Insider Threat Detection Using Windows File-Share Events: An Unsupervised ML Approach in a Controlled Zero Trust Lab

Course: CSDP 710 – Machine Learning
Semester: Spring 2026
Institution: University of Maryland Eastern Shore

1. Motivation

Windows Active Directory domains continuously generate file-access telemetry, Event 4656 (local file access) and Event 5145 (network share access), capturing who accessed what, from which machine, at what time, and whether access was granted. In most organizations this data is never analyzed.

Insider threats are hard to detect because attackers blend into normal activity; the signal is in the behavioral pattern, not the individual event. Zero Trust Architecture (ZTA) [1] mandates "never trust, always verify", but that principle only holds if behavioral signals are actually being modeled. Prior work has shown ML can detect insider threats [2,4] and authorization anomalies in ZTA settings [3], but these studies use logon events or cloud telemetry.

This research fills that gap by building an unsupervised ML pipeline directly from 4656 and 5145 events in a live, controlled on-premise AD lab.

2. Problem Statement

Given a stream of Windows file-share access events from domain users in a Zero Trust environment, can unsupervised machine learning detect anomalous user sessions without any pre-labeled attack examples?

The dataset is small by design in Phase 1 – 58 sessions across 6 users over 23 days – which rules out supervised learning but is sufficient to establish a behavioral baseline. Events are unlabeled and normal behavior varies across users, hours, and days, making simple threshold rules unreliable.

This paper covers Phase 1 only: building and validating the pipeline on baseline data. Phase 2 will introduce simulated attack sessions for labeled evaluation. Phase 3 will deploy the model in a live real-time sliding window.

3. Research Questions

- **RQ1:** Can unsupervised ML detect anomalous sessions from 4656/5145 events without labeled attacks? *(Phase 1 – partially addressed)*
- **RQ2:** Which anomaly category is best detected by which model among Isolation Forest, LOF, and One-Class SVM? *(Phase 2)*
- **RQ3:** Which behavioral features contribute most to anomaly score differentiation? *(Phase 1 – SHAP analysis)*
- **RQ4:** Is the pipeline deployable in real time with acceptable false positive rate and latency? *(Phase 3)*

4. Literature Review

4.1. How Prior Research Has Tackled This Problem

Le and Zincir-Heywood [2] developed an insider threat detection model using machine learning on user behavioral data, showing that session-level features capturing access patterns can effectively distinguish malicious from normal activity without requiring full attack logs upfront. Their work is directly relevant to our session aggregation strategy.

Parkhomenko et al. [3] applied Zero Trust principles to detect authorization anomalies in cloud environments, demonstrating that behavioral deviations from established access patterns are detectable using unsupervised methods. This reinforces the ZTA framing of our research.

Anderson et al. [4] examined AI-powered insider threat detection specifically within Zero Trust ecosystems, identifying that combining behavioral baselines with real-time scoring is a practical path toward continuous verification. Their findings support both our model selection and the Phase 3 real-time deployment goal.

The Microsoft Sentinel study [5] combined Isolation Forest with SHAP explanations on logon event streams, demonstrating that anomaly detection outputs can be made interpretable to security analysts. This directly informs the SHAP component of our pipeline.

4.2. Limitations of Current Works

Despite these contributions, four gaps remain. First, all prior work operated on logon events, cloud telemetry, or synthetic datasets – no study has applied unsupervised ML specifically to Windows file-access events (4656/5145). Second, no study compared Isolation Forest, LOF, and One-Class SVM on the same feature set across multiple anomaly categories in a ZTA lab. Third, SHAP explanations have not been applied at the file-access layer. Fourth, none of the existing works deployed and measured a detection pipeline in real time on a live enterprise event stream.

5. Planned Contributions

- **1** – First anomaly detection pipeline from 4656/5145 events in a live ZTA lab, addressing the file-layer gap (RQ1, RQ3).
- **2** – Three-model comparison (IF, LOF, OC-SVM) on the same feature set; per-category evaluation in Phase 2 (RQ2).
- **3** – SHAP analysis at the file-access layer, extending [5] to a richer event type (RQ3).
- **4** – Live real-time sliding-window deployment with measured FPR and latency in Phase 3 (RQ4).
- **5** – Fully scripted, reproducible lab on Windows Server 2022 AD with VMware clients.

