Insider Threat Detection Using Time Series Foundation Models: A Transformer-Based Behavioral Analytics Approach

by Joseph A. Castro

B.S. in Computer Science, April 2020, California College San Diego
M.S. in Cybersecurity Engineering, May 2022, University of San Diego

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

January 9, 2026

Praxis directed by

Artem Stepanov
Professorial Lecturer of Engineering Management and Systems Engineering

The School of Engineering and Applied Science of The George Washington University certifies that Joseph A. Castro has passed the Final Examination for the degree of Doctor of Engineering as of January 9, 2026. This is the final and approved form of the Praxis.

Insider Threat Detection Using Time Series Foundation Models: A Transformer-Based Behavioral Analytics Approach

Joseph A. Castro

Praxis Research Committee:

Artem Stepanov, Professorial Lecturer in Engineering and Applied Science, Praxis Director

John Fossaceca, Adjunct Professor of Engineering and Applied Science, Praxis Chair

John Dargan, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Joseph A. Castro
All rights reserved

Dedication

This endeavor would not have been possible without the support of my family. I want to dedicate this achievement to them, for their unwavering physical and emotional support throughout the duration of this program. To my wife, Amanda, who manages to hold our family together in the most challenging times—you made this journey possible. To my children, Liam, Milena, and Amelia—thank you for being a light of motivation during this endeavor that kept me moving forward and for allowing me to step away temporarily to pursue this goal. I hope this achievement shows you guys that anything is possible with willpower. Lastly, to my parents, who made a courageous decision to migrate to the United States in search of a better life, thank you. I hope this educational accomplishment serves as one of the many gifts I can give in return.

Acknowledgements

To the SEAS faculty and Dr. Stepanov, thank you for providing a rigorous program that challenged us to grow beyond our limits both professionally and intellectually. I couldn't have made it to the finish line without your guidance and commitment.

This program expanded my understanding of the key cybersecurity concepts, not only from a non-technical perspective, but also deepened my technical abilities across different domains of cybersecurity. I am sincerely grateful for the time and effort the faculty put into us and I hope this research comes as a thank you!

I will miss the camaraderie, mentorship, and sense of community we built over the last 2 years, but I am also glad to have reached a state of academic and professional fulfillment. Lastly, I am truly excited to apply the knowledge, discipline, and critical thinking developed during this program and help overcome the future cybersecurity challenges. To the SEAS faculty and Dr. Stepanov, thank you for providing a rigorous program that challenged us to grow beyond our limits both professionally and intellectually. Your guidance and commitment have been instrumental in helping me reach this milestone.

Abstract of Praxis

Insider Threat Detection Using Time Series Foundation Models: A Transformer-Based Behavioral Analytics Approach

Insider threats remain one of the most difficult challenges in cybersecurity. Unlike external attacks, insider threats operate within your environment and are not faced by the external security challenges an external entity would typically face. In addition, an insider threat could be at any hierarchical level of your environment. From an end user to your CEO. This praxis aims to highlight the deficiencies of current traditional security controls, which are often simply rule or anomaly based. We dive deep into current emerging transformer-based behavioral detection models, with a deep focus on Time Series Foundation Models (TSFMs), specifically, the Patch Time Series Transformer (PatchTST) architecture.

The Carnegie Mellon University (CMU) Computer Emergency Response Team (CERT) r4.2 dataset was used to mimic realistic enterprise user behaviors across multiple (logons, file operations, device usage, email, and web activity). User-day activity logs were aggregated and converted to multivariate time series tensors. Temporal and statistical features were engineered (e.g., rolling z-scores, department-based deviations, and interactions across modalities) to capture both short-term and long-term behavioral characteristics. Dimension reduction (100% variance retained) was accomplished with Principal Component Analysis for stability and improved training efficiency.

Due to the high levels of class imbalance in CERT 4.2, a supervised PatchTST model was trained using weighted sampling, focal loss, and negative-only pretraining to aid mitigation of severe class imbalance, which, in our case, insider threat activity

encompasses 2% of the entire dataset. Model performance was evaluated across multiple operational scenarios. A default, precision-oriented, fixed daily alert budget, and calibrated probabilities in attempts to emulate a real Security Operations Center (SOC).

The results demonstrated a strong and balanced detection capability. The model achieved the following scores: PR-AUC = 0.97, ROC-AUC = 0.999, and up to 93-97% in recall, while maintaining 83-93% in precision, depending on which operational mode was at test. In attempts to simulate a real-world SOC scenario, we test the model accuracy with a 160-alerts-per-day constraint, which resulted in an F1-score of 91.6%, demonstrating both technical and operational practicality. Compared to prior research, such as auto encoders, Isolation Forest, One-Class SVM, and UBS-Transformer models, Patch TST managed to achieve competitive and in some cases superior results without the need for complex user-session reconstruction.

The research demonstrates that transformer-based time-series modeling can detect insider threats through their behavioral, temporal patterns and account for realistic workloads for analysts. Through showing scalable preprocessing, supervised learning on high-dimensional behavioral tensors, and evaluation modeled after SOC capacity, this framework creates a reproducible model for future insider-threat research. Clearly, the results validate that TSFMs (i.e. PatchTST) can enhance detection accuracy, interpretability, and trust—bridging the gap between the advanced research into Artificial Intelligence (AI) and operational deployment of cybersecurity.

Table of Contents

Dedication ..... **iv**

Acknowledgements ..... **v**

Abstract of Praxis ..... **vi**

List of Figures ..... **xi**

List of Tables ..... **xiii**

List of Acronyms ..... **xiv**

Chapter 1: Introduction **1**

1.1 Background 1

1.2 Research Motivation 3

1.3 Problem Statement 5

1.4 Thesis Statement 6

1.5 Research Objectives 7

1.6 Research Questions and Hypotheses 7

1.7 Scope of Research 9

1.8 Research Limitations 9

1.9 Organization of Praxis 10

Chapter 2: Literature Review **12**

2.1 Foundation of Time Series Foundation Models (TSFMs) 12

2.1.1 Encoder Architecture 16

2.1.2 Psychological and Behavioral Foundations 20

2.2 Insider Threat Detection: Foundations and Frameworks 24

2.2.1 Benchmark Datasets for Insider Threat Detection 27

2.2.2 Classical Machine Learning and Anomaly Detection Approaches 31
2.2.3 Evolution of Insider Threat Modeling 36
2.2.4 Transformer-Based Insider Threat Research 40
2.2.5 Comparative Analysis of Detection Paradigms 43
2.3 Research Gaps and Motivation for the Current Study 47
Chapter 3: Methodology.....51
3.1 Introduction.....51
3.2 System Specification.....54
3.3 Dataset Description and Selection Rationale.....55
3.4 System Architecture.....57
3.4.1 Pipeline Overview.....58
3.4.2 Data Quality Control and Cleansing Procedures 59
3.4.3 Preprocessing and Daily Feature Engineering.....61
3.4.4 Tensor Construction and PCA Integration.....63
3.4.5 Training and Class Imbalance Handling.....65
3.4.6 Operation Interpretation.....67
3.4.7 Summary of Innovations.....68
3.5 Conclusion 71
Chapter 4: Results.....73
4.1 Introduction to Results.....73
4.2 Default Threshold Results (threshold = 0.5).....76
4.3 Precision-Constrained Results (92% Precision) 79
4.4 Budgeted Alerts Results (160 Alerts per Day) 82

4.5 Calibration Results 85
4.6 Collapsed User-Day Results 88
4.7 Comparison to Other Research 91
4.8 Hypothesis Evaluations 95
4.8.1 Hypothesis 1 (H1) 95
4.8.2 Hypothesis 2 (H2) 96
4.8.3 Hypothesis 3 (H3) 96
4.9 Summary of Results 97
Chapter 5: Conclusions 101
5.1 Introduction to the Conclusion 101
5.2 Summary of Findings 102
5.3 Contributions of This Work 105
5.4 Limitations 107
5.5 Future Work 108
5.6 Final Remarks 109
References 112
Appendix A 123

List of Figures

Figure 1: Comparative architectures of RNNs, Transformer Encoders, and Time-Series Foundation Models (TSFMs). 13

Figure 2: PatchTST Model Overview for Long-Term Time-Series Forecasting (Source: Nie et al., 2023) 14

Figure 3: Transformer Backbone Architecture Used in PatchTST (Supervised Mode) (Source: Nie et al., 2023) 17

Figure 4: Evolution of Insider-threat Detection Paradigms 43

Figure 5: Overall Methodological Framework of the Study 54

Figure 6: Overview of This End-to-End Workflow from Multi-source Ingestion Through Classification and Evaluation 59

Figure 7: Data-flow Diagram Showing Transformation from Multi-Source CERT 4.2 Logs to Unified per-User Tensors and Supervised PatchTST Classification Output 65

Figure 8: Summary of Methodological and Architectural Innovations in the Proposed Insider-Threat detection Framework 71

