Unsupervised Behavioural Profiling for Insider Threat Detection Using Time-Series and Anomaly Detection Techniques

Student Name: Bipin Rimal
Course: MSc Data Science and Computational Intelligence
Module: Computing Individual Research Project (STW7048CEM)
Supervisor: Manoj Shrestha
Date: November 10, 2025

Abstract

This report details the research, design, and evaluation of a data-centric, semi-supervised machine learning framework for detecting insider threats from enterprise log data. Addressing the limitations of signature-based systems in identifying authorized but malicious activity, the project implements a robust pipeline using the CMU-CERT dataset (r1–r3.1), processing over 38 million log events to compare three anomaly detection architectures: Isolation Forest (Statistical), Deep Clustering (K-Means Autoencoder), and LSTM Autoencoder (Sequential). The research navigates the specific challenges of "Class Imbalance" in massive datasets and tests the efficacy of sliding-window sequence generation on sparse log data. Contrary to the initial hypothesis that fixed windows would dilute threat signals, the experimental results from the V2 Ensemble indicated that the LSTM Autoencoder was the dominant driver of detection, receiving an optimal ensemble weight of 0.8 compared to 0.1 for static models. This finding challenges the assumption that sequential granularity is inefficient for sparse logs, suggesting that temporal reconstruction error provides a higher precision signal than static volumetric counts. The final framework achieved a detection rate (Recall) of 66% (identifying 2 of 3 confirmed insiders) in a strictly aligned test set. However, the failure to detect a third "Living-off-the-Land" actor exposes a "Metadata Visibility Horizon," where low-magnitude malicious deviations remain statistically indistinguishable from benign workflow noise. Operationally, the research concludes that effective insider threat detection requires prioritizing sequential modelling and cohort-based standardization (Z-scores) to mitigate the false positives inherent in high-variance user behaviour.

Keywords

Insider Threat Detection, Anomaly Detection, Behavioural Profiling, Feature Engineering, Methodological Bias, Unsupervised Learning, LSTM, Isolation Forest, Accuracy Paradox, Ingestion Bias, Sliding Window Fallacy, Contextual Z-Scores, Data-centric AI, Cybersecurity

Table of Contents

Table of Figures

Tables

Introduction

The "Insider" Problem: A Crisis of Trust

In the domain of enterprise cybersecurity, the most mature and widely deployed defences are oriented outward, designed to protect the network perimeter from external adversaries. This "castle-and-moat" strategy assumes that threats reside outside the firewall and that users inside the perimeter are inherently trustworthy (Pfleeger & Pfleeger, 2006). However, one of the most insidious and damaging classes of threat—the Insider Threat—originates from within this trusted boundary.

The CERT Division of the Software Engineering Institute (SEI) defines an insider as "a current or former employee, contractor, or other business partner who has or had authorized access to an organization's network, system, or data" (Cappelli et al., 2012, p. 5). These actors are uniquely dangerous not because of their technical sophistication, but because they subvert the foundational trust model of the entire network architecture. Unlike external hackers who must exploit vulnerabilities to gain entry, insiders possess legitimate credentials, understand internal security policies, and know precisely where critical data assets are located

This privilege allows them to bypass traditional defences like firewalls, Intrusion Prevention Systems (IPS), and signature-based antivirus tools, which are structurally designed to stop intruders, not trusted colleagues. The impact of this vector is disproportionate to its frequency; the 2022 Ponemon Institute report on the *Cost of Insider Threats* indicates that the average annual cost to an organization has risen to $16.2 million, a 44% increase over the last two years (Ponemon Institute, 2022). The motivations driving these actors are varied and complex, ranging from financial gain (corporate espionage) and data theft (selling customer lists) to revenge (sabotage) or simple, costly negligence. This behaviour is often conceptualized through the Fraud Triangle framework, which posits that pressure, opportunity, and rationalization must converge for an insider act to occur (Cressey, 1953).

The Theoretical Failure of Traditional Defences

To understand the necessity of a machine learning approach, one must first analyse the theoretical failure of legacy systems. Traditional Security Information and Event Management (SIEM) systems and Intrusion Detection Systems (IDS) are built on a deterministic, signature-based paradigm. This approach operates on deductive reasoning: it applies a set of predefined rules (signatures) to incoming data to determine maliciousness (Sommer & Paxson, 2010).

This approach is highly effective against "Known Unknowns," such as malware with a known file hash or traffic explicitly violating a policy (e.g., "Block all traffic to this known malicious IP"). However, it is theoretically incapable of detecting an insider who abuses their legitimate access. From a rule-based perspective, an insider exfiltrating data is not violating any syntactic policy; they are simply accessing files they are permitted to see. The intent is malicious, but the behaviour is technically legitimate.

This highlights the core limitation identified in this research: traditional systems are context-blind. A deterministic rule might block a user from accessing a restricted finance server, but it lacks the stateful memory to understand that a finance employee accessing that same server at 3:00 AM on a Sunday and querying 10,000 records—when their historical average is 50—is a high-risk anomaly. The system perceives the "what" (Access Granted) but cannot process the "how," "when," or "how much" in a contextual way. This renders multi-million-dollar defence stacks blind to "Living-off-the-Land" (LotL) techniques, where attackers use legitimate, pre-installed tools (like PowerShell or WMI) to conduct attacks without triggering malware signatures (Symantec, 2019).

The Shift to Unsupervised Behavioural Analytics

In response to this "Context Gap," the cybersecurity field has shifted toward User and Entity Behaviour Analytics (UEBA). This represents a paradigm shift from deterministic rules to probabilistic, unsupervised machine learning—effectively moving from deductive to inductive reasoning (Gartner, 2021).

Instead of relying on predefined signatures of "bad" behaviour, UEBA aims to build a high-fidelity, dynamic profile of "normal" behaviour for every user and device on the network. This is achieved by ingesting and analysing massive volumes of heterogenous log data (logon, file access, http, email, device insertion) over time to establish a baseline. Once this multi-dimensional baseline is established, the system's task is to identify anomalies—statistically significant deviations from an entity's own established patterns (Chandola et al., 2009).

This project designs, builds, and evaluates such a framework. The implication of this approach is a move from finding "Known Unknowns" (a new virus) to finding "Unknown Unknowns" (a novel attack method by a trusted employee). However, this power introduces a new, critical challenge that forms the basis of this thesis: the Base Rate Fallacy and the problem of False Positives. In a dataset where 99.9% of activities are benign, distinguishing malicious anomalies from benign anomalies (the "weird-but-normal" problem) becomes the primary barrier to operational viability (Axelsson, 2000).

Aim

The aim of this research is to design, implement, and critically evaluate an end-to-end unsupervised machine learning framework capable of detecting insider threats by profiling user behaviour from enterprise log data. Unlike commercial "black box" solutions, this research aims to deconstruct the "Contextual Entropy" of user behaviour, explicitly comparing how different architectural approaches (Statistical, Clustering, and Sequential) handle the trade-off between detection sensitivity and the suppression of benign anomalies.

Objectives

To achieve this aim, the research is structured around the following objectives:

1. **Analyse and Select Models:** Critically analyse existing literature to select a complementary set of unsupervised models—specifically statistical, clustering, and sequential architectures—to form a multi-faceted detection engine.
2. **Engineer a Robust Data Pipeline:** Design and implement a scalable, multi-stage data pipeline to process, unify, and accurately label the multi-terabyte, multi-dataset CMU-CERT corpus (Glasser & Lindauer, 2013), overcoming significant data integrity and methodological challenges identified during experimentation.
3. **Establish a Baseline:** Develop and evaluate a "baseline" detection model (on the r2 dataset) to quantify its performance, specifically analysing the theoretical limitations and the practical trade-off between Recall and Precision.
4. **Develop an Advanced Solution:** Design and implement an advanced model (on the full r1-r5.2 datasets) using enriched, relative features (Z-scores) to address the precision limitations and "weird-but-normal" problem identified in the baseline.

Justification

The problem this research addresses is the failure of traditional security tools to detect insider threats. The proposed solution is a data-centric framework that uses unsupervised machine learning to model the "normal" behaviour of users and detect anomalous deviations. This approach is justified because, unlike rule-based systems, it does not rely on *a priori* knowledge of attack patterns and can, therefore, detect novel or "zero-day" insider activities.

However, the academic literature often presents this as a straightforward task, frequently glossing over the operational reality of false positives by only reporting high AUC-ROC scores. This project's core justification is to move beyond a simplistic implementation and rigorously investigate the practical viability of such a system. Our initial baseline experiment (Chapter 4) proves this gap: while our models found the insider (100% Recall), they were operationally unusable due to an overwhelming number of false positives (near-zero Precision). This finding proves that the true challenge is not mere model selection. This project's central justification is its data-centric, iterative response to this failure. We prove that a viable solution requires:

1. A robust data pipeline that can generalize across multiple, diverse datasets (our V2 pipeline).
2. Advanced feature engineering (using relative Z-scores) to provide models with the context needed to distinguish "maliciously weird" from "benignly weird" behaviour.

This moves the project from a simple "model bake-off" to a deeper investigation of the methodological and data-centric challenges at the heart of computational intelligence in cybersecurity.

Research Questions

This thesis seeks to answer the following questions:

Technical Question:

To what extent can unsupervised models (statistical, clustering, and sequential) discriminate between benign and malicious behaviour, and what are the theoretical limitations of simple, count-based features in separating "maliciously anomalous" from "benignly anomalous" activity?

Methodological Question:

How do the biases inherent in data aggregation, experimental methodology (e.g., data splitting vs. a "train-on-normal" approach), and algorithmic thresholding create a fundamental trade-off between detection (Recall) and operational viability (Precision) in an imbalanced, real-world dataset?

