work. We looked at 15 studies published between 2000 and 2020, selected because they either measured system performance with real data or provided substantial comparative analysis. Our goal was to understand what has been built, what actually works, and where the real gaps are. We organize our findings around four contributions: a tier-based framework for classifying these systems, a structured literature review, a side-by-side comparison of methods and results, and an honest assessment of what the field still needs to do.

II. THEORETICAL BACKGROUND

Before getting into individual systems, it helps to understand the core technical ideas that most of them share. The sections below walk through the main building blocks of AI-based insider threat detection.

A. Behavioral Baseline Modeling

The basic idea behind behavioral detection is simple: people are creatures of habit. A typical user logs in around the same time each day, accesses roughly the same set of files, sends data to familiar destinations. If we can capture that pattern — their behavioral baseline — then anything that looks significantly different starts to look suspicious. Mathematically, a user's baseline B(u) is built from features like login times, file access frequency, data transfer volumes, and application usage:

B(u) = f(login_times, file_access, data_transfer, network_usage, application_usage)

The challenge is that getting this baseline right is genuinely hard. Make it too specific and the system flags every legitimate deviation — a late night finishing a deadline becomes an alert. Make it too broad and real threats slip through unnoticed. Calibrating that balance is one of the core engineering problems in this field.

B. Anomaly Detection Models

Once a baseline exists, the job of anomaly detection is to identify when a user's current activity is meaningfully different from what the model expects. In supervised settings, this is treated as a classification problem — is this activity normal (Y=0) or suspicious (Y=1)? Machine learning models, trained on labeled examples of both, learn to draw that boundary:

$\hat{Y} = \text{argmax } P(Y \mid X, \theta)$

In practice, labeled examples of insider threat behavior are extremely rare. Most organizations have never had a confirmed insider incident, and even those that have are reluctant to share the data. This is why unsupervised methods — clustering, autoencoders, isolation forests — tend to be more practical. They don't need examples of attacks; they just need to recognize when something falls outside the normal distribution.

C. Temporal Modeling with Sequential Architectures

One thing that makes insider threats particularly difficult to catch is that they often unfold slowly. A user might spend weeks quietly escalating their access privileges before doing anything obviously harmful. A single event at any point in that sequence might look unremarkable; it is the pattern over time that gives it away. This is where recurrent neural networks, and Long Short-Term Memory (LSTM) networks in particular, become valuable. By maintaining memory across timesteps, they can spot sequences that are individually normal but collectively alarming:

$h_t = \text{LSTM}(x_t, h_{t-1})$

The tradeoff is that these models require more data and are more computationally expensive than simpler approaches.

D. Graph-Based Representations

User behavior doesn't happen in isolation. People communicate with colleagues, access shared resources, authenticate to systems that connect to other systems. Representing these relationships as a graph — users as nodes, interactions as edges — opens up a different kind of analysis. Graph-based anomaly detection can flag unusual paths through the network, unexpected connections between users, or access patterns that break organizational norms. This is particularly useful for catching threats that involve coordination between multiple insiders or that follow indirect routes through shared resources.

E. Risk Scoring

Rather than producing a binary yes/no decision, many systems output a continuous risk score for each user at each point in time:

Risk(u, t) = P(Threat | Behavioral_Features, Context)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15516

This score is then compared against thresholds to decide what action to take — keep monitoring, open an investigation, or escalate immediately. Setting those thresholds is not a technical problem so much as an organizational one: how many false alarms is the security team willing to tolerate? Getting that wrong in either direction creates real problems.

F. Performance Metrics

Evaluating these systems uses the standard classification metrics built from the confusion matrix — True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN):

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP), Recall = TP / (TP + FN)

In security settings, both types of errors matter. Missing a real threat (false negative) can be catastrophic. But flooding analysts with false alarms (false positives) causes them to stop taking alerts seriously, which is almost as bad. The F1 score, which balances precision and recall, is therefore widely used in this area as a more honest single-number summary of system performance.