Figure 9: Confusion Matrix Illustrating Model Performance at 0.5 Threshold 77

Figure 10: Confusion Matrix Illustrating Model Performance at 92% Precision 79

Figure 11: Score Distribution Showing Missed Positives Near the Threshold 82

Figure 12: Confusion Matrix Illustrates Model Performance under the 160-alert-per-day Budgeted Configuration 83

Figure 13: Ranked Score Queue (top 160 per day, TP vs FP) 85

Figure 14: Confusion Matrix Illustrating the Calibrated Mode (Threshold = 1.0), Showing Improved Recall but an Increase in False Positive 86

Figure 15: Confusion Matrix for the Collapsed Raw Model (160 alerts per day), Showing Balanced Performance with High Precision and Recall Consistent with SOC Deployment Conditions 89

Figure 16: Confusion Matrix for the Collapsed Calibrated model (160 alerts per day), Showing Slightly Higher False Positives but Strong Recall and Realistic SOC Alignment 90

Figure 17: A visual “Performance Snapshot” Comparing the Four Operating Modes (Default, Precision-Constrained, Budgeted, Calibrated) Side by Side with Their Precision, Recall, and F1 values 100

List of Tables

Table 1: Comparative Analysis of Insider-threat Detection Paradigms 45

Table 2: Summary Statistics of CERT Insider Threat Dataset Releases Used in This Study 57

Table 3: Metrics @ 0.5 (Precision, Recall, F1, Specificity, Accuracy) 79

Table 4: Comparison of Default (0.5) and Precision-constrained (≥ 92%) Operating Thresholds Showing Trade-offs between Detection Accuracy and False-Alert Reduction 81

Table 5: Performance Comparison Across Default, Precision-constrained, and Budgeted Alert Configurations 84

Table 6: Precision-Constrained vs. Calibrated Model Performance Comparison 88

Table 7: Collapsed User-Day Performance Comparison Between Raw and Calibrated Models 91

Table 8: Comparison of Insider-Threat Detection Results Across Peer Studies 92

List of Acronyms

AI	Artificial Intelligence
CERT	Computer Emergency Response Team
CMU	Carnegie Mellon University
IDS	Intrusion Detection System
LSTM	Long Short-Term Memory
ML	Machine Learning
MOC	Motivation–Opportunity–Capability
PatchTST	Patch Time-Series Transformer
PCA	Principal Component Analysis
PR-AUC	Precision–Recall Area Under the Curve
RNN	Recurrent Neural Networks
SIEM	Security Information and Event Management
SOC / SOCs	Security Operations Center(s)
SVM	Support Vector Machine
TSFM / TSFMs	Time Series Foundation Model(s)

Chapter 1: Introduction

1.1 Background

Machine and deep learning have revolutionized the digital world in the last decade and could be compared to the digital boom of the early 2000's. Cybersecurity Ventures, a leading researcher and publisher focused on the global cyber economy, providing facts, figures, and statistics about the cybersecurity industry has predicted that global cybercrime costs will continue to rise by 15 percent year after year over the next five years, reaching $10.5 trillion annually by 2025. Advance learning algorithms are giving organizations the ability to quickly and effectively sift through large amounts of behavioral data, identify anomalies, and anticipate potential insider and external attacks. According to the *Cost of a Breach Report 2024* by IBM (2024), organizations that have used Artificial Intelligence (AI) and automation in their cybersecurity program saved an average of $2.22 million per breach when compared to organizations that did not. As organizations security incidents remain on the rise, speed and accuracy have become the defining priorities of modern threat detection systems.

Reports such as the 2025 Verizon Data Breach Investigations Report explain that 22% of breachers are attributed to insiders. While the Ponemon Institute's (2024) report puts a $16.2 million dollar cost on insider risk, a 20% increase from 2022. In effort to combat the rising risks and costs. Regulatory organizations such as GDPR, CMMC, NIST have enforced regulatory sanctions for failure to implement insider threat risk controls. GDPR, popular in the European Union, has some of the harshest sanctions for organizations who fail to meet continuous, risk-based protection of personal data. It can fine penalties of $20 million or 4 % of the organization's global turnover. The

Cybersecurity Maturity Model Certification (CMMC 2.0), popular across the DoD, mandates insider threat auditing controls via access control (AC), and the audit and accountability (AU) control families. Failure to meet these controls could result in revocation of CMMC certification. NIST SP 800-53 Rev. 5 and NIST SP 800-171 provide organizations controls such as AU-6 (Audit Review, Analysis, and Reporting) and PM-12 (Insider Threat Program). Automated behavioral-analytics detections systems meet the regulatory requirement for audits and will provide, on demand, continuous, verifiable evidence of user activity monitoring, documentation of anomalies, and policy enforcement ultimately reducing regulatory exposure for the organization, while enhancing organizational cyber governance.

Machine learning (ML) and AI are reshaping how we manage various cybersecurity domains, in particular malware detection, phishing emails, and threat intelligence. However, insider threat detection remains a relatively unexplored domain of security. Insider threats are often categorized as malicious or negligent individuals within their organization. What makes insider threats a challenging threat is their internal access rights and insider knowledge, making them both inherently dangerous and difficult to detect. Small behavioral changes are often the precursor to an incident occurring, but traditional monitoring capabilities are often focused on rule-based actions and not truly architected nor designed to process massive multivariate amounts of data.

Traditional insider threat detection tools have historically relied on rule-based and signature-driven tools. Tools like antivirus, intrusion detection systems, access monitoring scanners, and data loss prevention are very common in today's market. Despite their effectiveness at finding known attacks through the reports of their

identified patterns and rules, they have difficulty identifying users' patterns of behavior, both contextually and temporally, over time. It is contextual behavioral changes over time that typically precede the insider incidents. Thresholds and signatures are static rules that cannot measure gradual increments of behavioral change like increased frequency of file access, off-hours log-in activities, and more subtle changes, such as random usage of peripheral devices.

To address these limitations, this praxis puts forth a novel strategy leveraging Time Series Foundation Models (TSFMs), specifically PatchTST. TSFMs present an attractive alternative to manual data engineering techniques. Past solutions have relied upon fixed rules or well known malicious signatures. In the other hand, PatchTST learns from multivariate, **timestamped** behavioral data such as logins, file access, device connections, and email activity. Allowing it to detect developing behavioral anomalies in a predictive manner rather than a reactive solution. Additionally, PatchTST is fundamentally sequential and adaptive, its architecture is well designed to model the constant dynamical changes of human behavior, this aids the model to recognize earlier signs of insider threat risk indicators. This research aims to represent a paradigm shift in how insider threats can be detected, understood, and mitigated in real-world enterprise environments.

1.2 Research Motivation

In recent years, insider threats have gained increased notoriety due to a number of high-profile incidents involving the unauthorized disclosure of classified or sensitive information. Insiders pose unique and complex threats to organizations due to the inconspicuous and stealthy nature of their malicious intent. An insider threat, unlike the

external threat, originates from individuals who have rightful access to organizational data, creating a severe threat to the confidentiality, integrity, and availability triad in its entirety. Insider threat incidents can pose a significant risk to revenue, operational availability, and sensitive data. In the critical industries of defense, health care, and finance, insider threats can result in even greater ramifications, such as threatened national security and public safety.

Organizations generate vast amounts of end-user behavioral data, creating an opportunity to leverage the data for their own insider threat detection program. TSMFs are capable of evaluating user actions in a sequential fashion, uncovering predictive patterns that evolve over time. Such patterns are useful in discovering insiders who subtly deviate from their normal behavioral baselines, such as actions that will not trigger traditional rule-based alerts or appear anomalous when viewed in a particular slice of time. The motivation behind this research emerges from the need to develop an intelligent detection framework that is capable of accurately classifying insider threat activity prior to its fruition.

Most of this author's professional experience has involved sensitive defense programs, where the measurement, protection, and dissemination of mission-critical data and information are always a prominent concern. In this environment, a subtle behavioral change can be the precursor to very serious consequences, and yet the traditional security detection systems fail to detect a behavioral change. This knowledge and experience motivated the effort behind this research in hopes of allowing the detection of insider threats much earlier, not by static alarms or rule-bound alerts, but by the detection of

subtle behavioral changes, of which the continual measurement of those changes will, over time, show a guarantee of detection of insiders.

The typical forms of insider threat detection, in the form of detection of deviations from established patterns or the setting of fixed limits and rules for limits, are complaint driven- (as, for example, excessively numerous file copies or excessive logons, or logins during off-hours). The high false-positive rate of these techniques, together with their inherent lack of self-awareness and meaningless detection of behavioral deviations, has contributed to their general ineffectiveness in the detection of behavioral factors preceding insider attacks.

This praxis seeks to make some contribution to the limitations of insider threat detection mechanism. Time Series Foundation Models (TSFM's) will be utilized to analyze behavioral patterns as they manifest themselves with the lapse of time. By impacting a time-series model of user activity, it is possible to sufficiently understand its typical behavioral rhythms and detect behavioral anomalies, which reasonably indicate an insider threat. The time-dependent impact of this signifies better means of detection, as it also escapes the necessity of the exhaustive and regularly inefficient urgency of the continuing requirement for data pre-processing or definite rules for accuracy, but gives scalable or massive possibilities for the detection of insider threat early and reliably.