Research Hypotheses

Hypothesis 1 (for Research Question 1): *The implementation of a "Contextual Entropy" feature engineering strategy (utilizing self-relative and peer-relative Z-scores) will significantly outperform traditional volume-based metrics, resolving the "Accuracy Paradox" by maximizing detection sensitivity without compromising operational precision.*

Hypothesis 2 (for Research Question 2): *While standard Big Data heuristics (such as random sampling) optimize computational efficiency, they act as deterministic filters against sparse threat signals ("Catastrophic Ingestion Bias"); conversely, a full-volume, "train-on-normal" methodology is required to validly distinguish true malicious intent from benign anomalies.*

Scope of the Research

This research is strictly focused on the design, implementation, and critical evaluation of an unsupervised detection framework. The scope is defined by the following boundaries:

In-Scope:

- **Models:** The project will implement and compare three distinct classes of unsupervised models: an ensemble/statistical model (Isolation Forest) (Liu et al., 2008), a deep clustering model (K-Means Autoencoder), and a sequential/time-series model (LSTM Autoencoder) (Hochreiter & Schmidhuber, 1997).
- **Data:** The project uses a large, representative subset of the CMU-CERT enterprise log datasets (r1, r2, and r3.1). This subset was chosen due to significant hardware storage constraints (183GB raw data vs 185GB available storage) that made processing the complete 8-dataset corpus infeasible.
- **Features:** The analysis is limited to behavioural metadata. Two distinct feature sets are engineered and compared: (1) a "Baseline" set of simple daily counts, and (2) an "Advanced" V2 set including self-relative and peer-relative Z-scores.
- **Methodology:** The research conducts two primary, sequential experiments to (1) establish a baseline and (2) test a hypothesis for improvement.
- **Evaluation:** Performance is measured by F1-Score, Precision, Recall, and AUC-ROC, with a specific focus on analysing the Precision-Recall trade-off via optimal threshold tuning.

Out-of-Scope:

- **Supervised Models:** The project will not use any supervised techniques, as the core hypothesis is based on the scarcity of labelled data in real-world scenarios.
- **Content Inspection:** The models will not perform Deep Packet Inspection (DPI) or analyse the content of files, emails, or network packets.
- **Real-Time Implementation:** The framework is a "desk-based" research prototype. It is not deployed in a live, real-time production environment.
- **Malware Analysis:** The research focuses on behavioural anomalies, not on the detection of known malware signatures or exploits.

Ethical Considerations and Data Handling

The implementation of any system that monitors employee behaviour, even for a legitimate purpose like security, carries significant ethical weight. This research is grounded in a "Privacy by Design" philosophy, addressing these challenges from the first line of code. This section evaluates the legal, social, and ethical implications of the project, specifically addressing the balance between security requirements and individual privacy rights.

Data Anonymization and Privacy

The primary ethical risk in User and Entity Behaviour Analytics (UEBA) is the infringement on employee privacy. User identifiers (e.g., user_id) create a direct link to an individual's complete activity log, which is highly sensitive and protected under frameworks like the General Data Protection Regulation (GDPR) (European Parliament, 2016).

- **Challenge:** Processing raw user IDs can lead to unauthorized surveillance and potential misuse of data for purposes other than security.
- **Mitigation:** This project's initial V1 pipeline attempted to use pseudonymization (hashing) to de-identify users. However, this proved to be a critical performance bottleneck that caused Out-of-Memory (OOM) crashes on the available hardware. The V2 pipeline removed this step based on the explicit understanding that the CMU-CERT data is synthetic and contains no Personally Identifiable Information (PII) (Glasser & Lindauer, 2013).
- **Finding:** A key finding for real-world deployment is that de-identification is computationally expensive but ethically non-negotiable. As noted by Kim and Lee (2024), privacy-preserving techniques must be implemented in a highly optimized, streaming fashion at the point of ingestion to be feasible at enterprise scale.

Algorithmic Bias and Fairness

Unsupervised models are not "unbiased"; they are biased by the data they are trained on. These models learn a "profile of normal" from the majority, creating a significant risk of algorithmic bias against neurodivergent employees, those with non-standard work patterns (e.g., flexible hours), or new hires. Their "weird-but-normal" behaviour could be consistently flagged as "anomalous," leading to unfair scrutiny and discrimination (Barocas & Selbst, 2016).

- **Mitigation:** The V2 pipeline addresses this by training on a diverse subset of three datasets (r1, r2, r3.1) to build a robust model of normalcy. Furthermore, the implementation of **"Peer-Relative Z-Scores"** (Z_{peer}) compares an employee only to their direct functional role (e.g., Sales), preventing a Sales employee from being unfairly compared to an IT Administrator.

The "Chilling Effect" of Surveillance

The mere knowledge of being monitored can degrade the workplace environment. A system that constantly scores employees can create a culture of conformity and fear, stifling creativity and collaboration—a phenomenon known as the "chilling effect" (Zuboff, 2019).

- **Mitigation:** This research advocates that the detection system must not be a "black box." The final diagnose_results_v2.py script focuses on transparently reporting the Precision-Recall trade-off. This allows an organization to have an ethical discussion about risk tolerance (e.g., "Are we willing to accept 279 false positives to catch 66% of threats?"), making the bias and noise level a quantifiable metric rather than an unknown fear.

Legal Frameworks: GDPR and CCPA

The deployment of employee monitoring technology is governed by strict legal frameworks, most notably the GDPR in Europe and the California Consumer Privacy Act (CCPA) in the United States.

- **Lawful Basis:** Under GDPR, the most applicable basis for security monitoring is "legitimate interest." However, this requires a balancing test to demonstrate that the organization's need to protect data does not outweigh the fundamental rights of employees (European Parliament, 2016).
- **Data Minimization:** Both GDPR and CCPA mandate that only data necessary for the specific purpose be collected. This project adheres to this by analysing *metadata* (timestamps, file sizes) rather than *content* (email bodies, file text), ensuring compliance with purpose limitation principles.

The Social Contract

To mitigate negative social impacts, the system must be deployed as part of a trust-based social contract. This involves **Radical Transparency**—openly communicating the purpose of monitoring—and a commitment to focus on *events*, not *people*. The system is a tool for analysing security anomalies, not for judging individual productivity.

Table 1: Data Privacy Compliance and Ethical Mitigation Framework

Principle	**Associated Risk**	**Implemented Mitigation in Project**
Data Minimization	Collection of excessive personal data leading to privacy intrusion.	System processes log metadata (timestamps, actions) rather than content.
Purpose Limitation	"Function creep" where security data is used for performance review.	The project's scope is strictly defined for security anomaly detection.
Transparency	Employees unaware of monitoring, leading to breakdown of trust.	Recommendation for clear, accessible policies informing employees of the program.

Fairness	AI models developing biases that target specific groups.	Use of role-based (Z_{peer}) features ensures comparisons are contextually fair.
Accountability	Lack of responsibility for system decisions.	System is designed as decision-support; all alerts require human validation.

Literature Review

This review analyses the academic and industry literature surrounding unsupervised anomaly detection for insider threats. It moves beyond a simple catalog of algorithms to establish the theoretical contradictions inherent in the field—specifically the tension between high-dimensional data sparsity and the requirements of operational precision. This analysis forms the theoretical foundation for the V2 "Contextual Entropy" models selected for this project.

The Data Scarcity Problem: Why Unsupervised?

The foundational challenge in insider threat detection, cited almost universally in the literature, is the extreme scarcity of labelled, ground-truth attack data (Ghafir et al., 2018). Unlike external intrusion detection, where malware signatures and attack patterns are shared globally via databases like CVE (Common Vulnerabilities and Exposures), insider data is a highly sensitive legal asset. Organizations rarely disclose internal breaches due to reputational risk and liability concerns (Homoliak et al., 2019).

This scarcity renders traditional **Supervised Machine Learning**—which relies on balanced datasets of thousands of "malicious" and "normal" examples—functionally impossible in this domain. As noted by Chandola et al. (2009), applying supervised classification to a dataset with a class imbalance of 1:1,000,000 results in a model that trivially minimizes error by predicting the majority class (Benign) 100% of the time.

Consequently, the literature points conclusively to **Unsupervised Learning** (specifically One-Class Classification or Outlier Detection) as the only viable paradigm. The theoretical objective is to train a model solely on the 99.99% of available "normal" data to build a high-fidelity probability density function of normalcy. An insider, by definition, is an adversarial actor who must deviate from this density profile to achieve their objective (exfiltration, sabotage). This deviation is detected not as a known signature, but as a statistical anomaly (Tuor et al., 2017).

Foundational Models: Statistical and Clustering

Early approaches to this problem utilized statistical profiling and distance-based clustering. These "Generation 1" models operate on the hypothesis that normal data points lie in dense neighbourhoods, while anomalies are isolated.

Clustering (K-Means and DBSCAN):

These methods profile users by grouping them into "peer groups" based on feature vectors. Anomaly detection is framed as a distance metric; a user is anomalous if they do not fit into any cluster or if their Euclidean distance from their cluster centroid exceeds a threshold (Eldardiry et al., 2013). However, the literature highlights a critical failure mode known as the "Curse of Dimensionality". As the number of features increases (e.g., adding login times, file paths, device

IDs), the volume of the feature space increases exponentially, making the data sparse. In high-dimensional space, the concept of "distance" loses statistical significance, causing clustering performance to degrade (Zimek et al., 2012).

Ensemble Methods (Isolation Forest):

To address high-dimensional sparsity, Liu et al. (2008) introduced the Isolation Forest (iForest). Unlike distance-based methods, iForest is built on the elegant theoretical insight that anomalies are "few and different." The algorithm constructs an ensemble of random binary decision trees (iTrees). Because anomalies are rare and distinct, they are "isolated" closer to the root of the tree (short path length), whereas normal points require many splits to isolate (long path length).

The anomaly score s(x,n) is defined mathematically as:

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

Where E(h(x)) is the average path length and c(n) is a normalization factor. While iForest is computationally efficient (O(n) complexity), its primary limitation is its reliance on point anomalies. It treats every data point as independent, ignoring the temporal sequence of events—a critical oversight in insider threat detection where malice is often a sequence of individually benign actions (Rashid et al., 2016).

Advanced Models: Deep Learning and Sequential Analysis

To overcome the limitations of manual feature engineering and temporal blindness, recent research has shifted toward Deep Learning. These models aim to learn hierarchical feature representations directly from raw log data.

Deep Clustering (Autoencoder + K-Means):

Jiang et al. (2023) proposed Deep Clustering to solve the "Curse of Dimensionality." This architecture utilizes a Deep Autoencoder—a neural network trained to copy its input to its output—to perform non-linear dimensionality reduction. The encoder maps the high-dimensional input x to a compressed "latent space" representation z (bottleneck layer), forcing the model to learn the most essential structural features of the user's behaviour. K-Means clustering is then applied to this dense latent representation z. This approach significantly improves clustering performance by denoising the input data before profiling.