6. System Model and Data Collection

The lab consists of a Windows Server 2022 Active Directory domain controller for the `CSDP` domain and two to three VMware Windows 10 client machines joined to the domain. A Python collector on the domain controller subscribes to Events 4656 and 5145, serializes each event to JSON, and appends to a rolling log file. The dashboard is exposed through a Cloudflare tunnel. The detection pipeline runs five stages: collect, normalize, feature engineering, model scoring, and SHAP explanation.

Phase 1 data was collected over 23 days (April 12 to May 5, 2026) with no simulated attacks,

producing 18,572 events across 58 sessions and 6 users (*don, alice, arya, bob, jack, tom*). Two source IPs were observed (`192.168.21.135` and `192.168.21.132`) across four shared drives (H, F, G, E). Of all events, 12,954 were successful and 5,618 were failures. The event distribution by hour and access mask breakdown are shown in Figure 1.

7. Proposed Solution

Raw events are grouped by `subject_logon_id`, the Windows per-login session identifier, and aggregated into 12 behavioral features per session (Table 1). The `co_event_flag` is specific to this study and captures whether Events 4656 and 5145 co-occur within the same session, a behavioral fingerprint that logon-only studies cannot derive.

Three unsupervised models are trained on the full Phase 1 dataset with a contamination rate of 3%, treating all sessions as approximately normal since no attacks were present. **Isolation Forest** is the primary model for RQ1: it isolates anomalous sessions by random partitioning, requiring no labeled examples. **Local Outlier Factor** (k = 20) detects sessions in unusually sparse local neighborhoods, which is effective for gradual behavioral drift. **One-Class SVM** (RBF kernel) learns a tight boundary around the normal cluster and is well suited to hard permission violations.

Table 1: Session-level features engineered from Events 4656 and 5145

Feature	Description
`access_hour`	Median hour of session activity
`is_offhours`	1 if hour less than 7 or at least 20
`is_weekend`	1 if Saturday or Sunday
`total_events`	Total event count in session
`failure_count`	Count of audit failure events
`failure_rate`	Failure count divided by total events
`unique_files`	Distinct file objects accessed
`unique_drives`	Distinct drive letters accessed
`has_delete_attempt`	1 if DELETE appears in any access list
`has_write_dac`	1 if WRITE_DAC or WRITE_OWNER present
`unique_source_ips`	Distinct source IPs within the session
`co_event_flag`	1 if both 4656 and 5145 occur in session

8. Experimental Setup

The dataset consists of 18,572 events aggregated into 58 sessions across 6 users over 23 days with no simulated attacks. Events were loaded from JSON into a pandas DataFrame, timestamps parsed, and sessions grouped by `subject_logon_id`. All features were scaled using `StandardScaler` before model input. Isolation Forest used `n_estimators` = 200 and random state = 42. LOF used k = 20. OC-SVM used an RBF kernel with gamma set to scale. Contamination and nu were both set to 0.03 across all models. SHAP values were computed using `TreeExplainer` on the trained Isolation Forest, with mean absolute SHAP values averaged over all 58 sessions. All experiments ran on colab notebook.

9. Experiments

9.1. Feature Analysis

The session-level feature correlation matrix is shown in Figure 2. The strongest relationship is between `total_events` and `failure_count` (r = 0.71), which is expected since longer sessions accu-

mulate more failures. `has_delete_attempt` and `has_write_dac` show moderate correlation (r = 0.56), consistent with privilege-escalation sessions involving both. Notably, `unique_source_ips` is zero-variance across all Phase 1 sessions as all activity originated from the same two lab machines. This feature will become active in Phase 2 when new machines are introduced for simulated attack sessions.

9.2. Detection Results

Table 2 summarizes the output of all three models. Isolation Forest and LOF both flagged exactly 2 sessions at the 3% contamination threshold, agreeing on the same two sessions. One-Class SVM flagged 11 sessions, reflecting its tighter boundary behavior on a small, low-variance dataset. No session was flagged by all three models.