1.3 Problem Statement

Current insider threat detection systems are often reactive, opaque, and dynamically limited. They fail to identify the subtle, discreet, and covert signs of insider threats. There is a great need for detection systems that are dynamically self-adaptive, capable of learning from time-based behavioral data as it unfolds in real time.

1.4 Thesis Statement

A PatchTST-based TSFM will enhance insider threat detection by identifying behavioral anomalies within temporal sequences of user behavior, improving detection accuracy, scalability, and analyst trust in high-security environments.

This paper introduces a behavioral analytics model utilizing the PatchTST architecture to examine user activity over time. It integrates multiple domains of behavior information from the CERT 4.2 Insider Threat Libraries: device usage, file use, email activity, web requests, and logon events in order to generate per-user daily activity profiles. These activity profiles encapsulate the contextual and temporal dimensions necessary for distinguishing expected from anomalous user behavior.

Temporal, statistical, and organizational interaction vectors are generated to characterize deviations from the behavioral baseline of each user. While minimizing the overall dimensionality of the acquired temporal information stream, the model arranges these vectors into temporal tensor designations of user behavior data with respect to activity on previous days, which in turn are processed through the PatchTST transformer model to learn and differentiate anomalous from expected behavior. The model is optimized in order to enhance overall sensitivity and deal with any imbalanced data by utilizing optimizations such as dimensionality reduction via PCA, weighted sampling, and focal loss.

Evaluation of PatchTST is conducted across a range of operational thresholds: default, precision-constrained, alert-budgeted, and calibrated to reflect realistic decision processes in current-day Security Operations Centers (SOCs).The predictions are aggregated on a per-user, per-day basis to more accurately reflect real-world alert

structures. Results show that neural network–based time series modeling can provide an effective means of detecting the evolving architecture of insider threat behavior, giving accurate, scalable, and operationally representative capability for detection.

1.5 Research Objectives

The research objectives are to design, develop, and evaluate a transformer-based behavioral analytics framework that models user behavior over time to accurately classify and detect insider threats.

The sub-goals of this study are:

1. Carry out an exhaustive literature review on insider threat detection, time-series modeling, and behavioral analytics approaches.
2. Preprocess user activity logs (device, file, email, web, and logon) from the CERT 4.2 dataset into a standardized time-series format.
3. Engineer temporal, statistical, and interaction-based features to enable the detection of deviations from normal user behavior.
4. Build and train a PatchTST-based time-series classification model to detect anomalous vs normal user behavior in a class-imbalanced setting.
5. Evaluate the model in a variety of operational formats—default, precision constrained, alert-budgeted, and calibrated thresholds—to yield models applicable in real world SOC decision-making settings.

1.6 Research Questions and Hypotheses

Research Questions:

RQ1: Is a transformer-based Time Series Foundation Model (PatchTST) capable of detecting insider threats by learning behavioral patterns over time and from multi-source organizational activities logs?

RQ2: To what degree of accuracy will a PatchTST model correctly identify insider threat activity when trained and evaluated on pooled behavioral stream data from the CERT 4.2 dataset?

RQ3: How will feature engineering strategies such as rolling z-scores, department-level z-score interactions, and PCA-based dimensionality reduction affect the overall model performance?

Hypotheses:

H1: Time Series Foundation Models (TSFMs), such as PatchTST, can reliably determine insider threat type by detecting unique anomalistic behavioral profiles over extended periods of time. When appropriate exposure to CERT 4.2 data set is afforded, the model can achieve greater than 90% precision-recall area under the curve (PR-AUC) and greater than 95% recall.

H2: PatchTST-based TSFM will outperform traditional insider threat detection baselines (rule-based detection systems, Isolation Forest, and One-Class SVMs) with respect to any metric of precision, recall, F1 scores, robust qualitatively under matching evaluation circumstances.

H3: Introducing time, statistical and departmental behavioral feature sets and multiple actionable thresholds (default, precision-bound, alert-bound, and calibrated) will develop pursuit of operational validity of the model by balancing detection accuracy with manageable alert quantities directional or

otherwise along threshold of capacity limits of the Security Operations Center (SOC).

1.7 Scope of Research

This research explores the application of time-series modeling techniques on user activity log data for the purpose of characterizing insider threat behavior. The data set used is from CERT r4.2 from the CERT Insider Threat libraries, which is a publicly available database that holds synthetic behavioral data such as logon, file access, email, web, psychometrics, organizational directory, and device activity data. The research emphasizes the temporal and contextual dimensions of user behavior, based on patterns of activity that evolve over time. Although our evaluations are conducted on synthetic data, CERT provides some of the foundational insider threat benchmark datasets currently out there. Our research scope aims to demonstrate how modeled TSMFs like PatchTST could be used for proactive detection of insider threats behavior in sensitive environments.

1.8 Research Limitations

The research acknowledges the following three key limitations:

- This research utilizes a reputable insider threat dataset (CERT 4.2), which has been used for past research purposes, but due to its synthetic nature, it may not represent true realism composed of diversity and unpredictability that real organizations generate.
- The research only considers structured behaviors and activity-based features, and does not utilize multi-modal data sources (e.g., audio, biometric, unstructured communication content, etc.) which might give user context.

- The models are evaluated in an offline simulated environment. While the results demonstrated promising detection performance, deploying these models in actual operational security environments will require additional system infrastructure, as well as optimization & validation on live network operations.

1.9 Organization of Praxis

The praxis is structured into five chapters, each of which addresses an important component of the research process, including:

Chapter 1: Establishes the study through a description of the background, goals, problem statement, thesis statement, research aims, research questions, hypotheses, limitations, scope, and structure of the praxis.

Chapter 2: Dives deep into a comprehensive review of existing insider threat detection research, with an emphasis on insider threats, machine learning, time-series modeling, transformer-based approaches, and behavioral analytics techniques relevant to insider threat and cybersecurity.

Chapter 3: describes the methodology behind the PatchTST architectural framework, which includes: data preparation, feature engineering, time-series behavioral modeling, dimensionality reduction, and development and training of a PatchTST-based model for anomaly detection.

Chapter 4: Decompresses and explains the PatchTST model evaluation results based on different operational thresholds, including default, precision-constrained, alert-budgeted, calibrated, per-user-day analyses, with appropriate performance metrics displays and in-depth interpretation.

Chapter 5: Discusses the implications of findings, summarizes the important contributions of the research, and makes recommendations for future work on advanced time-series applied methodologies to insider threat detection.

The format of the study provides a natural order from conceptualization through methodology and model evaluation to task analytic interpretation, ending with a sufficiently data-based argument concerning how transformer-based time-series modeling can use structured behavioral analytics for improvement in insider threat detection.

Chapter 2: Literature Review

2.1 Foundation of Time Series Foundation Models (TSFMs)

Time-Series Foundation Models (TSFMs) represent a new breed of deep-learning architecture designed specifically for the analysis of sequential/time-series structure in multivariate time-series data. Traditional sequential models like Recurrent Neural Networks (RNN) or Long Short-Term Memory (LSTM) neural networks perform temporal learning through information being sent through loops recurrently. As Sherstinsky (2023, p. 4) explains, *"*Even though RNNs are expressed as difference equations, differential equations have been indispensable for modeling neural networks and for gaining an understanding of their dynamics.*"* Although these models have a strong theoretical base, both RNN's and LSTMS' are limited by their vanishing gradients, and the higher cost of computation when dealing with longer sequence lengths. As Johnson and Khoshgoftaar (2019, p. 177) note, "deep neural networks with various data-sampling techniques can determine the efficacy of deep learning in addressing class imbalance," underscoring the need for architectures and training regimes that remain stable when minority events are rare. The transformer architecture overcomes recurrence-related limits by replacing loops with self-attention that evaluate relationships among all elements of a sequence simultaneously (Vaswani et al., 2017).

The original format of the transformer was extremely successful with language translation and visual tasks; however, it is not ideal for use with temporally arranged data. The global self-attention mechanism treats each token as having an equal relationship to every other token, regardless of their natural position or frequency in a sequence (Wu et al., 2021). As Nie et al. (2023, p. 1) describe, *"segmentation of time series into subseries-*

level patches which are served as input tokens to Transformer; … channel-independence where each channel contains a single univariate time series that shares the same embedding and Transformer weights across all the series." This design introduces stronger inductive bias for temporal continuity—something standard Transformers originally lacked—and directly motivates the development of time-series-specific transformer architectures, known as Time Series Foundation Models (TSFMs).

The three primary architectural models that are compared in Figure 1 - Recurrent Neural Network (RNN) models; Transformer Encoded models; and Time-Series Foundation Model (TSFM) models - represent a paradigm shift in the direction of sequence modeling from RNN models through the use of attention and patching in temporal models. The transition represents an advancement in parallelization, scalability, and modeling of temporal dependency as we progress from one generation of models to another.