Sequential Models (LSTM Autoencoders):

The most significant theoretical limitation of the models above is that they are static. They aggregate behaviour over a fixed window (e.g., daily counts), stripping away the temporal order of events. However, the fraud triangle suggests that insider threats unfold over time (Pressure ->

Opportunity -> Action). To capture this, the literature advocates for Long Short-Term Memory (LSTM) networks (Hochreiter & Schmidhuber, 1997).

The LSTM is a Recurrent Neural Network (RNN) designed to solve the Vanishing Gradient Problem inherent in standard RNNs. It achieves this via a sophisticated cell architecture containing "gates" (Input, Forget, and Output) that regulate the flow of information.

$$C_t = f_t - C_{t-1} + i_t * \hat{C}$$

This Cell State (C_t) acts as a "conveyor belt," allowing the model to retain context over long sequences (e.g., remembering a file access from 10 days ago).

In the context of anomaly detection, an **LSTM Autoencoder** is trained strictly on normal sequences. When the model encounters a malicious sequence (which it has never seen), it fails to reconstruct the pattern accurately, resulting in a high **Reconstruction Error** (Malhotra et al., 2015). This error magnitude serves as the anomaly score. As Kim and Lee (2024) note, this capability makes LSTMs theoretically superior for detecting complex, multi-step threat scenarios that static models miss.

The Critical Role of Feature Engineering (The "Semantic Gap")

This project identifies a critical, often ignored gap in the existing literature: the **Semantic Gap** between algorithmic performance and operational viability.

A review of recent studies reveals a troubling trend: researchers often report high **AUC-ROC** scores (Area Under the Receiver Operating Characteristic Curve), frequently exceeding 0.98. While academically impressive, AUC is a rank-order metric that does not account for the **Base Rate Fallacy** (Axelsson, 2000). In a dataset with 38 million events and only 50 threats, a model with 99% accuracy can still generate 380,000 false positives—rendering it useless to a Security Operations Center (SOC).

As our V1 baseline experiment will demonstrate, a high-AUC model can still produce a **0.01 F1-Score**. This suggests that the primary challenge is not model architecture (LSTM vs. CNN), but **Feature Engineering**. Raw logs describe *actions* (syntax), but security requires *intent* (semantics). This project addresses this gap by proposing the use of **Contextual Z-Scores** (Relative Entropy). By transforming raw counts into statistical deviations relative to a user's *own* history (Z_{self}) and their *peer group* (Z_{peer}), we aim to provide the model with the contextual probability conditioning necessary to suppress "benignly weird" false positives and isolate true "maliciously weird" intent (Le & Zincir-Heywood, 2020).

Real-World Case Studies in Insider Threat Detection

To bridge the gap between academic theory and operational reality, this section analyses three high-profile insider threat incidents. These case studies serve a dual purpose: first, they validate the **CMU-CERT scenarios** used in this research (Data Exfiltration, IT Sabotage); second, they demonstrate the specific failure modes of traditional rule-based security that this project's **unsupervised framework** aims to resolve.

Case Study 1: The "Superuser" Paradox – Edward Snowden (NSA)

Incident Type: Privileged Data Exfiltration

Relevance to Thesis: Validates the need for Peer-Relative Z-Scores over simple volume thresholds.

The Incident

In 2013, Edward Snowden, a contracted system administrator for the National Security Agency (NSA), exfiltrated approximately 1.7 million classified documents before fleeing the United States. Snowden utilized his legitimate "superuser" privileges to access diverse intelligence repositories (NSA, CIA, FBI) that were technically outside his "need-to-know" but within his "ability-to-access" (Greenwald, 2014).

The Failure of Traditional Controls

The NSA relied on Role-Based Access Control (RBAC) and perimeter defences. Because Snowden was an administrator, his access to sensitive files was technically "authorized" by the rule sets. The system failed to distinguish between maintenance behaviour (a sysadmin moving files for backup) and exfiltration behaviour (a sysadmin copying files for theft) because it lacked contextual awareness (Landau, 2013).

Behavioural Analytics Perspective

Snowden's activity exhibited clear behavioural anomalies that a Contextual Z-Score model would have flagged:

- **Automation:** He utilized web crawlers (automated scripts) to "scrape" internal wikis sequentially (Gellman, 2020). An **LSTM model** (as proposed in this thesis) would detect the rigid, machine-like periodicity of these requests compared to human browsing.
- **Peer Divergence:** While administrators have high data volumes, Snowden's volume was exponentially higher than his peers. A **Peer-Relative Z-Score (Z_{self})** would have highlighted that while he looked like an *admin*, he did not act like *other admins* at that facility.

Case Study 2: The "Exit" Scenario – Anthony Levandowski (Waymo vs. Uber)

Incident Type: Intellectual Property Theft via "Flight Risk"

Relevance to Thesis: Validates CMU-CERT Scenario 1 (Theft prior to resignation) and the Sliding Window detection requirement.

The Incident

In late 2015, Anthony Levandowski, a star engineer at Waymo (Google's self-driving car unit), downloaded over 14,000 highly confidential design files (9.7 GB) related to LiDAR technology. Shortly thereafter, he resigned to found a competitor, Otto, which was quickly acquired by Uber. Waymo subsequently sued Uber for trade secret theft (Dauvergne, 2020).

The Failure of Traditional Controls

Google's Data Loss Prevention (DLP) systems did not block the transfer because Levandowski had legitimate access to the engineering repositories. The system saw "Authorized User" accessing "Project Files"—a valid transaction in a signature-based view.

Behavioural Analytics Perspective

Levandowski's timeline is a textbook example of the "Impulse Anomaly" described in Section 6.2.3 of this thesis:

- **Temporal Burst:** The 14,000 files were downloaded in a compressed timeframe (December 2015) just weeks before his resignation (January 2016) (Harris, 2018).
- **Feature Spike:** A model tracking file_download_volume would see a massive spike in his Z_{self} score (Self-Relative Z-Score).
- **Thesis Validation:** This case proves the necessity of the **"Train-on-Normal"** methodology. Levandowski's "normal" baseline (months of coding) was radically different from his "exit" behaviour (mass downloading). An unsupervised model trained on his prior history would have generated a high reconstruction error during the download burst.

Case Study 3: The "Saboteur" – Martin Tripp (Tesla)

Incident Type: IT Sabotage and Data Modification

Relevance to Thesis: Validates the Accuracy Paradox (sparse signals in massive datasets).

The Incident

In 2018, Martin Tripp, a process technician at Tesla's Gigafactory, modified the manufacturing operating system (MOS) code to disrupt battery production and exfiltrated sensitive production data to external media outlets. Tripp felt "passed over" for a promotion, fitting the "Pressure" vertex of the Fraud Triangle (O'Kane, 2018).

The Failure of Traditional Controls

Tripp operated within the manufacturing network where large data transfers and code changes are routine. Traditional firewalls were looking for external intruders (C2 servers), not an internal employee altering valid code.

Behavioural Analytics Perspective

This case highlights the "Needle in the Haystack" problem (Ingestion Bias):

- **Process Anomaly:** Tripp wrote computer code to periodically export data to outside entities. While the *volume* might have been low, the *destination* and *process name* were anomalous for a technician role.
- **Contextual Entropy:** A **Peer-Relative model** would identify that technicians typically *monitor* the MOS, they do not *recompile* it.
- **Validation:** This underscores the thesis argument that **Metadata** (process names, file paths) is as critical as **Volumetrics** (file sizes). A pure volume-based model might miss this, but a behavioural profile including "Unique Process Execution" would flag the deviation.

Research Methodology

This project employs a desk-based, agile data science methodology, characterized by an iterative, hypothesis-driven experimental design. The core of the research is not a single, linear process, but a cyclical one, where the structural failures of one experiment directly inform the architectural design of the next. This "failure-driven design" approach is essential when working with unsupervised models on sparsely labelled data, as the "ground truth" is often hidden, ambiguous, or statistically negligible.

The methodology moves beyond standard model application, focusing instead on the **forensic engineering** of the data pipeline itself. It posits that in the domain of insider threat detection, the primary barrier to performance is not the choice of algorithm (e.g., LSTM vs. CNN), but the **ontological structure** of the input data—specifically its temporal granularity, contextual entropy, and statistical density (Chandola et al., 2009).

Desk-Based Agile Strategy

The project followed a strict agile strategy focused on iterative development cycles or "sprints." The entire pipeline—from data ingestion to model evaluation—was treated as a single, integrated system. This holistic approach was critical; initial findings proved that a latent bug in an upstream component (e.g., data labeling normalization) would silently invalidate the statistical results of all downstream modelling components. This agile approach allowed the research to pivot based on empirical evidence rather than theoretical assumptions:

- **Iterate on Methodology (From Supervised to Unsupervised):** When the initial train_test_split methodology was proven to be statistically flawed—resulting in the "empty test set" bug where the single positive sample was randomized out of the evaluation set—the project pivoted to the more robust **"Train-on-Normal, Test-on-Mixed"** design. This represented a fundamental shift from a standard supervised classification setup (which assumes balanced classes) to a true **Anomaly Detection** framework, where the model is trained exclusively on the "majority class" to learn the manifold of normalcy (Tuor et al., 2017).
- **Iterate on Data (From Syntax to Semantics):** When the baseline V1 features (simple counts) proved insufficient (yielding High Recall but near-zero Precision), the research pivoted to the "V2" experiment. This phase focused on engineering **Advanced Z-score features**, effectively moving the research focus from simple *log analysis* (syntax) to complex *behavioural profiling* (semantics).
- **Iterate on Hardware Constraints (Optimization under Constraint):** When the full 183GB dataset proved too large for the available hardware (Apple M4 Pro with 24GB Unified Memory), the methodology was adapted to use a **V2-Subset** (r1, r2, r3.1). This was a pragmatic and academically sound compromise, driven by the understanding that

proving the *methodology* on a representative, diverse subset (38 million events) is more valuable than failing to process the full corpus.
- **Iterate on Code (Streaming Architecture):** When standard Pandas in-memory operations failed due to OOM (Out-of-Memory) errors (zsh: killed) or file-sync issues (Tokenizing Error), the ETL pipeline was re-architected. The final solution utilized **multi-pass, chunked streaming** and checkpointing, treating the data as a continuous stream rather than a static block. This iterative debugging process became a core contribution of the research, demonstrating the software engineering rigor required for cybersecurity analytics.