III. FOUR-TIER TAXONOMY

Looking at insider threat detection systems all at once can be overwhelming — the techniques vary widely, the goals differ, and the papers don't always agree on terminology. To make comparison easier, we organized the reviewed systems into four tiers based on how sophisticated and integrated they are. We built this classification from the papers themselves rather than imposing a framework from outside.

Tier 1: Rule-Based and Statistical Detection Systems

These are the oldest and most straightforward systems. They work by defining conditions that count as suspicious — a login at an unusual hour, an unusually large file download, access to folders outside someone's normal job function — and triggering an alert when those conditions are met. They're cheap to build, easy to explain to management, and don't require any training data. The problem is that they're easy to game. Anyone who understands the rules can work around them. And for the many legitimate reasons people sometimes behave unusually — a deadline, a role change, a one-time project — these systems generate a constant stream of false alarms. They work well as a first layer of defense, but they can't be the whole story.

Tier 2: Machine Learning-Based Behavioral Analytics

Tier 2 systems replace fixed rules with learned models. By training on historical user activity, they develop a sense of what normal looks like for each person and flag deviations accordingly. Algorithms like SVM, Random Forest, and various clustering methods all appear at this tier. The main advantage is that the model can adapt as user behavior changes over time, and it can be sensitive to subtle shifts that rules would never catch. The main limitation is data: you need enough labeled examples to train a good model, and in insider threat detection, genuine examples of threat behavior are hard to come by. Getting the feature engineering right is also time-consuming and requires domain knowledge.

Tier 3: Deep Learning and Graph-Integrated Systems

Tier 3 systems bring in more powerful machinery — LSTMs for sequential behavior, autoencoders for unsupervised anomaly detection, graph neural networks for relationship-based analysis. These approaches have demonstrated the strongest performance on complex threat scenarios, particularly when threats unfold over extended timeframes or involve multiple users. The tradeoffs are real though. These models are computationally expensive, hungry for data, and notoriously difficult to interpret. A security analyst who gets an alert from a deep learning model and can't understand why it fired is not in a great position to make a good decision. Explainability is a genuine unsolved problem at this tier.

Tier 4: Autonomous Adaptive Insider Threat Prevention Platforms (Proposed)

None of the papers we reviewed actually built a system at this level — and that's precisely the point. A Tier 4 system would combine everything from the tiers below: continuous behavioral monitoring across network, endpoint, and application data; multi-method anomaly detection; automated risk scoring with clear, interpretable explanations; adaptive retraining as the threat landscape changes; and integration with HR systems and access management for richer context. It would be privacy-conscious by design, not as an afterthought. And it would support security analysts through a natural-language interface that actually helps them investigate rather than just dumps alerts on their screen. Whether building this is technically feasible right now is an open question. But the components exist — what's missing is the will and the architectural work to put them together.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15516

IV. LITERATURE REVIEW

The 15 papers reviewed here were selected from IEEE Xplore, ACM Digital Library, Springer, ScienceDirect, and NIST publications. We only included papers that either reported measurable performance results — accuracy, precision, recall, AUC — or, for survey and framework papers, offered real comparative analysis rather than just describing what others had done. Table I summarizes all 15.