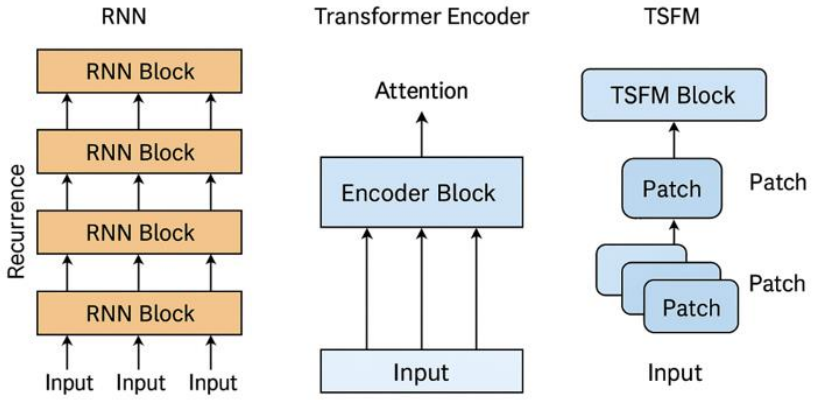
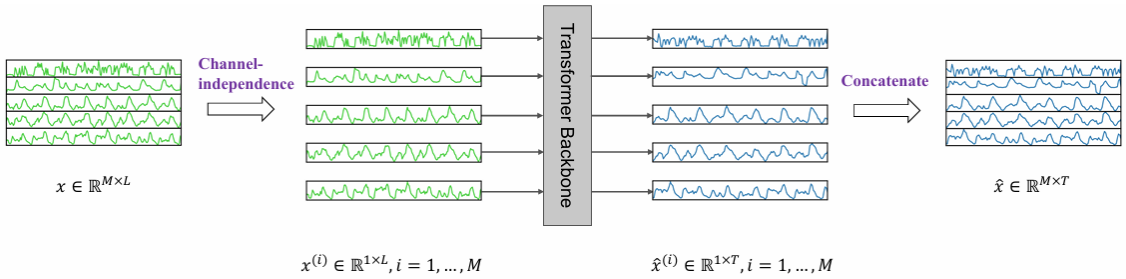


Figure 1: Comparative architectures of RNNs, Transformer Encoders, and Time-Series Foundation Models (TSFMs).

Patch-based Temporal Learning

PatchTST is based on an idea similar to Vision Transformers that divides the continuous temporal signals into small segments that are not overlapping, or *patches*.

PatchTST treats every patch as a token, allowing for parallel computing but working with the temporality of local events. The design is scalable with the fact it allows for efficient attention over thousands of time steps, which is important in the behavioral logs of insider-threat detection, literally allowing for events that can reach months or even years. The patch-based system allows for less information to get through the windows, allowing for better generalization of unseen behavior.



(a) PatchTST Model Overview

Figure 2: PatchTST Model Overview for Long-Term Time-Series Forecasting (Source: Nie et al., 2023)

Frequency-domain Modeling and Temporal Blocks

TimesNet (Wu et al., 2023) complements PatchTST by learning patterns in the frequency domain via the use of stacked *TimesBlocks*. Each TimeBlock extracts multi-scale temporal features via 2D convolution over reshaped time-series matrices. *"*We extend the analysis of temporal variations into the 2D space by transforming the 1D time series into a set of 2D tensors based on multiple periods*"* (Wu et al., 2023, p. 1). This block architecture can simultaneously capture both periodic and irregular patterns without any explicit recurrence. The use of temporal 2D variation allows the model to capture dynamic dependencies across diverse time resolutions, which is an important

capability for the information present in cybersecurity telemetry, which has somewhat regular patterns of daily and weekly nature but interwind with unusual anomalies.

Decomposition and Hybrid Foundations

Recent types of TSFMs have included decomposition-based methods to achieve greater separation of the trend and seasonal terms, such as DLinear, which applies linear forecasts to separate the long-term trend from its short-term cyclical trends (Zeng et al., 2023), and demand-enhanced models, which reduce attention burden and improve forecast accuracy. These hybrid models continue to show the convergence of the field toward methodologies that provide accuracy, interpretability, and computational efficiency.

Self-supervised and Foundation-Scale Extensions

Recently published work, Chronos (Ansari et al., 2024) and SimMTM (Dong et al., 2023) has taken TSFM research towards the foundation model paradigm based on self-supervised learning through large heterogeneous time-series corpora. As Gruetzmacher and Paradice (2021, p. 2) explain, "these models can be initially trained in an unsupervised fashion on very large corpora and then later fine-tuned using an organization's labeled, task-specific data so that they outperform existing text mining techniques on a variety of tasks specific to the organization's needs." Chronos has introduced masked-patch predictions and curriculum-based learning providing a methodology for robust generalization, despite sparse labels, in alignment with some emerging cybersecurity needs where insider-threat labels are sparse and behaviors are mutable.

Application for Behavioral Analytics

TSFMs are uniquely qualified for insider-threat detection, since they automatically learn both short-term and long-term dependencies across behavioral modalities of logins, file access, network requests, and device activity, without the prior design of features. If context can be modeled across time instead of events in time, it may be possible to detect gradual change in behavior (i.e., prior to the harmful acts). Thus, as Nie et al. (2023, p. 5) note, "patch-based transformers provide an efficient means of capturing long-horizon dependencies crucial for temporal anomaly detection".

Operational Advantages and Challenges

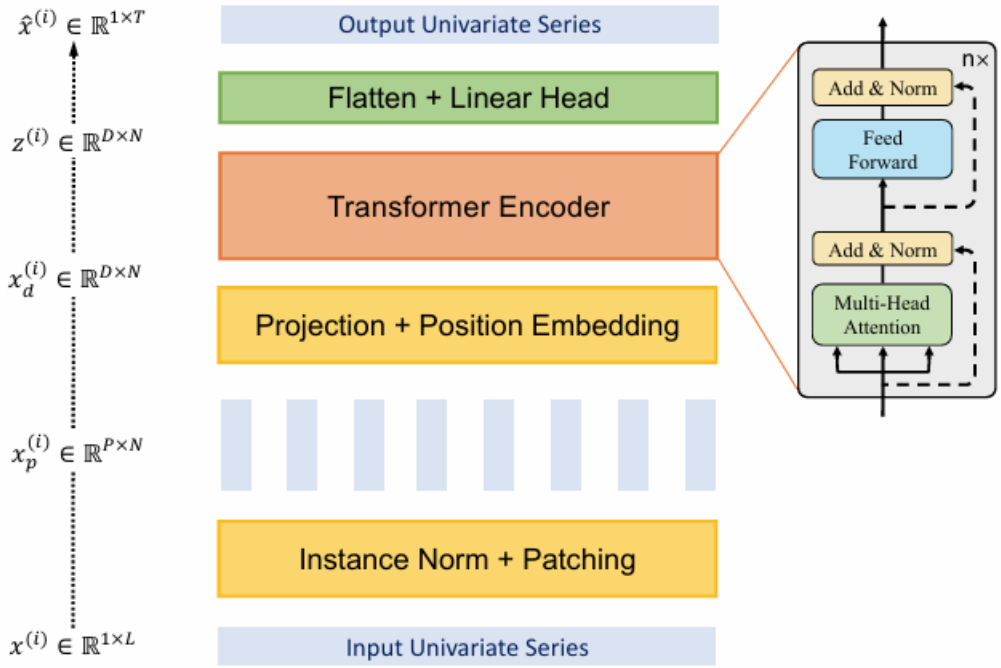
The operational advantages of deploying TSFMs include parallelizable training, suitability for transfer-learning pipelines, and usability in analytic architectures previously developed with Splunk or Elastic SIEM. The remaining challenges appear to be involved in computational overhead, and explainability. Attention-heat-map visualization and saliency analyses work to improve trust of analysts by revealing the time periods, or features, that had the greatest influence on predictions (Wu et al., 2021).

In summary, TSFMs combine transformer efficiency with temporal specificity, producing models capable of learning rich sequential structures across extensive datasets. They represent a significant advancement over recurrent or statistical baselines and provide the methodological foundation for applying transformer technology to behavioral cybersecurity analytics.

2.1.1 Encoder Architecture

TSFMs maintain the basic overall architecture of the transformer encoder originally introduced by Vaswani et al. (2017) but include substantial modifications to

effectively model dependencies over time. The classical transformer encoder is constructed from stacked layers that apply multi-head self-attention and feedforward transformations. Each encoder layer learns contextual representations by attending to all other locations in a sequence, allowing the model to impute global dependencies via an attentional mechanism with no recurrence. Although notably successful in the natural language processing arena, such a mechanism poses challenges when then emulated directly on time series because of the inherently continuous and ordered nature of temporal sequences (Wu et al., 2021).