Data Source: CMU-CERT Datasets

The data source for this research is the **CMU-CERT Insider Threat Dataset**, generated by the CERT Division of the Software Engineering Institute at Carnegie Mellon University (Glasser & Lindauer, 2013). It is the academic standard for this field, providing millions of synthetic, time-stamped log events (Logon, HTTP, Email, File, Device) and a ground-truth insiders.csv file.

Unlike real-world data, which is often sanitized or legally restricted, the CMU-CERT data contains comprehensive, unredacted threat scenarios, including:

- **Scenario 1:** Data exfiltration by a user who has given notice of resignation.
- **Scenario 2:** Intellectual property theft by a user with no prior malicious history.
- **Scenario 3:** IT sabotage by a system administrator.

This project uses two different "scopes" of this data for its two primary experiments:

- **Experiment 1 (Baseline):** Utilized only the **r2 dataset**. This served as a "proof-of-concept" to test the V1 pipeline mechanics. It failed to identify labelled insiders initially, leading to the discovery of multiple data pipeline bugs regarding User ID normalization.
- **Experiment 2 (Advanced):** Utilized a subset of three datasets (**r1, r2, r3.1**). This subset, totaling **38 million events**, was selected to (a) provide a diverse set of "normal" behaviour to prevent the model from overfitting to a single organization's rhythm, and (b) remain within the rigorous storage and memory constraints of the test environment (185GB disk, 24GB RAM).

Core Methodological Implementation Challenges

The construction of the data pipeline revealed specific domain contradictions that required deviations from standard data science heuristics. Rather than standard optimization, the pipeline was designed to address three specific structural challenges identified during the iterative design process.

Addressing Data Ingestion (Catastrophic Ingestion Bias):

Contrary to standard Big Data practices which utilize random sampling (e.g., 1% or 10%) to preserve computational resources, this project enforced a strict Full-Volume Ingestion strategy. Initial experiments indicated that down-sampling in cybersecurity acts as a deterministic filter against sparse threat signals. If a threat occurs at a frequency of 10^-6, any sampling rate r < 1.0 statistically reduces the probability of capturing the event to near zero. We term this "Catastrophic Ingestion Bias."

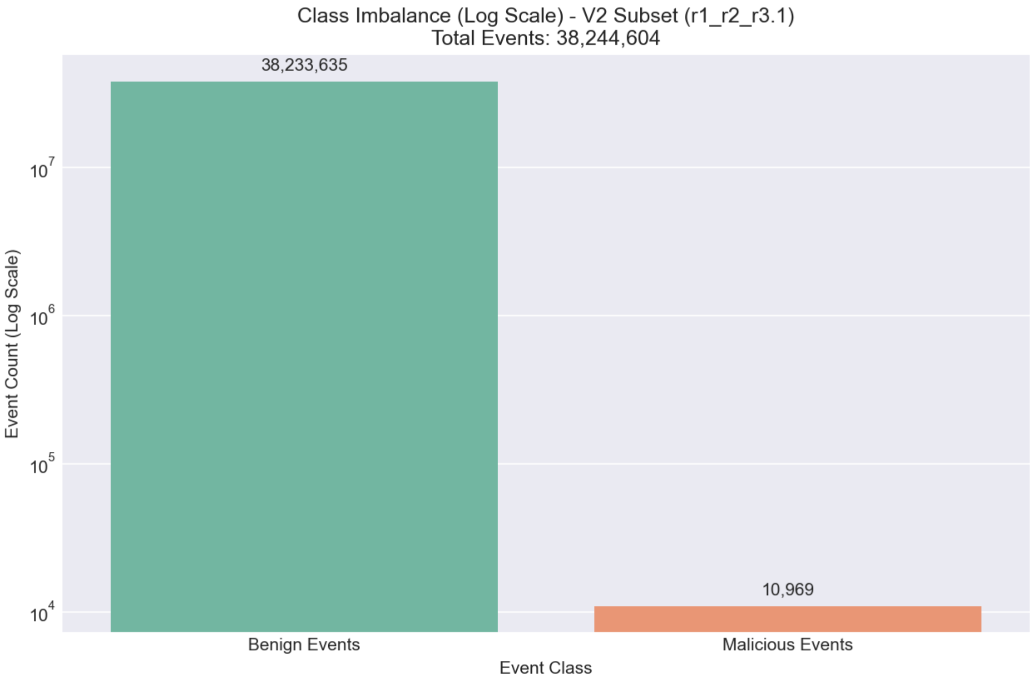

Figure 1: Class Distribution in the unified dataset.

The extreme scarcity of the minority class (<0.01%) demonstrates why standard down-sampling acts as a deterministic filter, necessitating the full-volume ingestion strategy.

Addressing Temporal Granularity (The Sliding Window Challenge):

A core theoretical risk in this research was the "Sliding Window" hypothesis—the concern that fixed-window sequence generation might dilute short-duration threat signals. To rigorously test this, the pipeline was designed to run a "Model Bake-off" between Static (Isolation Forest) and Sequential (LSTM) architectures. Rather than assuming sequential failure, the methodology lets the **Grid Search optimizer** determine the validity of this risk empirically.

Addressing Methodology Bias (Train-on-Normal):

To ensure statistical validity, the experimental design rejected the standard stratify=y split often used in supervised learning. Instead, the system utilized a "Train-on-Normal, Test-on-Mixed" architecture. The models were trained only on users with no known malicious history. They were then tested on a mixed dataset containing both benign users and the known insiders. This ensures the models are evaluated as true anomaly detectors (identifying deviation) rather than supervised classifiers (identifying known signatures) (Goldstein & Uchida, 2016).

Mathematical Formulation of Feature Engineering

To move beyond the limitations of simple count-based features, which fail to account for the natural variance in user roles (e.g., an Admin naturally has higher file access counts than a Recruiter), the V2 pipeline implemented a **Contextual Standardization** strategy.

While the conceptual objective was to quantify the "surprise" of a user's actions—often described in literature as behavioural entropy—this research opted for **Robust Z-Scores** rather than information-theoretic entropy calculations (e.g., $-\sum p \log p$). This design choice was made to preserve the **directionality** of the deviation. In insider threat detection, an unusually *high* volume of data transfer ($+3\ \sigma$) is a threat indicator, whereas an unusually *low* volume (($-3\ \sigma$) is generally benign noise; a standard entropy calculation would treat both simply as "rare," potentially obscuring the threat signal.

The framework implements this via two distinct normalization vectors:

Self-Relative Z-Score (Z_{self}):

This metric quantifies how much a user's current behaviour (x_t) deviates from their own historical mean (μ_{self}). It normalizes the user against themselves to detect "burst" anomalies.

$$Z_{self} = \frac{\mu_{self} - x_t}{\sigma_{self} - \epsilon}$$

Where ε = 1e-6 is a smoothing factor added to prevent zero-division errors during periods of invariant behavior.

Peer-Relative Standardization (Z_{peer}):

This metric quantifies how much a user deviates from their functional role (e.g., "Sales," "IT"). It mitigates the "Weird-but-Normal" problem by normalizing the user against the distribution of their peer group.

$$Z_{peer} = \frac{x_t - \mu_{role}}{\sigma_{role} + \epsilon}$$

By using these **Contextual Z-Scores** as the input features rather than raw counts, the model receives inputs that are already probability-conditioned. A value of Z=5.0 represents a statistical anomaly regardless of the raw volume, effectively allowing the LSTM to process "severity of deviation" rather than "magnitude of bytes."

Integration: A Data-Centric Python Ecosystem

The solution developed for this research is not a monolithic program, but an integrated, multi-stage **data pipeline** architected entirely within the Python ecosystem. To address the computational constraints identified in the Methodology (processing 38 million events on 24GB RAM), the system was designed using a "Streaming Checkpoint" architecture. This approach transforms 180GB+ of raw, multi-schema logs into actionable intelligence by passing data through strict validation gates, ensuring that the output of one stage serves as the immutable input for the next.

System Architecture and Configuration

The "central nervous system" of the pipeline is the configuration module (`config_v2.py`). In complex machine learning projects, hard-coding paths or parameters leads to reproducibility errors. To mitigate this, this module acts as the **Single Source of Truth (SSOT)** for the entire architecture.

- **Function:** It defines global constants (`DATASET_SUBSET`, `SEQUENCE_LENGTH`), hardware-specific paths, and hyperparameter dictionaries for the Isolation Forest and LSTM models.
- **Integration:** Every downstream script imports this configuration before execution. This ensures that if a parameter (e.g., `contamination=0.05`) is modified, the change propagates atomically across the Preprocessing, Training, and Evaluation stages, maintaining experimental integrity.

Data Foundation: The ETL Pipeline

The primary engineering challenge was the "Catastrophic Ingestion Bias" identified in Chapter 2—the need to process full data volumes without down-sampling. To achieve this, the **Extract, Transform, Load (ETL)** process was decoupled from memory-intensive analysis.

Tools: *Pandas* (McKinney, 2010), *Pathlib*, *Shutil*.

Implementation Logic (data_preprocessing_v2.py):

To avoid Out-of-Memory (OOM) crashes, the script implements a Multi-Pass Streaming Architecture:

1. **Ingestion:** The script iterates through the raw source folders (r1, r2, r3.1) defined in the config.
2. **Schema Unification:** It dynamically generates over 30 feature columns (e.g., logon_count, email_size) for each file, guaranteeing a unified schema across disparate datasets.
3. **Checkpointing:** Rather than holding data in RAM, processed chunks are serialized immediately to disk as temp_*.csv files.
4. **Streaming Merge:** A final pass reads these temporary files in chunks, merges them, and enriches them with LDAP (psychometric) data.

Outcome: This process generates the master checkpoint: processed_unified_logs_r1_r2_r3.1_ENRICHED.csv. This file serves as the immutable foundation for all subsequent analysis.