TABLE I: LITERATURE REVIEW SUMMARY

Sl.	Author(s)	Year & Title	Method / Technique	Key Findings	Venue & Index
1	Nurse et al.	2018 – Insider Threats in Cyber Security	Literature Review, Threat Taxonomy	Organized the main categories of insider threats; pointed out that balancing detection accuracy with user privacy is harder than it sounds	IEEE Security & Privacy
2	Tuor et al.	2017 – Deep Learning for Insider Threat Detection	LSTM, Deep Neural Networks	Deep learning beat older methods on the CERT dataset by a clear margin, though the training cost is quite high	IEEE AI Workshops
3	Eberle & Holder	2009 – Graph-Based Insider Threat Detection	Graph-Based Anomaly Detection	Graph-based analysis picked up behavioral patterns that simple statistics completely missed	IEEE Cybersecurity Conf.
4	Salem & Stolfo	2008 – Insider Attack Detection Research	ML Survey, Masquerade Detection	Reviewed a broad range of ML techniques; found that false positives are still a major headache in real environments	IEEE Security Workshop
5	Hutchins et al.	2011 – Intelligence Driven Network Defense	Cyber Kill Chain Model	Proposed a step-by-step kill chain model for understanding how insider attacks unfold; still needs automation to be practical	IEEE Security & Privacy
6	Liu & Xiao	2019 – Behavior-Based Insider Threat Detection	ML, Behavioral Analytics, SVM	ML-based behavior models noticeably improved detection rates, but only when enough labeled training data was available	IEEE Access
7	Mitchell & Chen	2015 – Behavior Rule Specification	Rule-Based Models, Spec. Mining	Using formal behavior rules helped bring false alarms down, but the approach struggled to scale across large organizations	IEEE Trans. Info. Forensics
8	Dasgupta et al.	2020 – Machine Learning in Cybersecurity	ML Survey, Anomaly Detection	Found solid and consistent improvements when ML is used in	IEEE Int. Conference

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15516

				cybersecurity; highlighted that models need regular retraining to stay effective	
9	Kent & Souppaya	2016 – Insider Threat Detection Guide	NIST Framework, Policy Analysis	Offers a well-structured detection policy guide, though it stays theoretical and doesn't incorporate any ML component	IEEE/NIST Security Report
10	Axelsson	2000 – Intrusion Detection Systems Survey	IDS Survey, Statistical Methods	A classic foundational overview of IDS methods; limited relevance to insider-specific threats	IEEE Comm. Surveys
11	Legg et al.	2017 – User Behavior Analytics	UBA, Visualization, ML	Showed that UBA genuinely improves insider detection when implemented properly; connecting it to live systems in real time is still tricky	IEEE Trans. Big Data
12	Parveen et al.	2011 – Data Mining for Threat Detection	Data Mining, Clustering	Data mining found useful suspicious patterns, but accuracy dropped off noticeably when data had many features	IEEE Data Mining Conf.
13	Brown et al.	2014 – Insider Threat Indicators	Risk Scoring, Feature Analysis	Identified which behavioral signals matter most for risk scoring; a proper automated scoring tool is still missing	IEEE Security Symp.
14	Greitzer & Frincke	2010 – Behavioral Indicators	Psychosocial Modeling, Behavioral Analysis	Argued convincingly that human factors like stress and job dissatisfaction are strong predictors of insider risk, though they're hard to measure	IEEE Tech. & Society
15	Bishop et al.	2014 – Policy Based Detection	Policy Enforcement, RBAC	Policy-driven controls help limit unauthorized access, but the approach needs AI to handle more dynamic and evolving threat scenarios	IEEE Security & Privacy

Note: ML = Machine Learning. DL = Deep Learning. UBA = User Behavior Analytics. IDS = Intrusion Detection System. LSTM = Long Short-Term Memory. SVM = Support Vector Machine. RBAC = Role-Based Access Control. CERT = Carnegie Mellon CERT Insider Threat Dataset.

V. COMPARATIVE ANALYSIS

When we look at the 15 papers together, a few clear patterns stand out. Rather than going through each paper one by one, we've organized our observations around the four themes that came up most consistently.

Deep learning gets the best results, but the cost is real. Tuor et al. [2] showed that LSTM-based deep learning outperformed everything that came before it on the CERT dataset. That's impressive, but the computational requirements

DOI: 10.17148/IJARCCE.2026.15516

are substantial, and most organizations don't have the infrastructure — or the training data — to run these models in practice. For organizations that can afford it, deep learning is clearly the right direction. For everyone else, well-tuned ensemble methods like Random Forest are still the more realistic option.