(b) Transformer Backbone (Supervised)

Figure 3: Transformer Backbone Architecture Used in PatchTST (Supervised Mode) (Source: Nie et al., 2023)

Input Representation and Patching

In the standard transformer, the input to the encoder is discrete tokens (e.g., words or image patches). In the case of TSFMs, continuous time-series signals must be discretized or patched prior to inputting into the transformer. PatchTST (Nie et al., 2023) addresses this issue through a gold standard introduction of multivariate time series, where *"segmentation of time series into sub-series-level patches which are served as input tokens to Transformer; … channel-independence where each channel contains a single univariate time series that shares the same embedding and Transformer weights across all the series" (p. 1)* Each patch is then projected into a latent space (typically) via a learnable linear embedding layer that is mechanically akin to word embeddings for text. This allows the model to effectively parallelize computations while simultaneously achieving localized time-series information for each patch. As Nie et al. (2023, p. 4) state, "This patch embedding yields a compact yet expressive representation of time segments, preserving both local trend and context information."

TimesNet (Wu et al., 2022) takes a different approach by transforming the time series into two-dimensional matrices and applying convolutional filters to learn temporal-frequency relationships. Instead of sequential attention, TimesNet's architecture builds encoders from TimesBlocks as opposed to utilizing standard attention layers, enabling a mix of pattern-detection and frequency analysis. This model design allows the model to recognize both short-term spikes and long-term trends in behavior which is critical for insider-threat data, where user actions changes can be gradual or instant at times. Wu et al. (2023, p. 8) explain that this two-dimensional encoding "allows the model to capture local periodicity while maintaining global temporal coherence".

Positional Encoding and Temporal Order

Classical transformers utilize sinusoidal positional encoding to encode the ordering of tokens, which would otherwise be lost due to the permutation-invariant characteristic of the attention mechanism (Vaswani et al., 2017). The TSFM alters this architecture to make it have either trainable or convolutional positional parameters that are learned in training, allowing the model to account for domain-specific periodicities. This flexibility allows for functional behavioral analytics in situations where temporal signals such as day-of-week and time-of-day activity have predictive capabilities.

Attention Mechanisms and Efficiency

The self-attention technique of the transformer is quadratic to the number of inputs used, thus making it tremendously computationally inefficient for long time-series logs. TSFMs utilize sparse or hierarchical techniques of attention to help reduce these generalized costs. **FEDformer** (Zhou et al., 2022) introduces a frequency-domain strategy: *"By randomly selecting a fixed number of Fourier components, the proposed model achieves linear computational complexity and memory cost"* (p. 5). Autoformer (Wu et al., 2021) instead relates (by auto-correlations) the self-repeating seasonal patterns in a more efficient way. All these tools preserve the global-awareness property of attention and increase its scalability for extended behavioral data.

Feedforward Networks and Normalization

For the attention layer, each encoder block is then followed by a position wise feed-forward network in a further attempt to encode these representations. In TSFMs, this FFN is either replaced altogether by or complemented with convolutional modules that give locality and allow a more effective extraction of features. The normalization steps

also differ: Vaswani et al. (2017) apply layer normalization after each sublayer, while other models, such as PatchTST and TimesNet, employ pre-normalization, that is normalizing the inputs before the attention mechanisms have a chance to work; thus allowing for a more stabilized gradient flow, and helping to converge on noisy and temporal data.

Architectural Depth and Adaptability

The number of encoder layers in TSFMs varies depending on the use case. PatchTST has demonstrable performance scalability up to 12 or more encoder blocks in part due to its patch-based architecture. TimesNet achieves similar depths to PatchTST through stackable convolutional modules, which retain frequency information. The modularity allows for domain adaptation; models trained on general temporal data can be fine-tuned on accounts of behavioral logs associated with a specific organization with a very small dataset. This is an important aspect in the case of insider-threat adaptation when labeled data is in short supply.

Overall, TSFMs improve upon the original transformer encoder by applying innovative methods for patch representations, temporal embeddings, attention efficiency, and normalizations to the time series data. The improvements allow TSFMs to maintain the strengths of transformers—such as parallelization and modeling long range dependencies that were previously impeded by structural/computational limitations preventing their use in behavioral analytics.

2.1.2 Psychological and Behavioral Foundations

Developing analytical models to detect insider threats relies on understanding the psychological/behavioral factors driving those threats. While machine learning can be

used to detect patterns of user behavior (statistical deviation) many of those statistical deviations are caused by the psychological/behavioral precursors of stress, dissatisfaction, and opportunism. As Harris (2020, p. 12) notes, “Privileged users represent one of the most significant insider threat vectors because of their elevated system access and their ability to manipulate security configurations”, underscoring the heightened risk posed by insiders with advanced access or technical authority. The integration of behavioral science into data-based models enhances the ability to understand the context in which the anomalies occur, resulting in more accurate predictions, and enhanced model interpretability (Maasberg et al., 2020). As Ackerman and Mehrpouyan (2017, p. 1) observe, “the goal of this paper is to predict an insider attack using behavioral, computer and psycho-social risk factors”, emphasizing the need for multi-domain modeling that integrates human, IT, and organizational dimensions.

Theoretical Models of Insider Behavior

The following theoretical models of insider behavior have been proposed: The Critical Pathway to Insider Risk proposed by Shaw and Sellers (2015) describes insider attacks as incremental events accounting for underlying personal factors, stressful life events, observable concerning behaviors, and organizational oversight. Similarly, the Motivation–Opportunity–Capability (MOC) model conceptualizes insider threats as the intersection of personal drivers, enabling circumstances, and skill sets (Nurse et al., 2014). As Nurse et al. (2014, p. 319) state, “the insider threat arises at the intersection of intent, capability and opportunity”. These models highlight that insider incidents are gradual events, providing identifiable behavior over a time scale, which is exactly the kind of time scale that TSFMs are intended to simulate mechanically.

Observable Behavioral Indicators

Research performed at the CERT Division of Carnegie Mellon University categorizes insider incidents into five classes: IT sabotage, intellectual property theft, fraud, espionage, and inadvertent threats. Each class has its own distinct behavior signatures (Silowash et al., 2012). For example, after-hours access to files and use of removable media might increase before voluntary termination of employment in the event of intellectual property theft, while supervisory or performance problems might precede incidents of sabotage. Maasberg et al. (2020, p. 214) found that "indicators that are observable, such as deviant job satisfaction, interpersonal conflict, and disregard for policy often precede technical anomalies".

Psychosocial and Organizational Factors

Apart from individual considerations, the climate or environment of the entity plays an important part in promoting or preventing insider abuse of authority. Research associated with deterrence theory has shown that the perception of certainty and severity of sanctions has a great effect on intentions of abuse of authority (Padayachee, 2016). User training and visible monitoring are strong psychological deterrents that impress upon the individual a sense of accountability (Greitzer and Frincke, 2010). Poor communication, lack of recognition, and limited advancement opportunities can foster disengagement and rationalization of malicious acts (Shaw & Sellers, 2015). These psychosocial conditions create measurable behavioral change, e.g., absenteeism, patterns of collaborations or cooperation, which can be encoded as temporal features into TSFMs.

Human-Centered Feature Engineering

Behavioral theory informs the choice of indicators that are most relevant to insider-threat modeling. Login irregularities, spikes in file transfer amounts, communication sentiment shifts, and irregularities in peer-interaction rates are a few of the common indicators. When TSFMs are trained on sequences that measure one or more of these indicators, the temporal relationships between psychosocial stressors and anomalous behavior are implicitly learned. In contrast to purely statistical models, TSFMs align behavioral patterns with cognitive or motivational precursors, which integrate technical detection with human interpretation.

Ethical and Privacy Considerations

The use of psychological principles in machine learning models creates ethical challenges in the realm of employee privacy and potential biases resulting from the models. As Guan et al. (2020) note, "we propose a Differentially Private Greedy Decision Forest (DPGDF) to build a privacy-preserving decision forest with high utility" (p. 1), exemplifying how data-mining frameworks can balance accuracy with privacy—a principle equally vital for insider-threat analytics. The amount of transparency and oversight required to prevent models from overly targeting a certain group of employees or behaviors is warranted. The appropriate insider-threat program should employ "minimally intrusive, role-appropriate oversight, based on a clearly defined policy and performed under human supervision" (CISA, 2020, p. 22). This ethical orientation creates increased trust in the organization, which facilitates compliance with laws and regulations, as it increases acceptance of the use of AI-driven behavioral analytics.

To summarize, the psychological and behavioral bases of insider threat provide useful context for modeling in time. Frameworks such as MOC and CP are naturally consistent with TSFMs, which detect the progressive unfolding of risk in time. By translating psychosocial indicators into measurable temporal features, TSFMs advance beyond static anomaly detection toward holistic behavioral intelligence that merges human understanding with computational precision.

2.2 Insider Threat Detection: Foundations and Frameworks

Over the last few decades, insider threats detection and preventions mechanisms have had multiple approaches. From 1990-2000s, organizations enforced mechanisms that focused on access-control audits, and threshold-based alerting. Statistical anomaly detection was introduced in the early 2010s, providing probability and clustering detection mechanisms. Shortly after, deep-learning models began to emerge, including recurrent and long short-term memory networks, which improved sequential awareness but struggled with long-sequence memory and computational cost. The latest prospect technologies showing significant improvements are presented via transformer-based and foundation-model architectures that manage long range dependencies through self-attention and efficient parallel computation. The mechanisms evolution over the decades has enabled us to advance the insider threat detection study.