Table 2: Phase 1 anomaly detection results across 58 sessions

Model	Sessions flagged	Percentage
Isolation Forest	2	3.4%
LOF	2	3.4%
One-Class SVM	11	19.0%
At least 2 models agree	2	3.4%
All 3 models agree	0	0.0%

The two sessions agreed upon by IF and LOF are behaviorally distinct. Session `0x5ed7004` (user *tom*) shows access at 01:00 with a 100% failure rate across 66 events, consistent with a failed brute-force attempt or a misconfigured scheduled task. Session `0x2beaf09` (user *don*) accessed 16 unique files at 20:00 with a 49.4% failure rate and included a delete attempt, which looks like off-hours data enumeration. The IF anomaly score distribution and the event volume versus failure rate scatter are shown in Figure 3. Model agreement and the IF versus LOF score comparison are shown in Figure 4.

The OC-SVM result is not a pipeline failure. On a dataset this small with low feature variance particularly with `unique_source_ips` being zero, OC-SVM draws an overly tight boundary. This will be re-evaluated in Phase 2 when the dataset is larger and more behaviorally diverse.

9.3. SHAP Feature Importance

The global SHAP importance ranking is shown in Figure 5. `has_write_dac` is the top feature with a mean absolute SHAP value of 0.293, followed by `total_events` (0.211) and `failure_count` (0.188). `access_hour` ranks fourth at 0.186, confirming that temporal signals are meaningful even in a dataset containing no deliberate attacks. `unique_source_ips` contributes zero as expected.

The per-session SHAP dot plot for the most anomalous session is shown in Figure 6. For session `0x5ed7004` (*tom*), both `failure_rate` and `access_hour` push the anomaly score strongly in the anomalous direction. The combination of a 1am access time with a 100% failure rate produces a clear, human-interpretable explanation for the flag – exactly the kind of output that makes the pipeline useful in a real security operations context.

10. Future Work

Phase 2: Simulated attacks and labeled evaluation. Four attack categories will be scripted on the VMware clients to produce labeled ground truth: unusual access (sensitive file access at 2am), permission escalation (repeated WRITE_DAC and WRITE_OWNER attempts), behav-

ioral deviation (login from a new IP followed immediately by file access), and policy violation (access during an unapproved time window). Models will be trained on Phase 1 normal data only and evaluated against Phase 2 labeled sessions, producing per-category precision, recall, and F1 scores (RQ2). This will also activate `unique_source_ips` for the first time.

Phase 3: Real-time deployment. The trained Isolation Forest will be integrated into the existing Python dashboard using a sliding window over the most recent sessions. Events will be scored as they arrive. False positive rate and processing latency will be measured against targets of below 5% FPR and under 2 seconds per batch (RQ4).

11. Conclusion

This paper presents Phase 1 of an ongoing research effort to apply unsupervised machine learning to Windows file-share telemetry for insider threat detection in Zero Trust environments. Working from 18,572 real events collected over 23 days, the pipeline engineers 12 per-session behavioral features and evaluates three unsupervised models. Isolation Forest and LOF agree on two anomalous sessions in the baseline data. SHAP analysis identifies `has_write_dac`, `total_events`, and `access_hour` as the dominant anomaly drivers.

What Phase 1 establishes is that the pipeline works end-to-end on real data and the feature set captures meaningful behavioral variation. What it does not yet provide is per-category precision and recall or real-time performance metrics, those are the goals of Phases 2 and 3. This report is the documented baseline for that continued work.

References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," *NIST Special Publication 800-207*, National Institute of Standards and Technology, 2020.

[2] D. C. Le and A. N. Zincir-Heywood, "Machine learning based insider threat modelling and detection," *Proc. IEEE/IFIP Network Operations and Management Symposium Workshops*, pp. 1–6, 2019.

[3] I. Parkhomenko, L. Myrutenko, R. Ohiievych, and M. Savonik, "Using Zero Trust principles for detecting authorization attacks in cloud environments," *Proc. IEEE International Conference on Computer Science and Information Technologies*, 2023.

[4] M. Anderson, S. Reynolds, E. Carter, O. Mitchell, and M. Yusuff, "AI-powered insider threat detection in Zero Trust ecosystems," *Technical Report*, January 2025.