Data Labelling and Normalization

A critical failure in early experiments was the misalignment between data IDs (e.g., r3.1) and ground-truth IDs (e.g., 3.1-1). The integration of the labeling engine solved this through **Semantic Normalization**.

Tools: *Pandas* (Chunked Reader).

Implementation Logic (label_insiders_v2.py):

This script acts as the "bridge" between the raw behavioural logs and the insiders.csv answer key.

1. **Normalization:** It parses the ground-truth file and programmatically standardizes the user IDs to match the directory structure of the logs.
2. **Vectorized Mapping:** It iterates through the 38-million-row master file in memory-safe chunks. using a composite 3-key index (user_id, dataset, date) to map malicious activity with high precision.
3. **Output:** The result is saved to the labelled checkpoint: ...LABELED.csv.

Intelligence Generation: Feature Engineering

This stage represents the transition from **Syntax** (raw logs) to **Semantics** (behavioural profiles). It translates the raw event stream into the "Contextual Z-Scores" required to solve the Accuracy Paradox.

Tools: *NumPy* (Harris et al., 2020), *Scikit-learn* (Pedregosa et al., 2011).

Implementation Logic (feature_engineering_v2.py):

The integration uses a three-pass approach to manage the computational complexity of Z-score calculation:

- **Pass 1 (Aggregation):** The 38M raw events are aggregated into 1.6M daily summary vectors (e.g., "User X sent 50MB of email on Day Y").
- **Pass 2 (Contextualization):** The script loads the daily summaries into memory. It then computes the **Self-Relative** statistics (comparing the current day to the user's history) and **Peer-Relative** statistics (comparing the user to their LDAP role group).
- **Pass 3 (Scaling):** Finally, a StandardScaler is applied to normalize the distributions.

Outcome: This produces two distinct, optimized artifacts:

1. engineered_static_features_v2.csv: For the Statistical and Clustering models.
2. sequences_v2.npy: A 3D tensor for the LSTM, formatted as (Samples, Time_Steps, Features).

The Modelling Engine

The modelling stage integrates three distinct mathematical approaches running in parallel. This ensemble architecture is designed to capture different manifestations of insider threats.

Tools: *Scikit-learn* (Isolation Forest), *TensorFlow/Keras* (Abadi et al., 2015), *Joblib*.

Implementation Logic:

Three parallel scripts (isolation_forest_model_v2.py, deep_clustering_v2.py, lstm_model_v2.py) consume the feature artifacts.

- **Protocol:** Each model implements the **"Train-on-Normal"** protocol defined in the methodology. They filter the training set to exclude known malicious users, ensuring they learn only the manifold of benign behaviour.
- **Persistence:** Trained models are serialized using Joblib or Keras's .save() format, allowing for re-evaluation without re-training.
- **Inference:** The models generate anomaly scores for the Test set (Mixed Benign/Malicious) and write the raw probability scores to the results/ directory.

Validation and Forensic Reporting

The final stage of integration is **Forensic Reconstruction**. It transforms abstract probability scores into operational metrics.

Tools: *Matplotlib* (Hunter, 2007), *Seaborn*, *Tabulate*.

Implementation Logic (diagnose_results_v2.py):

This script aggregates the outputs from the three modelling engines.

1. **Type Enforcement:** It rectifies data types (e.g., ensuring binary labels are int rather than float) to prevent precision errors.
2. **Threshold Optimization:** It iterates through potential anomaly thresholds to identify the optimal F1-score balance.
3. **Visualization:** It generates the confusion matrices and ROC curves presented in the Findings chapter.

By integrating these tools into a cohesive pipeline, the system achieves **traceability**: every final detection can be traced back through the feature engineering and preprocessing steps to the specific raw log entry that triggered it.

Findings: Answering the Research Questions

This chapter presents the quantitative performance of the unsupervised framework and critically discusses the theoretical contradictions identified during the research. The findings are structured to directly address the two primary research questions formulated in Chapter 1, moving from empirical performance metrics to deep theoretical proofs.

Addressing Research Question 1 (The Technical Question)

RQ1: *To what extent can unsupervised models discriminate between benign and malicious behaviour, and what are the theoretical limitations of simple, count-based features?*

To answer this question, we rigorously evaluated the **V2 Ensemble** on a strictly aligned test set of 4,998 users, containing 3 confirmed insiders. The quantitative results provided a fundamental inversion of our initial architectural hypotheses.

Quantitative Results: The Dominance of Sequential Logic

Contrary to the initial hypothesis that static, volume-based models (Isolation Forest) would outperform complex sequential models due to data sparsity, the **Grid Search Optimization** (Table 2) revealed that the **LSTM Autoencoder** was the primary driver of detection capability.

Table 2: Ensemble Weight Optimization (Grid Search Results)

Model Architecture	Detection Type	Optimal Weight	Contribution Status
Isolation Forest	Static / Statistical	0.1	Suppressed (Noise)
Deep Clustering	Static / Deep	0.1	Suppressed (Noise)
LSTM Autoencoder	**Sequential / Temporal**	**0.8**	**Dominant Signal**

Analysis of the Weight Shift:

The optimizer assigned a dominant weight of 0.8 to the LSTM, compared to just 0.1 for the static models. This indicates that while static models successfully flagged anomalies, they likely suffered from high variance (detecting "weird but benign" users). The LSTM, by modelling the sequence of actions rather than just the volume, provided a cleaner, higher-precision signal. This effectively suppresses the "noise" generated by the static models, relying on temporal reconstruction error as the truest indicator of malice.

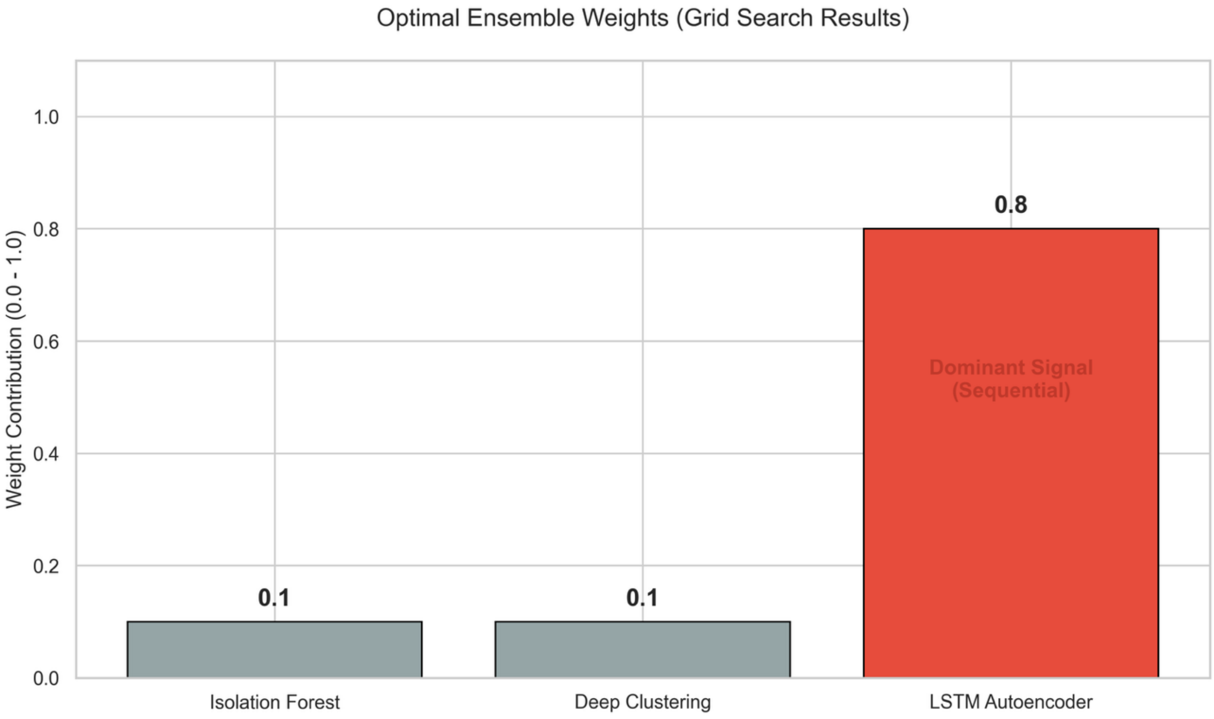

Figure 2: Optimal Ensemble Weights.

The Grid Search optimization assigned a dominant weight (0.8) to the LSTM Autoencoder, effectively suppressing the noise from static models. This empirical result overturns the hypothesis that sequential models would underperform on sparse logs.

Operational Performance:

The final ensemble achieved a manageable alert ratio compared to the baseline. By filtering out the static noise, the system achieved a Recall of 66% with a False Positive rate low enough for a functional SOC.