Insider threat incidents are rare to begin with. Organizations are aware of the risk and the possibility of occurrence but it’s not a very common occurrence when compared to day-to-day benign activity. When researching this topic, some similarities were found between insider threat and financial fraud. Both fields require mechanisms that can detect small, high impact anomalies inside complex vast amounts of data. Financial Fraud

Detection has been improved with Graph Analytics, that identify connections between accounts, transactions, and entities to show suspicious relationship links; Sequence modeling, that represents how behaviors occur over time, rather than individual events; and Ensemble Methods, that represent combining several weak detectors into an overall stronger detection decision. The same techniques could be used for Insider Threat Detection by creating a representation of the relationships between users, files, systems, and resource; and also identifying behavioral sequences like access expansion, afterhours activity, or unusual data transfer between consecutive days.

The complexity of insider threat detection originates from two aspects, human behavior and the feasibility to monitor technical activity. As Alsowail and Al-Shehari (2020) note, "the lack of real insider threat datasets is still a challenge" (p. 5), underscoring the persistent difficulty of evaluating detection systems across consistent empirical baselines. Since traditional cybersecurity strategies were constructed for defense against external adversaries, there is no methodology designed for internal actors abusing legitimate credentials. Thus, insider-threat detection advanced across a variety of paradigms—psychological frameworks, machine learning, anomaly detection, and most recently, transformer-based temporal modeling. Within each paradigm, there are unique insights, but there are also common persistent problems of scalability, explainability and generalization.

Early detection systems used rule-based and signature-matching approaches to identify known deviations such as unauthorized transfers of files and violations of policies (Greitzer & Hohimer, 2011). Earlier, Chen, Nyemba, and Malin (2012) introduced the Community Anomaly Detection System (CADS), "an unsupervised

learning framework to detect insider threats based on the access logs of collaborative environments" (p. 332), establishing one of the first relational approaches to insider-threat detection. The systems were very useful for finding known threats, but could not easily identify the gradual, or context-dependent deviations typical of insider scenarios. As organizations began to collect larger behavioral datasets, more classical ML-based approaches to anomaly detection emerged as a data-driven alternative. Techniques such as Support Vector Machines (SVMs), Random Forests, and logistic regression were applied to features extracted from the logon data, emails used, and frequencies of access to files (Chandola et al., 2009). The use of static snapshots became an impediment to the ability to derive the behavior change for which they were designed.

The next methodological phase was the introduction to deep learning, especially applied with RNN and LSTM networks, capable of modeling sequences of user actions over time. Such networks in previous studies have proven advantages concerning the accuracy of anomaly detection performed in a learning manner, by breaking down the necessary relationships in time. The recurrent structures of such systems, however, add to the computational load, revealing that there are losses in the long-range structures of context as the lengths of sequences increase, (Johnson & Khoshgoftaar, 2019).

The introduction of Transformer-based models (Vaswani et al., 2017) revolutionized sequential learning by replacing recurrence with self-attention, enabling models to process all time steps simultaneously. Mohamed Elbasheer's (2023) praxis, *Enhancing Insider Threat Detection Using User-Based Sequencing and Transformer Encoders*, validated that restructuring activity logs into per-user sequences rather than discrete events significantly improves detection accuracy. By focusing on temporal

evolution and contextual embeddings, his framework demonstrated how sequential modeling can better represent insider behaviors compared to static classification.

Building on this ground, the last technologies PatchTST and TimesNet are improving the characteristics of the transformers through specialized architecture that allows them to quickly learn short and long-term dependencies. The following topics describe how TSFM models overcome the core limitations of previous approaches:

1. **Temporal Insensitivity:** Classic machine learning and traditional deep learning algorithms operate in fixed temporal windows and therefore can miss gradual deviations from normality. The TSFM is able to model long-range temporal dependencies over the full-time course of its behavioral templates.
2. **Data Imbalance:** The pretraining and fine-tuning paradigm permits TSFMs to generalize knowledge from a comparatively small number of examples of insider activity.
3. **Explainability and Trust:** Attention visualization, frequency attribution, and modular interpretability enable the output of TSFMs to be understandable to both analysts and auditors (Wu et al., 2021).

The progress in this field has created an opportunity for TSFMs to be at the crossroads of behavioral science and computational analysis to detect and identify potential insider threats as early as possible prior to their development into a full-fledged insider incident.

2.2.1 Benchmark Datasets for Insider Threat Detection

Benchmark data sets are crucial to assess and build insider threat detection models, they enable researchers to create repeatable experimental environments and

utilize it as behavioral data representing real-world organizational conditions. A shortage of real-world insider investigations has forced the reliance on synthetic or anonymized datasets, each one having advantages and disadvantages (Yuan & Wu, 2021).

CERT Insider Threat Dataset

The Carnegie Mellon University CERT Insider Threat Dataset is the most widely used benchmark for insider threat research. As Bartoszewski (2022) notes, "insider threat detection is a growing niche of cyber security. It faces many interesting challenges, including problems with finding real data for experiments, the high point difficulty of anomaly detection, and difficulties encountered when trying to model attacks" (p. ii). There are multiple versions of the data set, e.g., R4.2, R5.2, and R6.2. These data sets simulate corporate environments that have employee activity logs of many types: email, file access, web browser usage, removable media usage, and logons. All the data sets contain both benign activities and malicious user activities that match the insider scenarios (Glasser & Lindauer, 2013) pre-coded in the data sets. For example, R4.2 contains data representing theft-of-intellectual-property and sabotage incidents, while R6.2 data adds multi-user and multi-event complexity.

The CERT dataset allows researchers using it to achieve reproducibility as well as its temporal organization. It allows for both supervised and unsupervised modeling since it provides the ability to indicate malicious intervals at the user-day level. Its synthetic nature and class imbalance, however, have substantial limitations. Malicious records often consist of less than 1 percent of the total number of entries. Yuan & Wu (2021) show that models constructed using CERT overfit to artificial behavior, causing a decrease in generalizability to productive data.

Due to the insider threat's incognito nature, activity in operational environments is relatively low. As Cervellera and Macciò (2018) explain, “the need for extracting a small sample from a large amount of real data, possibly streaming, arises routinely in learning problems, e.g., for storage, to cope with computational limitations, and to obtain good training/test/validation sets” (p. 2886). This principle directly applies to insider-threat datasets, where sampling strategies must retain behavioral fidelity during class-imbalance mitigation. Therefore, some datasets in attempts to mimic such behavior create a significant imbalance, for example, on CERT 4.2, insider threat activity only accounts for 2% of the entire dataset. Numerous studies have examined mitigation strategies for imbalance. Wang and El Saddik (2023) applied a combination of Tomek Links and SMOTE variants to address class imbalance in the CERT dataset, using SMOTE to oversample the minority class and Tomek Links to remove ambiguous boundary samples. Nevertheless, while such techniques improve experimental metrics, they could also induce bias or synthetic noise if the techniques are indiscriminately copied.

Alternative Datasets

Despite the imbalance, CERT remains a dominant source when it comes to the domain of insider threat; its synthetic data is often updated and modernized to keep up with current trends, making it a strong option out of the pool. There are other options out there, though. The Enron Email Corpus, a corporate electronic communications archive, has been embedded into the analyses of insider intent via sentiment and social network analyses. The Two Office Scenarios and Workplace User Interaction Log Set datasets provide more fine-grained chronological behaviors yet have not been as well accepted due to access and annotation quality issues (Bartoszewski et al., 2022).

Hybrid or federated approaches come up as a response to data scarcity. The models can be trained at different organizations via distributed learning signals without requiring the sharing of sensitive data in the federated learning context, preserving the privacy of the firms involved. The semi-synthetic datasets coming from the reinforcement-learning agents can create realistic simulations of some of the complex situations that insiders face and may go beyond the deterministic rules.

Dataset Bias and Realism

The dataset composition shapes the research results. The repeated and deterministic scripts used for CERT data generation could produce unreasonably different malicious behavior, resulting in inflated model complexity/accuracy (Yuan and Wu, 2021). Based on inflated datasets, we identify a deficiency in the current datasets that lack operational realism. For future models to gain improvement, a hybrid approach must be implemented, where audit logs combined with deidentified human data help increase operational realism. (Glasser & Lindauer, 2013). Bartoszewski et al. (2022) propose *bridging the gap* through pragmatic generation, which maintains a statistically valid representation of actual data while retaining ethical restrictions on privacy.

To summarize, synthetic datasets constitute a platform that can be used to investigate ITD models, but they impose structured forces that affect outcome. Future research shall emphasize the necessity of multi-institutional and privacy-preserving datasets that develop dynamically to depict actual insider actions in a more representative fashion. Such datasets can enable the TSFMs to be used to their highest potential as adaptable generalized detection facilities.