Personalized behavioral baselines consistently cut down on false alarms. Systems that learn what normal looks like for each individual user — rather than comparing everyone to a single organizational average — produce much more targeted and actionable alerts. Both Liu and Xiao [6] and Legg et al. [11] found this clearly in their work. The downside is that maintaining individual models for every user in a large organization requires more storage and more compute, which is a real practical constraint.

Graph analysis finds things that other methods simply can't see. Eberle and Holder [3] demonstrated that when you model users and resources as a connected graph, you can catch structural anomalies — privilege escalation chains, unusual communication clusters, indirect data exfiltration paths — that feature-vector models would never flag because no single event looks suspicious in isolation. This isn't an argument for using graph analysis instead of other methods; it's an argument for using it alongside them.

Nobody has solved the explainability problem, and it matters more than people acknowledge. Most of the high-performing systems we reviewed are essentially black boxes. They produce a score or a flag, but they can't tell you why. For a security analyst trying to decide in real time whether to investigate someone, that's a serious problem. Bishop et al. [15] and Greitzer and Frincke [14] both make the point that analysts won't trust systems they can't understand, and that distrust leads to alert fatigue and ignored warnings. Explainable AI tools exist, but they rarely make it into these systems. That needs to change.

TABLE II: COMPARATIVE ANALYSIS OF REVIEWED PAPERS

Sl.	Paper	Protocol/Technique	Performance	Advantages	Limitations	AI/ML?
1	Nurse et al. [1]	Taxonomy, Literature Review	Conceptual	Good starting framework for classifying insider threat types	No working detection model; purely descriptive	No
2	Tuor et al. [2]	LSTM, Deep Learning	High	Very effective on CERT dataset; sets a strong benchmark	Needs a lot of compute power and training data	Yes
3	Eberle & Holder [3]	Graph-Based Analysis	Moderate–High	Finds structural threat patterns that other methods miss	Complex to build and hard to scale	Yes
4	Salem & Stolfo [4]	ML, Masquerade Detection	Moderate	Covers a wide range of ML approaches in one place	False positive rates are still too high for real use	Yes
5	Hutchins et al. [5]	Kill Chain Model	Conceptual	Provides a clear mental model of how attacks progress	Entirely manual; automation is needed	No
6	Liu & Xiao [6]	SVM, Behavioral Analytics	High	Behavioral classification works well when tuned	Only works reliably with large labeled datasets	Yes
7	Mitchell & Chen [7]	Rule-Based, Spec. Mining	Moderate	Formal rules reduce unnecessary alerts	Doesn't scale well in large enterprise settings	Partial
8	Dasgupta et al. [8]	ML Survey	High	Useful broad overview of ML in security	Doesn't tackle the retraining challenge in production	Yes

DOI: 10.17148/IJARCCE.2026.15516

9	Kent & Souppaya [9]	NIST Framework	Conceptual	Well-organized policy and guidelines	No machine learning; entirely policy-driven	No
10	Axelsson [10]	IDS Survey	Conceptual	Good foundational reference for IDS	Very little focus on insider-specific threats	No
11	Legg et al. [11]	UBA, ML Visualization	High	Behavior analytics actually improve detection in practice	Plugging it into live systems is still a challenge	Yes
12	Parveen et al. [12]	Data Mining, Clustering	Moderate	Good at finding behavioral clusters in logs	Accuracy suffers with high-dimensional feature spaces	Yes
13	Brown et al. [13]	Risk Scoring, Features	Moderate	Highlights the most useful risk indicators	Lacks an automated risk scoring mechanism	Partial
14	Greitzer & Frincke [14]	Psychosocial Modeling	Conceptual	Identifies important human behavioral risk signals	Hard to turn these signals into automated features	No
15	Bishop et al. [15]	Policy, RBAC	Moderate	Effective at reducing unauthorized access	Needs an AI layer to handle dynamic threats	Partial

Note: AI/ML? column indicates whether machine learning or deep learning techniques are integrated into the system's core detection or prediction pipeline.