[5] Microsoft, "Detecting anomalous user behavior with Microsoft Sentinel and machine learning," *Microsoft Tech Community Blog*, 2023.

Figures

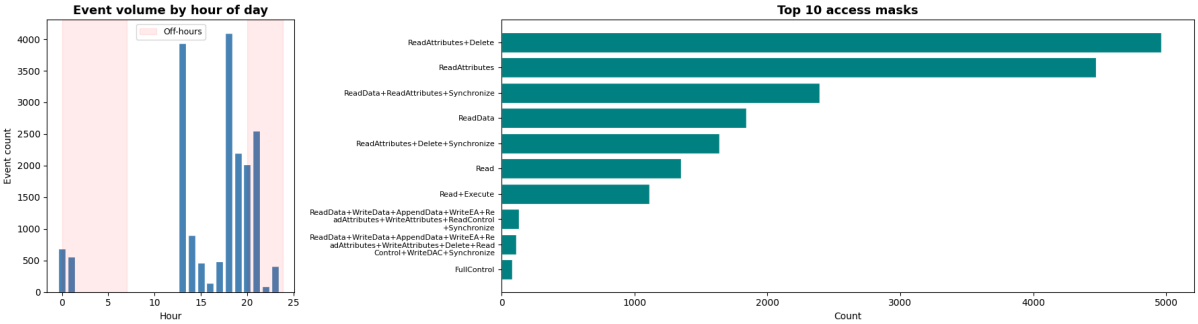


Figure 1: Event volume by hour of day (left) and top-10 access masks (right). Activity concentrates between 13:00 and 22:00. `ReadAttributes+Delete` is the most frequent mask, consistent with normal file-browser enumeration.

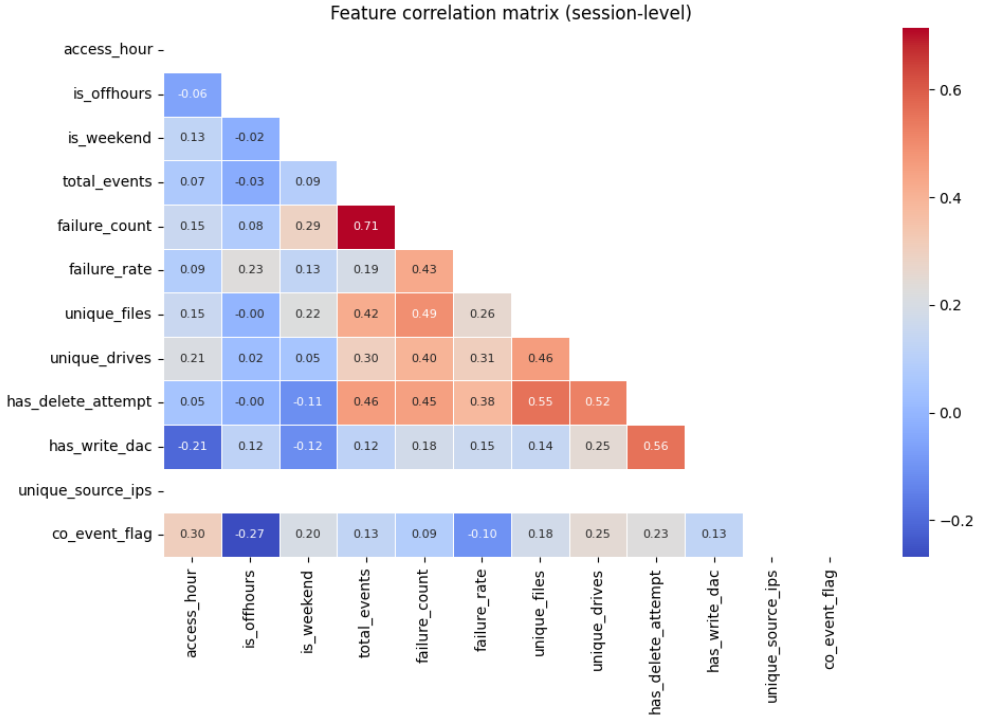


Figure 2: Session-level feature correlation matrix. The strongest pair is `total_events` and `failure_count` ($r = 0.71$). `unique_source_ips` is zero-variance in Phase 1.