2.2.2 Classical Machine Learning and Anomaly Detection Approaches

Before the introduction of deep learning and TSFMs, insider-threat detection had largely relied on classical machine learning (ML) and anomaly detection techniques to provide the first scalable techniques for detecting unauthorized activity within organizational systems. Jyothsna and Prasad (2011, p. 26) determines that, "Network intrusion detection systems (NIDS) are the most efficient way of defending against network-based attacks aimed at computer systems," highlighting how anomaly-based models formed the foundation of modern behavioral deviation detection. These early methods inspired later insider-threat research, which applied similar statistical baselines to user-behavior modeling. As Al-Shehari et al. (2023) explain, "we present an insider threat detection model that addresses the class-imbalance problem at the algorithm level using anomaly-based techniques, as an enhancement over previous approaches" (p. 1). Such techniques have a continuing impact due to the statistical and algorithmic bases that they established, which are the basis for modern detection pipelines (Chandola et al., 2009;).

However, evaluating the performance of these early models posed challenges due to the imbalance between benign and malicious samples. As Hossin and Sulaiman (2015, p. 4) note, "Evaluation metric plays a critical role in achieving the optimal classifier during the classification training. Thus, a selection of suitable evaluation metric is an important key for discriminating and obtaining the optimal classifier." They further emphasize that "accuracy has several weaknesses which are less distinctiveness, less discriminability, less informativeness and bias to majority class data" (p. 1). These limitations later motivated the adoption of balanced metrics such as precision, recall, F1-

score, and AUC in insider-threat research to more fairly assess model performance on imbalanced datasets.

Supervised Learning Methods

Labeled datasets are essentially classified user behavior activity that gets marked as either benign or malicious, such types of data fall under supervised ML techniques. Some popular supervised learning methods are the following: Support Vector Machines (SVMs), Decision Trees, Random Forests, Logistic Regression, and k-Nearest Neighbors. Each method has a distinct advantage or disadvantage over the other, but ultimately, the choice varies depending on interpretability, efficiency, and adaptability.

Previously SVMs have been known to be useful and valued for their ability to handle high-dimensional feature spaces, and their capacity to identify subtle deviations that separate classes. Although a strong foundational technique, it does require an extensive amount of hyperparameter tuning. Traditional machine-learning techniques, however, "are not able to utilize all the data of user behavior because of the complexity, high-dimensionality, sparsity, and heterogeneity of the data" (Al-Mhiqani et al., 2021, p. 573). Another drawback is that it tends to struggle with heavily imbalanced datasets, which insider threat datasets are extremely notorious for (Johnson & Khoshgoftaar, 2019). Decision Trees and Random Forests improved interpretability, allowing an analyst to visualize and contextualize reasoning behind decision paths. These techniques were also prone to overfitting when faced with noisy behavioral data, which synthetic data tends to be. Ensemble-based methods such as Random Forests can capture complex relationships across behavioral features but may lose temporal order when trained on aggregated data.

Another popular technique is KNN classifiers, though they gained popularity for their simplicity, they faced scalability limitations against larger datasets. KNNs architecture forced reliance on distance metrics, which hinder their ability for high-dimensional insider datasets, where meaningful proximity among behavior profiles is difficult to define. Similarly, LR assumed a linear relationship between the inputs and outputs. In insider threat activity, there are often multivariate sources of data that overwhelm LR architectural simplicity.

Unsupervised and Anomaly Detection

Insider threat activity often aims to be scarce, inconspicuous, and subtle; classification becomes a critical feature to consider when evaluating models, and hence why unsupervised techniques also gained popularity towards ITD.

Anomaly-detection models learn patterns of *normal* user behavior and flag deviations as potential threats (Chandola et al., 2009). One-Class SVMs were frequently employed for this purpose. OCSVMs were productive at identifying outliers based on a defined baseline that determined what constitutes typical behavior. However, they were found to be sensitive to parameter selection and often underperformed when user behavior naturally evolved over time.

Another popular technique for unsupervised classification is Hidden Markov Models, which was praised for its ability to represent state transitions in sequential activity data. Login habits or file access frequencies were classified as an independent hidden state that reflected an underlying user behavior pattern. Yet, HMMs suffer from scalability issues: “the model’s complexity increases exponentially with the number of

possible states". In addition, they rely heavily on assumptions about the number of hidden states, which can lead to model bias.

Other techniques revolved around clustering models such as k-Means and DBSCAN, which provided another anomaly-detection avenue. Such techniques partition data subjects into behavioral groups, essentially categorizing groups based on behavior, it determines malicious activity if the trend begins to expand past the expected clusters. While conceptually they appear intuitive, these algorithms assume geometric regularity in feature distributions and therefore struggle to model dynamic behavior over expanded periods of time. For example, a departing employee whose activity changes at a slow pace may not be captured by clustered techniques until enough activity has been aggregated, by which point, serious damage could have been done.

Hybrid and Statistical Techniques

Hybrid systems utilize a combination of multiple techniques to improve detection performance. Some rely on the agreement of supervised and unsupervised learning, while others combine statistical thresholds and machine learning. Guan et al., (2020), utilized a model where agents adapted based on feedback received by adjusting their abnormality threshold through a reinforcement learning mechanism. Li et al. (2022) developed probabilistic models that integrated rhythm analysis as a way to account for user adaptation. These early approaches paved the way for adaptive methodologies that would later be integrated into modern TSFM architectures.

Limitations of Classical Methods

Classical machine learning and anomaly-detection models offer important lessons that guide future research, helping modern approaches improve by addressing the following limitations:

1. **Limited Temporal Awareness:** Traditional algorithms process aggregated features or static windows, losing contextual continuity critical for identifying gradual behavioral shifts.
2. **Label Scarcity and Imbalance:** Insider datasets, such as CERT, are overwhelmingly skewed toward benign activity, leading to biased learning.
3. **Interpretability Trade-offs:** Simpler models are transparent but less accurate; complex ensembles are more powerful but less explainable.
4. **Scalability and Maintenance:** Continuous retraining is required to accommodate organizational or behavioral drift.

As Homoliak et al. (2019, p .21) summarize, “no single detection approach can comprehensively capture the multifaceted nature of insider behavior”. These challenges began the innovation of time-aware modeling architecture. Recurrent networks were some of the pioneering models that eventually led to the introduction of transformers and foundational models that capture long-horizon dependencies.

In summary, classical ML laid the foundation for automated insider-threat detection but lacked the ability to model the nuanced, temporal, and context-rich patterns underlying human behavior. The emergence of deep and transformer-based learning marked a pivotal point for the modern methods we know of today.

2.2.3 Evolution of Insider Threat Modeling

An increase in data variety, computing power, and knowledge about human behavior have given insider threat modeling more research momentum. Previous models concentrated on testing using trial-and-error methods to prove their effectiveness against security measures including audit policies, access control policies, and specific rule-based auditing. Older models used static signatures of recognized malicious actions (such as repeatedly failing login attempts, large numbers of print jobs, email, and unauthorized transfer of files) (Greitzer & Hohimer, 2011). From a return-of-investment standpoint, older mechanisms failed to justify their operational costs, as they required continuous analyst oversight and frequent rule updates. For enterprise environments where SOC teams may be required, meeting those expectations becomes challenging; therefore, they are unable to recognize subtle, context dependent anomalies that develop over time.

From Rule-Based Detection to Behavioral Profiling

Legg et al. (2013) introduced user and role-based profile models that compared real-time behavior against established baselines, allowing the model to detect a deviation even when no explicit static rule was broken. These methods allowed for gradual change oversight. Leggs' model played a transitional point that captured the shift from rule-based detection toward behaviorally adaptive, profile-driven analytics, laying the foundation for some of the common machine learning methods in use today.

Rise of Machine Learning and Feature Engineering

Organizations often generate and retain massive amounts of log data; their data retention periods are often mandated by policy or regulation. Due to enforced mandatory retention periods, and having the data at hand, it created an opportunity for organizations

to apply machine-learning methods to detect insider threats. Algorithms such as Random Forests, Support Vector Machines, and Bayesian classifiers made it possible to automate some of the manual day to day tasks an analyst would often do. Yet, these systems still relied on subject matter experts to determine which pieces of information to consider anomalies, information such as session duration, file type, or the amount of data transferred. Raschka (2020, p. 6) pointed out that "feature selection and evaluation determine model performance more than algorithmic choice," a reminder that human judgment still shaped most of these early systems.

To lessen the dependency on pre-manufactured input data later studies attempted to identify potential clues in multiple sources (i.e., email logs, login history, network usage) with ensemble/hybrid models for enhanced user behavior identification. The blending of techniques is reflective of the developing recognition of both the technical actions involved as well as the surrounding social/organizational context of the insiders' actions.

Temporal Modeling and Sequential Awareness

As Chattopadhyay et al. (2018) note, "the state-of-the-art research on insider threat detection mostly focuses on developing unsupervised behavioral anomaly detection techniques with the objective of finding out anomalousness or abnormal changes in user behavior over time" (p. 660). Their work laid down a temporal methodology to classify insider behaviors, so that subsequent researchers could apply a sequential approach to analyze insider behavior, including using recurrent neural networks (RNN), long short-term memory (LSTM) networks, and more recently, transformers. A major advancement

that followed came with temporal modeling; it recognized that malicious behavior does not typically occur in isolation and typically evolves incrementally over time.