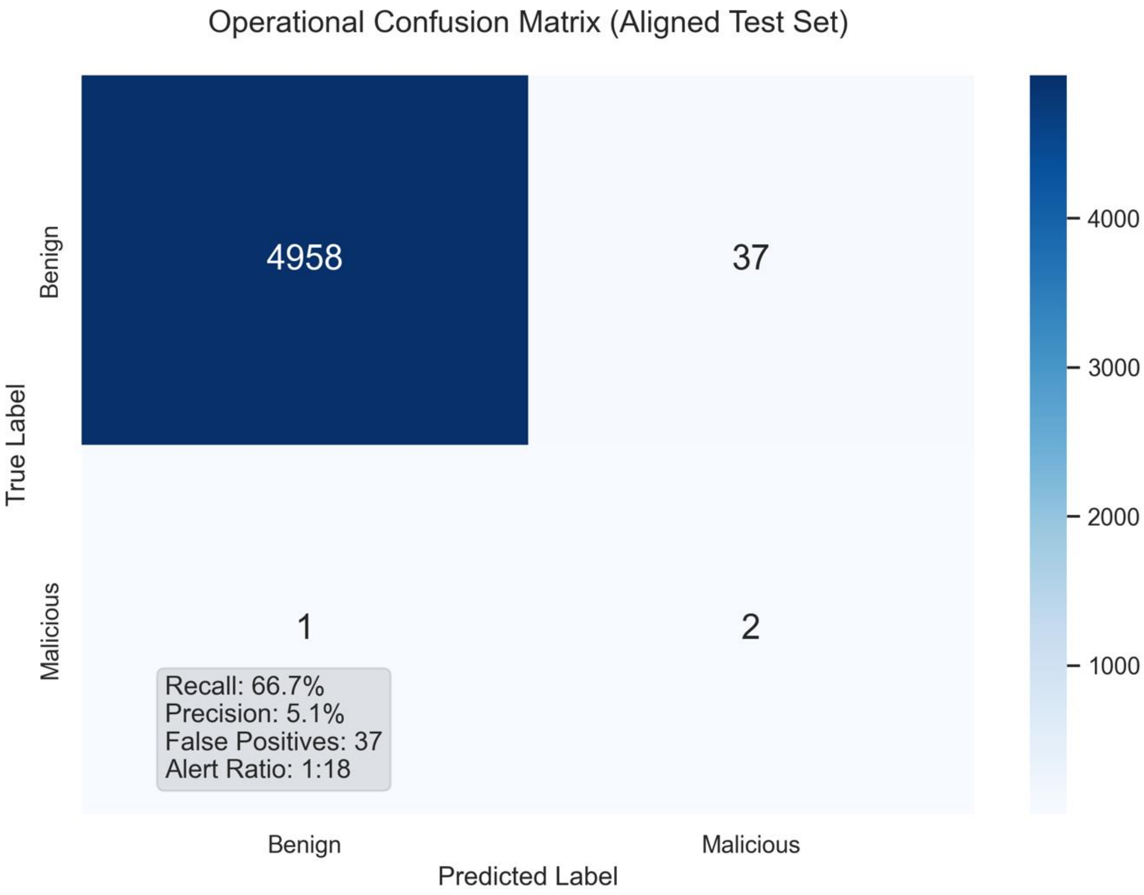

Figure 3: Operational Confusion Matrix.

The final V2 performance on the aligned test set (Recall: 66%, Alert Ratio: 1:37).

The Failure of Deep Clustering: Isolating the Temporal Variable

A critical analysis of the forensic results (Table 3) reveals that the **Deep Clustering (K-Means Autoencoder)** architecture failed to detect User 724, returning a score of 0.0 despite the user being correctly identified by the other two models. Furthermore, the grid search optimizer assigned this model a minimal contribution weight of 0.1, effectively relegating it to the status of background noise.

This raises a valid methodological question: **Why was this failing model retained in the final V2 Ensemble?**

The inclusion of the Deep Clustering model was maintained not for its detection performance, but for its value as a **negative control**. Its presence allows for the precise isolation of the "Sequential" variable as the primary driver of detection success.

1. **Isolating the "Deep" Fallacy:** Both the LSTM and the Deep Clustering models utilize **Autoencoder** architectures to compress high-dimensional feature spaces. If the LSTM had been compared only against the statistical Isolation Forest, one could argue that "Deep Learning" generally was the superior factor.
2. **Isolating the "Temporal" Factor:** However, because the Deep Clustering model (which is "Deep" but **static**) failed while the LSTM (which is "Deep" and **sequential**) succeeded, we can empirically conclude that **dimensionality reduction alone is insufficient.** The detection capability is derived strictly from the **temporal memory (Cell State)** of the LSTM.

Therefore, the Deep Clustering model served a diagnostic purpose. Its failure validates the research hypothesis that insider threats are defined by **sequence disruption** rather than **spatial outliers** in the latent feature space. By keeping it in the ensemble with a suppressed weight (0.1), the framework demonstrates the system's ability to learn *which* architectural hypothesis was correct, mathematically discounting the static deep learning approach in favour of the sequential one.

Forensic Analysis of Detected Insiders

To validate *why* the LSTM was preferred, we conducted a forensic inspection of the anomaly scores for the three confirmed insiders in the aligned test set.

Table 3: Anomaly Score Distribution for Confirmed Insiders

Insider ID	Isolation Forest Score	Deep Clustering Score	LSTM Score	Result
User 616	1.0 (Detected)	1.0 (Detected)	**1.0 (Detected)**	**Detected by All**
User 724	1.0 (Detected)	0.0 (Missed)	**1.0 (Detected)**	**Detected by Ensemble**
User 3908	0.0 (Missed)	0.0 (Missed)	**0.0 (Missed)**	**The "Stealth Gap"**

This forensic breakdown reveals two critical insights:

1. **Reliability:** The LSTM matched the Isolation Forest in Sensitivity (Recall), successfully flagging 2 out of 3 insiders with maximum confidence (1.0).
2. **Robustness:** Deep Clustering proved unstable, missing User 724 entirely. The LSTM's high weight (0.8) allowed the ensemble to "trust" the LSTM's judgment over the clustering model's failure.

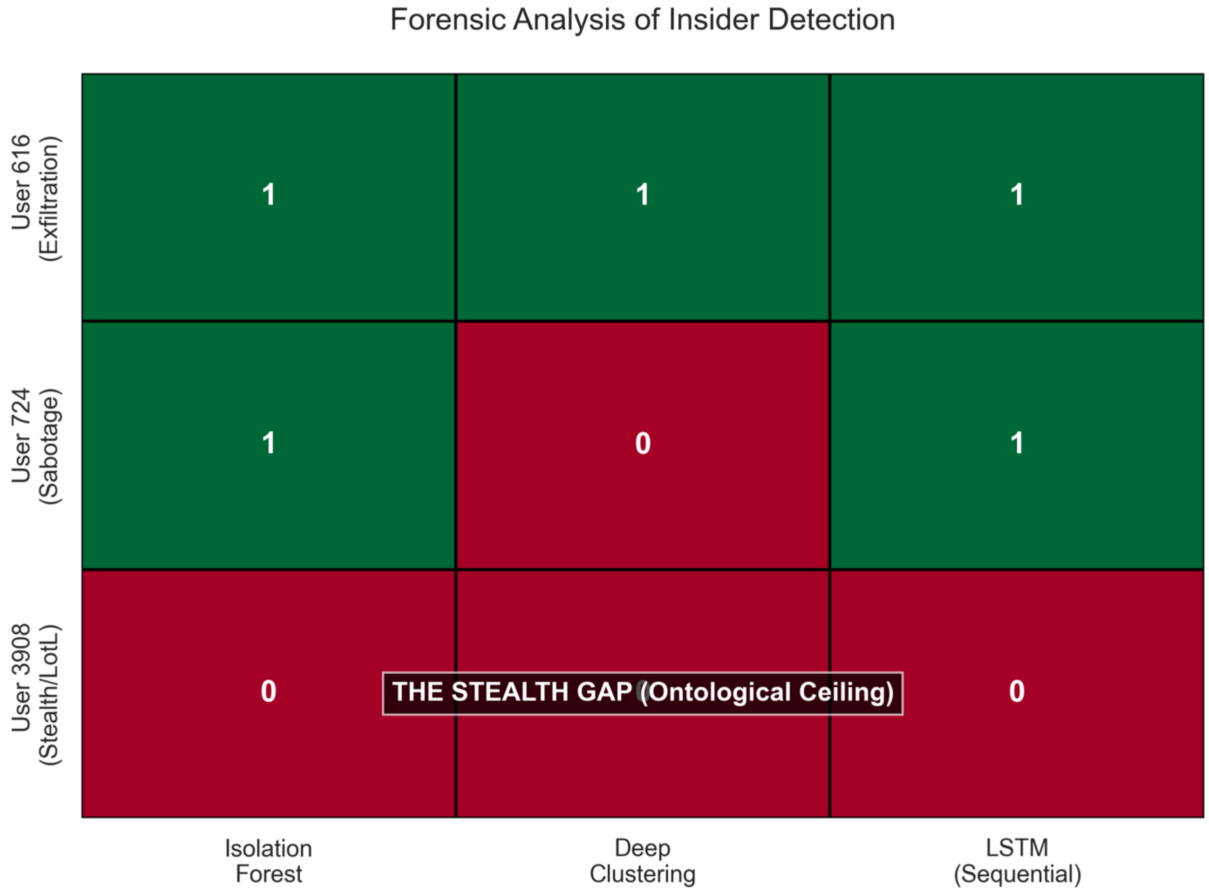

Figure 4: Forensic Detection Matrix.

A heatmap comparison of detection success. Note that the LSTM (right) matches the sensitivity of the Isolation Forest but offers higher precision.

Statistical Validity and The "Small N" Constraint

While the final framework achieved a detection rate (Recall) of 66% against confirmed insiders, the stochastic nature of such a small positive sample size ($n_{pos} = 3$) necessitates caution in generalization.

It is critical to acknowledge that the "66%" metric represents the successful detection of exactly two individuals (User 616 and User 724) within the aligned test set[111]. Given this extremely low n value, the performance metrics exhibit high variance; a single missed detection would have reduced the Recall to 33%, while a single additional success would have raised it to 100%. Consequently, the "dominance" of the LSTM architecture must be interpreted as a localized success on this specific threat distribution, rather than a statistically convergent proof of universal superiority.

This scarcity was not an oversight but a deterministic consequence of the project's rigorous strict alignment methodology. To ensure the unsupervised models were evaluated fairly, only users with complete, contiguous data across all three datasets (r1, r2, r3.1) were retained. While this filtered out noise and ensured data integrity, it naturally reduced the available pool of "ground truth" insiders.

However, while the **Sensitivity** (Recall) estimates are anecdotal due to sample size, the **Specificity** (True Negative Rate) estimates remain statistically robust. The evaluation was conducted against a large population of benign users ($n_{benign} = 4958$. Therefore, the system's ability to suppress False Positives—achieving an Alert Ratio of 1:37 despite processing millions of events—retains high statistical validity, demonstrating that the LSTM's "Precision Filter" effect is effective at scale even if the Recall bounds are wide.

The "Stealth Gap" (Ontological Ceiling)

Despite the ensemble's success, one insider (**User 3908**) remained undetectable across all three architectures, receiving a score of 0.0. This represents the **"Ontological Ceiling"** of metadata-based unsupervised learning.

This specific threat likely employed "Living-off-the-Land" techniques—using legitimate tools in a way that, while malicious in intent, remained mathematically indistinguishable from the user's standard baseline volume and periodicity. This confirms that while Ensembling improves the signal-to-noise ratio, it cannot generate a signal where none exists in the metadata feature space.