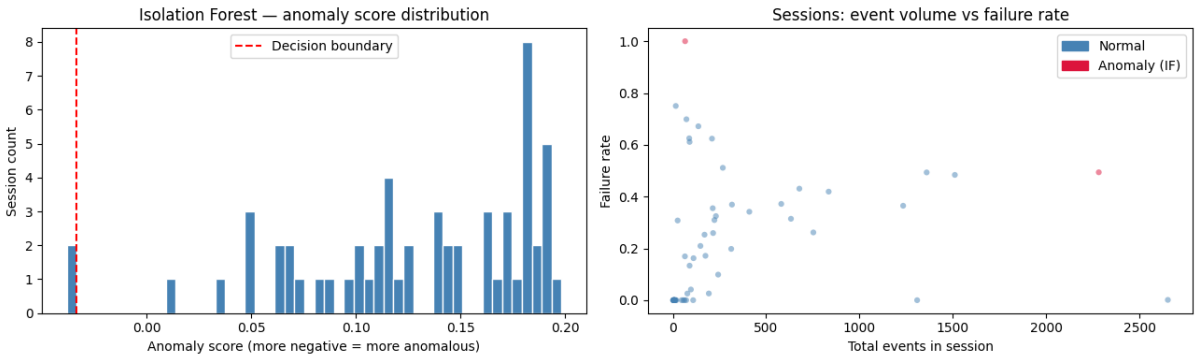


Figure 3: Isolation Forest results. Left: anomaly score distribution with the decision boundary marked. Right: session event volume versus failure rate; red points are flagged anomalies.

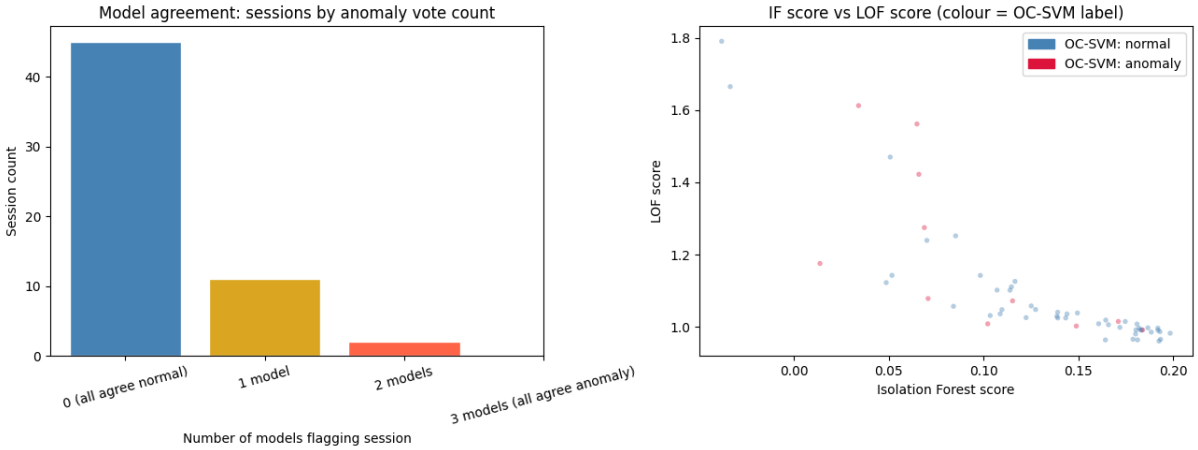


Figure 4: Model agreement. Left: number of models flagging each session. Right: Isolation Forest score versus LOF score, colored by One-Class SVM label.