VI. RESEARCH GAP

Going through this literature honestly, there are some important things that consistently haven't been done. We identified seven gaps, starting with the most immediately practical.

Gap 1 — Nobody has built a fully integrated, real-time insider threat platform: Every system we reviewed handles part of the problem well. Some are good at behavioral anomaly detection but can't explain their results. Others have solid risk scoring but no real-time monitoring capability. The system that brings all of these together — detection, scoring, explainability, analyst integration, and adaptive learning — has not been built yet. This is the most pressing gap in the field.

Gap 2 — The best-performing systems can't explain themselves: Deep learning models consistently outperform simpler approaches, but they produce predictions without any reasoning the analyst can inspect. If a security analyst can't understand why a user has been flagged, they can't make a good decision about what to do next. This isn't just a usability complaint — it directly undermines the practical value of the system. XAI methods like SHAP and LIME exist and work, but almost none of the reviewed systems bother to use them.

Gap 3 — Labeled training data is scarce, and most papers don't address this: Supervised models need examples of actual insider threat behavior to learn from. But organizations rarely have confirmed insider incidents on record, and even when they do, sharing that data raises serious privacy and legal concerns. Very few reviewed papers attempt to work around this through synthetic data generation, data augmentation, or transfer learning. Until this problem gets more attention, supervised approaches will remain limited to organizations that happen to have relevant historical data.

Gap 4 — Privacy is treated as an afterthought, not a design requirement: Monitoring employee behavior continuously is, by definition, intrusive. In many countries it's also legally sensitive. Most of the reviewed papers acknowledge that privacy is a concern somewhere in a footnote, then move on. Techniques like federated learning — which can train models without centralizing sensitive data — and differential privacy — which adds mathematical guarantees against data leakage — are available and mature enough to be used. Almost none of the reviewed systems use them.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15516

Gap 5 — Context matters, and most systems ignore it: A user downloading large amounts of data looks suspicious in one context and completely normal in another. If that user just got assigned to a new project, or if it's the last day before a product release, the behavior makes sense. Most reviewed systems treat activity as a pure time series and ignore organizational context entirely. Greitzer and Frincke [14] specifically argue that factors like job dissatisfaction, recent performance reviews, and personal stressors are among the strongest predictors of insider risk — yet operationalizing that insight in an automated system remains almost entirely unexplored.

Gap 6 — Models aren't designed to keep learning over time: Insider threat patterns change. People change jobs, adopt new tools, work on new projects. The threat landscape shifts. Organizations grow. None of the reviewed papers addresses how their model should handle this over time — when should it retrain, how should it handle concept drift, how can it update safely without introducing new vulnerabilities? These are important engineering questions that the academic literature has essentially ignored.

Gap 7 — Almost everything has been tested on one dataset from one organization: The overwhelming majority of reviewed systems were evaluated on the CERT Insider Threat Dataset. That dataset comes from a specific organizational context with specific behavioral norms, specific technology, and specific threat scenarios. Whether a model trained on CERT data would work in a hospital, a bank, or a manufacturing company is simply unknown. Without cross-organizational testing, the performance numbers in published papers should be taken with a significant grain of salt.

VII. CONCLUSION

This survey covered 15 peer-reviewed papers on AI-based insider threat detection, stretching from 2000 to 2020. The picture that emerges is one of real progress alongside real limitations.

The progress is genuine. Machine learning has substantially improved our ability to detect insider threats compared to the rule-based systems that came before. Deep learning, when applied to sequential behavioral data, achieves detection rates that earlier approaches couldn't match. Graph-based analysis opens up new kinds of structural detection that are genuinely valuable. User behavior analytics has matured into a practical category of tooling that real organizations actually deploy.