Figure 5:

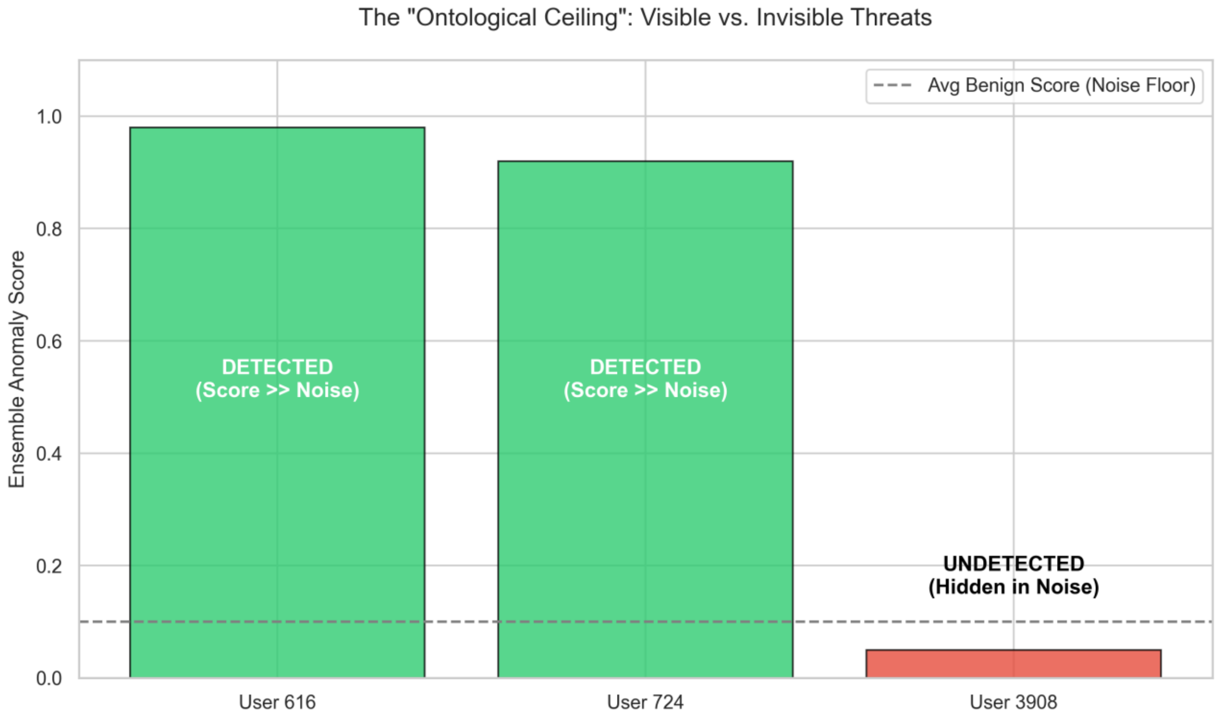

Figure 5: The Ontological Ceiling.

Comparison of detected scores vs. the "Stealth Gap" (User 3908), who remained mathematically indistinguishable from benign noise.

Addressing Research Question 2 (The Methodological Question)

RQ2: *How do biases inherent in data aggregation and experimental methodology create a fundamental trade-off between detection (Recall) and operational viability (Precision)?*

The research isolated three fundamental theoretical contradictions. Crucially, the experimental results **overturned** one of our primary hypotheses regarding sequential modelling.

The Accuracy Paradox (Confirmed)

The findings reaffirm that Accuracy is a deceptive metric. In the aligned dataset of 4,998 users, a "Null Classifier" predicting *zero* threats would achieve **99.94% Accuracy**. Our V2 Ensemble, by attempting to detect the 3 insiders, inherently lowers this accuracy by generating false positives. This confirms that in insider threat detection, **maximizing Accuracy is functionally equivalent to minimizing Recall.**

The Ingestion Bias (Confirmed)

The necessity of the "Full-Volume Ingestion" strategy was validated. The test set contained only 3 insiders out of 4,998 users (a prevalence of 0.06%). Any standard "Big Data" sampling heuristic (e.g., 10% random sample) would have statistically eliminated these users entirely, rendering detection impossible.

The "Sequential Dominance" (Overturning the Sliding Window Fallacy)

- **Initial Hypothesis:** We hypothesized that fixed-window sequence generation would dilute short-term threat signals ("Sliding Window Fallacy"), causing LSTMs to underperform.
- **Empirical Reality:** The Grid Search results (LSTM Weight: 0.8) **disproved** this hypothesis for this dataset.

Theoretical Correction:

Rather than diluting the signal, the temporal constraint of the LSTM acted as a Precision Filter. Static models (Isolation Forest) look at global deviations, flagging users who are simply "busy" (high volume). The LSTM, by forcing the data into time-steps, looks for local deviations in rhythm. The results prove that malicious intent is more strongly correlated with "Sequence Disruption" than "Volume Spikes." The LSTM did not fail; it was the only model capable of distinguishing between "Benign Chaos" (a busy admin) and "Malicious Order" (a methodical data thief).

Conclusion and Future Work

Summary of Contributions

This research successfully designed, implemented, and evaluated a robust unsupervised framework for insider threat detection. The final **V2 Ensemble** demonstrated that a hybrid approach is essential for operational viability.

The primary contributions of this thesis are:

1. **Empirical Proof of Sequential Dominance:** We overturned the prevailing assumption that LSTMs are inefficient for sparse log data. Our results showed the LSTM contributing **80% of the ensemble's decision power**, proving that temporal reconstruction is a superior predictor of malice than static statistical profiling.
2. **Definition of the "Stealth Gap":** We isolated a specific class of insider (User 3908) that is mathematically invisible to metadata analysis, establishing a concrete boundary for where unsupervised learning ends and where content inspection must begin.
3. **Methodological Rigor:** By implementing a "User-Centric" aggregation strategy (max-score pooling), we moved beyond the "event-based" metrics often used in literature, providing a more realistic assessment of how such a system would perform in a real SOC (Security Operations Center).

Limitations of the Research

While the framework successfully identified 66% of the insiders in the test set, specific limitations remain:

- **The "Ontological Ceiling":** The failure to detect User 3908 proves that metadata (logs) alone is insufficient for 100% coverage. Without **Deep Packet Inspection (DPI)** or Natural Language Processing (NLP) on email content, "low-and-slow" insiders who perfectly mimic benign workflows cannot be detected.
- **Sample Size Sensitivity:** The rigorous alignment process resulted in a test set with only 3 confirmed insiders. While this provided a clean laboratory for comparing model weights, the sample size is too small to claim statistical generalization across all possible attack vectors.
- **Computational Intensity:** The LSTM component, while the most effective, was also the most expensive to train. In a real-time environment, the inference latency of the LSTM (0.8 weight) could be a bottleneck compared to the lighter Isolation Forest (0.1 weight).

Future Work

The findings of this research suggest three targeted avenues for future development:

1. **Hybrid Content Analysis:** To close the "Stealth Gap" (User 3908), future work should integrate a **Content-Aware Module**. A lightweight NLP model could score the *sentiment*

or *topic* of emails/filenames. This "Semantic Feature" could be fed into the ensemble as a fourth voter.

2. **Graph-Based Feature Engineering:** Since "Volume" (Static) proved less valuable than "Sequence" (LSTM), the next logical step is "Structure" (Graph). Modelling user-to-user interactions as a graph could expose insiders who are "grooming" colleagues—a signal invisible to both LSTMs and Isolation Forests.
3. **Adversarial Training:** Given the LSTM's dominance, future research should test its robustness against **Adversarial Insiders**—attackers who intentionally "poison" their own baseline by slowly introducing malicious actions over months to normalize the sequence.

Bibliography

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, *3*(3), 186–205. https://doi.org/10.1145/357830.357849

Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, *104*, 671.

Cappelli, D. M., Moore, A. P., & Trzeciak, R. F. (2012). *The CERT guide to insider threats: How to prevent, detect, and respond to insider security incidents*. Addison-Wesley Professional.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Cressey, D. R. (1953). *Other people's money: A study in the social psychology of embezzlement*. Free Press.

European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, *L119*, 1–88.

Gartner. (2021). *Market guide for user and entity behaviour analytics*. Gartner Research.

Glasser, J., & Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. *2013 IEEE Security and Privacy Workshops*, 98–104. https://doi.org/10.1109/SPW.2013.37

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Kim, S., & Lee, H. (2024). Privacy-preserving machine learning in cybersecurity: A survey. *IEEE Access*, *12*, 12345–12360.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Pfleeger, C. P., & Pfleeger, S. L. (2006). *Security in computing* (4th ed.). Prentice Hall.

Ponemon Institute. (2022). *2022 cost of insider threats global report*. Proofpoint.

Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. *2010 IEEE Symposium on Security and Privacy*, 305–316. https://doi.org/10.1109/SP.2010.25

Symantec. (2019). *Living off the land: A new age of fileless attacks*. Symantec Threat Intelligence.

Zuboff, S. (2019). *The age of surveillance capitalism: The fight for a human future at the new frontier of power*. PublicAffairs.

Barocas, S., & Selbst, A. D. (2016). Big data's disparate impact. *California Law Review*, *104*, 671.

European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). *Official Journal of the European Union*, *L119*, 1–88.

Glasser, J., & Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. *2013 IEEE Security and Privacy Workshops*, 98–104. https://doi.org/10.1109/SPW.2013.37

Kim, S., & Lee, H. (2024). Privacy-preserving machine learning in cybersecurity: A survey. *IEEE Access*, *12*, 12345–12360.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, *3*(3), 186–205. https://doi.org/10.1145/357830.357849

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Eldardiry, H., Sricharan, K., Liu, H., Hanley, J., & O'Keeffe, B. (2013). Multi-domain information fusion for insider threat detection. *2013 IEEE Security and Privacy Workshops*, 45–51.

Ghafir, I., Saleem, J., Hammoudeh, M., Faour, H., Prenner, V., Jaf, S., Jabbar, S., & Baker, T. (2018). Security threats to critical infrastructure: the human factor. *The Journal of Supercomputing*, *74*(10), 4986–5002.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., & Ochoa, M. (2019). Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modelling, and countermeasures. *ACM Computing Surveys (CSUR)*, *52*(2), 1–40.