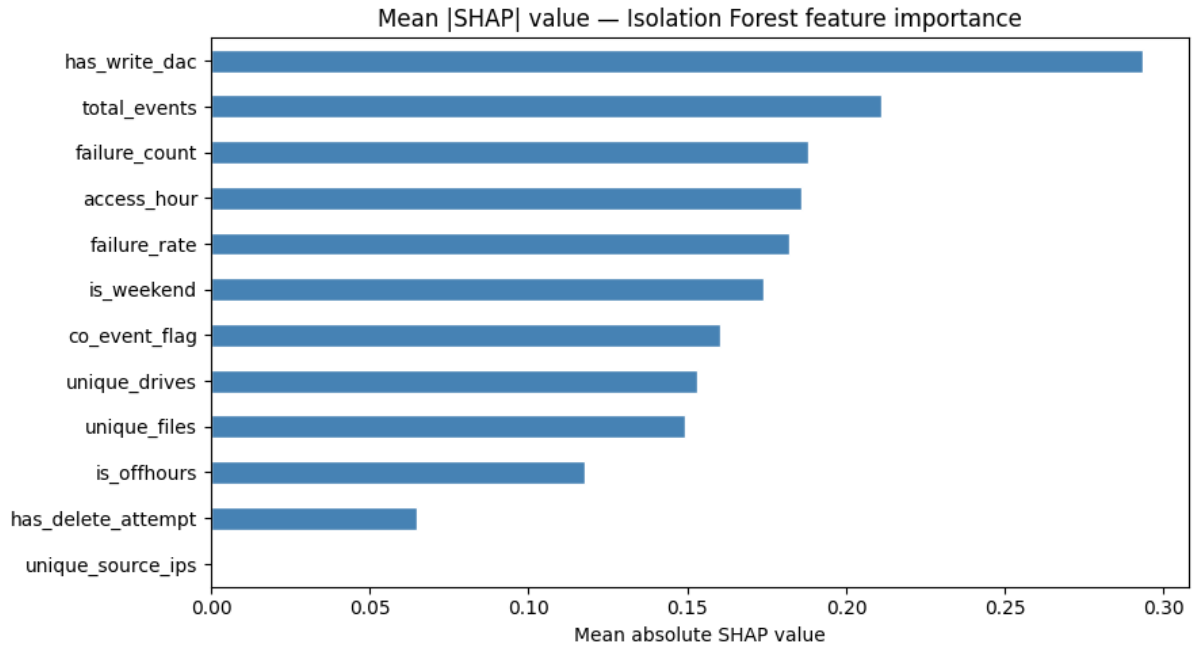


Figure 5: Mean absolute SHAP values for Isolation Forest (global feature importance). `has_write_dac` is the dominant feature. `unique_source_ips` contributes zero due to zero-variance in Phase 1.

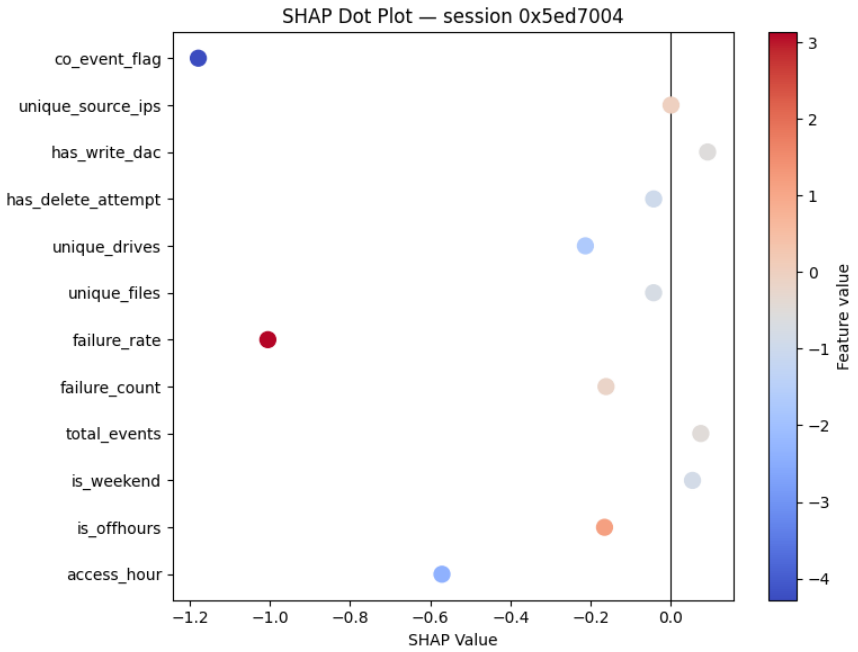


Figure 6: SHAP dot plot for the most anomalous session (user *tom*, session `0x5ed7004`). `failure_rate` and `access_hour` are the primary drivers of the anomaly score.