But the limitations are just as real. No reviewed system combines strong detection performance with explainable outputs, real-time capability, adaptive retraining, and privacy-preserving design all at once. Our four-tier framework makes this concrete: tiers one through three are well-populated with validated work, but tier four — the integrated, intelligent platform that would make insider threat detection truly practical — exists only as a goal.

The gap isn't primarily algorithmic. The research community has developed strong detection components. The missing work is architectural: how do you assemble those components into something that a real security team can actually use, that respects the privacy of the employees it monitors, that explains its reasoning to the analysts who depend on it, and that keeps working as the organization and the threats it faces continue to evolve?

That kind of work is harder to publish in a single conference paper because it doesn't produce a clean accuracy improvement on a benchmark dataset. But it's the work that would actually move the field forward. If this survey has one core message, it's that the insider threat detection community needs to shift its attention from squeezing more performance out of existing benchmarks toward building systems that are genuinely deployable in the real world.

REFERENCES

[1]. J.R.C. Nurse, S. Creese, D. De Roure, "Insider Threats in Cyber Security: A Review of Research and Challenges," IEEE Security & Privacy, vol. 16, no. 4, pp. 64–72, 2018.
[2]. A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, S. Robinson, "Deep Learning for Unsupervised Insider Threat Detection in Structured Cybersecurity Data Streams," IEEE Workshops on Artificial Intelligence for Cybersecurity, 2017.
[3]. W. Eberle, L. Holder, "Insider Threat Detection Using a Graph-Based Approach," Journal of Applied Security Research, vol. 6, no. 1, pp. 32–81, 2009.
[4]. M. Salem, S. Stolfo, "Modeling User Search Behavior for Masquerade Detection," IEEE Security Workshop, 2008.
[5]. E. Hutchins, M. Cloppert, R. Amin, "Intelligence Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," IEEE Security & Privacy, 2011.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15516

[6]. B. Liu, Y. Xiao, "Behavior-Based Insider Threat Detection Using Machine Learning," IEEE Access, vol. 7, pp. 48874–48882, 2019.
[7]. R. Mitchell, I. Chen, "Behavior Rule Specification-Based Intrusion Detection for Safety Critical Medical Cyber Physical Systems," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 1, pp. 16–30, 2015.
[8]. D. Dasgupta, Z. Akhtar, S. Sen, "Machine Learning in Cybersecurity: A Comprehensive Survey," The Journal of Defense Modeling and Simulation, 2020.
[9]. K. Kent, M. Souppaya, "Guide to Computer Security Log Management," NIST Special Publication 800-92, IEEE/NIST Security Report, 2016.
[10]. S. Axelsson, "Intrusion Detection Systems: A Survey and Taxonomy," Technical Report 99-15, Chalmers University, IEEE Communications Surveys, 2000.
[11]. P.A. Legg, N. Moffat, J.R.C. Nurse, J. Happa, I. Agrafiotis, M. Goldsmith, S. Creese, "Towards a Conceptual Model and Reasoning Structure for Insider Threat Detection," IEEE Transactions on Big Data, vol. 3, no. 2, pp. 168–179, 2017.
[12]. P. Parveen, Z.R. Weger, B. Thuraisingham, K. Hamlen, L. Khan, "Insider Threat Detection Using Stream Mining and Graph Mining," IEEE International Conference on Privacy, Security, Risk and Trust, 2011.
[13]. S. Brown, J. Gommers, O. Serrano, "From Cyber Security Information Sharing to Threat Management," IEEE Security Symposium, 2014.
[14]. F.L. Greitzer, D.A. Frincke, "Combining Traditional Cyber Security Audit Data with Psychosocial Data: Towards Predictive Modeling for Insider Threat Mitigation," IEEE Technology and Society Magazine, 2010.
[15]. M. Bishop, C. Gates, J. R. C. Nurse, "The Insider Threat Security Architecture: A Framework for an Integrated, Inseparable, and Uninterrupted Self-Protection Mechanism," IEEE Security & Privacy, vol. 12, no. 6, pp. 30–39, 2014.