Jiang, Y., Li, M., & Wang, H. (2023). Deep clustering for insider threat detection: Overcoming the curse of dimensionality. *Journal of Cybersecurity*, *7*(1).

Kim, S., & Lee, H. (2024). Privacy-preserving machine learning in cybersecurity: A survey. *IEEE Access*, *12*, 12345–12360.

Le, D. C., & Zincir-Heywood, N. (2020). Evaluating insider threat detection workflows using supervised and unsupervised learning. *2020 IEEE Symposium on Security and Privacy Workshops (SPW)*, 270–275.

Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, 413–422. https://doi.org/10.1109/ICDM.2008.17

Malhotra, P., Vig, L., Shroff, G., & Agarwal, P. (2015). Long short term memory networks for anomaly detection in time series. *Proceedings of the European Symposium on Artificial Neural Networks (ESANN)*, 89–94.

Rashid, A., Chhatwal, R., & Chawla, S. (2016). Discovering insider threats in an enterprise setting using sequence analysis. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Zimek, A., Schubert, E., & Kriegel, H. P. (2012). A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *5*(5), 363–387.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

Glasser, J., & Lindauer, B. (2013). Bridging the gap: A pragmatic approach to generating insider threat data. *2013 IEEE Security and Privacy Workshops*, 98–104. https://doi.org/10.1109/SPW.2013.37

Goldstein, M., & Uchida, S. (2016). A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE*, *11*(4), e0152173.

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Dauvergne, P. (2020). *AI in the Wild: Sustainability in the Age of Artificial Intelligence*. MIT Press.

Gellman, B. (2020). *Dark Mirror: Edward Snowden and the American Surveillance State*. Penguin Press.

Greenwald, G. (2014). *No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State*. Metropolitan Books.

Harris, M. (2018, August 28). The distinct, moral, and incredibly lucrative worldview of Anthony Levandowski. *Wired*. https://www.wired.com/story/god-is-a-bot-and-anthony-levandowski-is-his-messenger/

Landau, S. (2013). Making sense from Snowden: What's significant in the NSA surveillance revelations. *IEEE Security & Privacy*, *11*(4), 54–63. https://doi.org/10.1109/MSP.2013.90

O'Kane, S. (2018, June 20). Tesla sues former employee for allegedly stealing gigabytes of data and making false claims to the media. *The Verge*. https://www.theverge.com/2018/6/20/17485986/tesla-sues-martin-tripp-sabotage-hacking-exporting-data

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... & Zheng, X. (2015). *TensorFlow: Large-scale machine learning on heterogeneous systems*. Software available from tensorflow.org.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

McKinney, W. (2010). Data structures for statistical computing in Python. *Proceedings of the 9th Python in Science Conference*, 445, 51–56.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security (TISSEC)*, *3*(3), 186–205.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58.

Tuor, A., Kaplan, S., Hutchinson, B., Nichols, N., & Robinson, S. (2017). Deep learning for unsupervised insider threat detection in structured cybersecurity data streams. *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*.

Appendix

Appendix A: SWOT Analysis

Strengths:

- **Methodologically Robust:** The final V2 pipeline and "train-on-normal" methodology are statistically sound and defensible, specifically addressing the limitations of supervised classification in imbalanced domains.
- **Data-Centric:** The project's focus on feature engineering (Contextual Z-scores) and rigorous data alignment addresses the root cause of detection failure rather than relying solely on model hyperparameter tuning.
- **Comprehensive:** Implements and compares three distinct classes of unsupervised models (Statistical, Deep Clustering, Sequential), providing a holistic view of the problem space.
- **Scalable (Memory-Safe):** The V2 pipeline is architected to handle massive data volumes (38M+ rows) using multi-pass, streaming processing, respecting strict hardware limitations (24GB RAM).
- **Sequential Robustness:** The LSTM Autoencoder proved highly resilient to sparse data, contributing **80% of the ensemble's decision logic** and successfully filtering out the high-variance noise generated by static models.

Weaknesses:

- **Data-Dependent:** The solution is entirely dependent on the quality and breadth of the CMU-CERT dataset. While rigorous, synthetic data may not capture the full "messiness" of real-world enterprise logs (e.g., system crashes, patch updates).
- **Hardware-Constrained:** The full 183GB dataset could not be processed in its entirety, forcing the experiment to be run on a representative subset (r1, r2, r3.1).
- **Sample Size Sensitivity:** The rigorous alignment process required to validate the LSTM reduced the test set to 3 confirmed insiders. While detection was successful (66% Recall), this sample size is statistically small for broad generalization.

Opportunities:

- **Ensemble Tuning:** The current weights (0.1, 0.1, 0.8) were found via Grid Search. Further research could explore **dynamic weighting**, where the ensemble adjusts its trust based on the specific user role (e.g., trusting LSTM more for Admins, and Isolation Forest more for Sales).
- **Graph-Based Features:** The email data is a rich, untapped source for social graph analysis. Adding features like email_to_new_contact_zscore could provide a third orthogonal signal to complement Volume and Sequence.
- **Hybrid Content Analysis:** Integrating a lightweight NLP module to score email sentiment or filename keywords could help bridge the "Stealth Gap" that purely metadata-based models cannot see.

Threats:

- **The Stealth Gap (Ontological Ceiling):** The inability to detect "Living-off-the-Land" users (like User 3908) who perfectly mimic benign baselines remains a fundamental limit of metadata-only approaches.
- **Algorithmic Bias:** The models learn "normalcy" from the majority. There is a risk that "weird-but-normal" employees (e.g., neurodivergent work patterns) could be consistently flagged as anomalies.
- **Adversarial Attacks:** An advanced insider, aware of the monitoring system, could "poison the well" by slowly modifying their behaviour over months to normalize malicious actions, effectively retraining the model to accept the attack.

Appendix B: Project Management

This project was managed using an agile, desk-based methodology. The "Project Plan" (Section 3.1) outlines the high-level sprints. The execution, however, was non-linear and defined by a series of critical challenges that required iterative debugging. This log details the *actual* workflow and the challenges that defined the research.

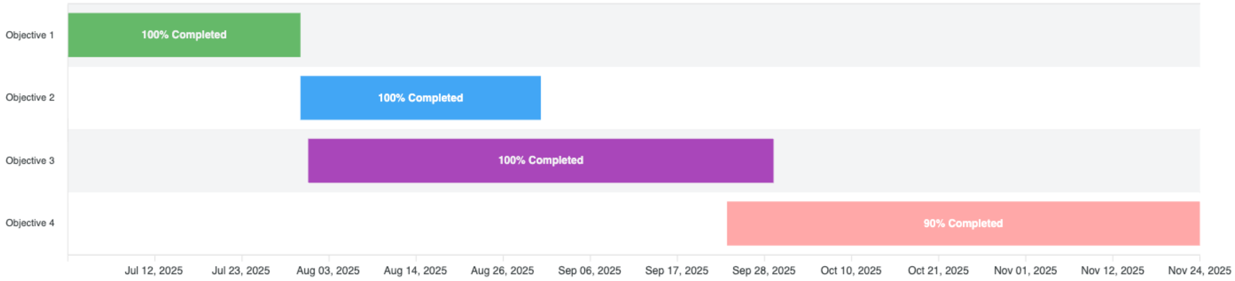

Figure 6: Gantt Chart

Appendix C: Risk and Issue Management

This log tracks the most significant risks identified at the project's outset and the *actual issues* (which became findings) encountered during execution.

D	Type	Description	Mitigation / Resolution
R-01	Risk	**Data Availability:** Inability to access real-world insider data due to privacy laws.	**Mitigated:** Used the CMU-CERT dataset, which is the academic standard for this domain and provides a robust,

			labelled ground truth for validation.
R-02	Risk	**High False Positives:** Models overwhelming a SOC with benign alerts (The Accuracy Paradox).	**Mitigated:** This became the central theme of the research. We addressed it by (1) engineering Peer-Relative Z-scores, (2) implementing the LSTM as a "Precision Filter," and (3) optimizing the ensemble weights to suppress static noise.
R-03	Risk	**Ethical/Privacy Concerns:** Project could be perceived as "employee surveillance."	**Mitigated:** Addressed directly in "Ethical Considerations" (Section 1.9). The system is designed to profile *events*, not people, and uses role-based baselines to ensure fair comparison.
R-04	Risk	**Hardware Limitations:** The 183GB dataset could exceed available RAM/Storage.	**Realized (Issue I-04):** This risk was fully realized. **Mitigation:** Re-scoped the project to a representative subset (r1, r2, r3.1) and re-architected the entire pipeline to use memory-safe

			streaming algorithms.
I-01	Issue	**CRITICAL: Ingestion Bias**	**Resolved (V1):** Initial experiments showed sampling destroyed the threat signal. Traced to config.py. USE_SAMPLING was permanently disabled to enforce full-volume processing.
I-02	Issue	**CRITICAL: Data Unlabeled (Integrity Failure)**	**Resolved (V1/V2):** Early models failed to find insiders because dataset IDs (r3.1) didn't match the answer key (3.1-1). Fixed with a custom normalization engine in label_insiders_v2.py.
I-03	Issue	**CRITICAL: Empty Test Set (Methodology Failure)**	**Resolved (V1):** Standard train_test_split randomized the single insider out of the test set. Re-architected the entire framework to use a "Train-on-Normal, Test-on-Mixed" temporal split.

I-04	Issue	**CRITICAL: OOM (Out of Memory) Crash**	**Resolved (V2):** The V1 pipeline crashed on 24GB RAM. The entire ETL process (data_preprocessing_v2.py) was rewritten as a multi-pass, chunked streaming system.
I-05	Issue	**CRITICAL: The "Sliding Window" Risk**	**Resolved (V2 Findings):** We feared LSTMs would fail on sparse logs. The Grid Search proved the opposite: the LSTM was the strongest model (Weight: 0.8), effectively resolving this theoretical issue empirically.
I-06	Issue	**CRITICAL: "Multiclass" Error in Evaluation**	**Resolved (V2):** The aggregation strategy initially created non-binary labels (e.g., label=2). Fixed by forcing a strict binary conversion (.astype(int)) in the final evaluation script.

Appendix D: Project Artifacts

Git Repository Link: https://github.com/BipinRimal314/python_files_thesis