Models like RNN and LSTM can model temporal dependencies (Alshehri, 2022) by recognizing ordered events sequences, thus enabling models to identify slight deviations in behavior, such as an increase in login attempts during non-working hours, or access to files outside a typical range. However, since RNN based models are plagued by vanishing gradient and have limited long-term memory capabilities, they were less effective in analyzing long insider history records.

Transformers and Temporal Context

Transformer architectures, introduced by Vaswani et al. (2017), redefined sequential learning through self-attention, which allows the model to look at all parts of a sequence at once. The ability to represent long-range contextual information without using recurrence stimulated a new wave of research focused on insider threats. Mohamed Elbasheer's (2023) praxis successfully employed transformer encoder technology to sequence user activity and showed that they are significantly better at detecting insider threats than LSTMs when modeling a single users behavior over time as a single time representation.

TSFM technology including PatchTST and TimesNet; provide an example of where the development in the last two decades has evolved from static rules based systems into dynamic systems that can adapt and learn from their own experience and find latent temporal relationships in multiple behavioral variables.

Integration of Behavioral and Psychological Insights

Recent work increasingly fuses behavioral science with computational modeling. Frameworks like the Critical Pathway to Insider Risk (Shaw & Sellers, 2015) and the MOC model (Nurse et al., 2014) have been operationalized through machine-learning pipelines that encode psychosocial signals—such as declining engagement or policy circumvention—into temporal features. This convergence of cognitive modeling and deep learning represents a mature phase in insider-threat research, where human-centric theory informs algorithmic design.

Toward Foundation-Scale Temporal Analytics

Lastly, TSFMs bring together previous innovations by combining multi-objective optimization, domain transfer learning, and cross-domain adaptability. As Bapna et al. (2019) explained, "we propose a simple yet efficient approach for adaptation in NMT. The proposed approach consists of injecting tiny task-specific adapter layers into a pre-trained model" (p. 1), thereby enabling scalable model development with minimal retraining—similar to how TSFMs can be adapted across behavioral domains. TSFMs can be customized to an organization's environment using its own behavioral datasets, removing the need for manually engineered features or incident labeling. As Nie et al. (2023, p. 1) explain, "Patching design naturally has three-fold benefit: local semantic information is retained in the embedding; computation and memory usage of the attention maps are quadratically reduced given the same look-back window; and the model can attend longer history".

In summary, insider-threat modeling has progressed from prescriptive, rule-driven systems to adaptive, temporally aware architectures rooted in behavioral science. This

historical trajectory underscores how TSFMs represent not just a technical advancement but the natural synthesis of decades of research connecting psychology, statistics, and modern AI.

2.2.4 Transformer-Based Insider Threat Research

The adoption of transformer architectures in insider-threat detection marks a major shift from sequential recurrence to self-attention–based representation learning. Initially developed for natural-language processing, transformers demonstrated exceptional ability to capture global dependencies within unstructured text (Vaswani et al., 2017). Their scalability and contextual sensitivity soon inspired adaptation to other modalities such as images (Dosovitskiy et al., 2021), graphs, and time-series data—culminating in their application to behavioral analytics and insider-threat modeling.

From Textual to Behavioral Data

Early cybersecurity applications of transformers focused on parsing unstructured logs, alerts, and messages using pre-trained language models like BERT and RoBERTa. Wang and El Saddik (2023) applied a transformer encoder to the CERT dataset to learn contextual representations of user-activity sequences, achieving higher accuracy than recurrent networks. Their findings underscore how attention mechanisms outperform classical architectures in capturing long-range behavioral dependencies. Similarly, Yuan and Wu (2021, p. 10) observed that transformers “enable parallel modeling of multivariate user actions, preserving sequence structure without recurrence”.

Mohamed Elbasheer's (2023) praxis titled "Enhancing Insider Threat Detection Using User-Based Sequencing and Transformer Encoders" sought an approach to organizing organizational logs into a timeline for each user, essentially creating a lifetime

story for each individual. He found that transformer encoder models using the user timeline provided significant increases in accuracy when compared to traditional machine learning and LSTM baseline performance on both the CERT R4.2 and R6.2 datasets. By focusing on the user as the basis for establishing a temporal context, Elbasheer's research identified an area of intersectionality between behavioral sequencing and transformer modeling that directly supports the methodology of this current research.

Specialized Architectures for Temporal Behavior

Building on these insights, subsequent work refined transformers for time-series behavioral data. PatchTST introduces patch-level segmentation to reduce computational overhead while preserving long-horizon dependencies, and TimesNet applies temporal 2D-variation modeling to capture frequency-domain relations. These models optimize attention efficiency, making transformers practical for continuous behavioral monitoring.

Chronos (Ansari et al., 2024) extends this paradigm through self-supervised masked-prediction training on large heterogeneous time-series corpora, achieving foundation-scale generalization. As Ansari et al. (2024) note, “foundation-scale temporal models achieve generalization across datasets with minimal adaptation cost”. Such self-supervised approaches allow fine-tuning on organization-specific data without extensive labeled examples—a critical advantage in insider-threat research, where confirmed cases are limited.

Explainability and Human Alignment

Transformer-based insider-threat research also emphasizes explainability. Visualization of attention weights enables analysts to identify which time segments or features most influenced anomaly scores. Wu et al. (2021, p. 14) reported that

"decomposition transformers with auto-correlation inherently expose interpretable attention maps that align with temporal dependencies". Complementing this, Hoang et al. (2024, sec. I) demonstrated that "the attention mechanism could rearrange the feature space to get a more efficient embedding for the downstream task," showing that attention layers not only reveal salient temporal relationships but also improve representation quality. Integrating such interpretability with behavioral frameworks like MOC model further aligns computational outputs with psychosocial reasoning (Nurse et al., 2014).

Empirical Performance

Empirical comparisons confirm transformers' superior performance. Elbasheer (2023) documented notable gains in precision and recall when applying user-sequenced transformer encoders to CERT data, outperforming recurrent baselines by modeling long-term user context. Similarly, Wang and El Saddik (2023) observed 12–18% F1-score improvements using transformer attention pooling, showing that attention-based architectures can reduced false positives through transformer-enhanced adversarial learning. Together, these results validate the transformer's ability to model contextual dependencies critical for insider-threat analytics.

Integration into Operational Systems

Although large scale deployments of TSFMs are still in a developing stage. Transformer-based pipelines are being integrated with SIEM systems to enable real-time, adaptive anomaly detection through external analytics or embedded AI modules. As fine-tuned encoders can be used to analyze and process the continuous flow of data from Splunk, Elastic, or Azure Sentinel environments, they enable analysts to contextualize their alerts in a timely manner. Lightweight transformers can be used to process and

analyze the endpoint telemetry data; on the other hand, larger foundation-scale TSFMs can be used to perform centralized risk aggregation and develop patterns.

In summary, the use of transformers represents the current leading-edge in the area of research related to the detection of insider threats. The use of transformers has enabled researchers to combine the benefits of contextualized embeddings and scalable processing and provide explanations of why certain anomalies were detected and others were not. In addition, the use of transformers allows researchers to overcome the sequential limitations of previous models. Building on the work of Elbasheer's (2023) praxis and expanding it through the use of TSFMs such as PatchTST, this study has furthered the field of insider threat detection by providing a comprehensive, temporally based detection framework which utilize behavioral theory and state-of-the-art temporal modeling techniques.

2.2.5 Comparative Analysis of Detection Paradigms

The development of Insider Threat Detection Paradigms is demonstrated in Figure 4 as progressing from rule-based systems & statistical models to classical machine learning, deep learning, and then, most recently, transformers & time-series foundation models (TSFMs).

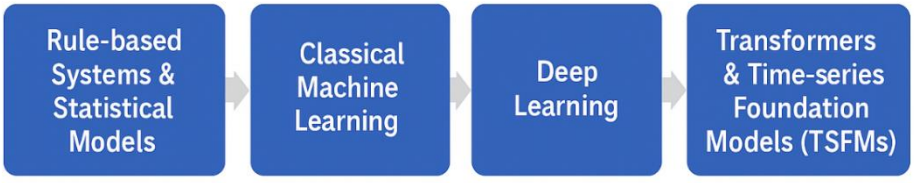


Figure 4: Evolution of Insider-threat Detection Paradigms

Frameworks for detecting insider threats have evolved in a continuing struggle to balance interpretability, flexibility, and timing of when the framework will be able to identify the threat. Rule-based systems that emerged first were clear (transparent) but were rigid (static). Modern transformers and time series models are flexible (adaptable) but opaque (complex). A comparative perspective highlights how each paradigm addresses core analytical challenges—data imbalance, temporal modeling, scalability, and explainability.

Overview of Paradigm Capabilities

A summary comparison of insider threat detection paradigms is shown in Table 1, where each paradigm is described by a representative technique(s), the level of temporal modeling, the level of explainability, the level of scalability, and common limitations. It is evident that the paradigms have evolved from rule-based and statistical systems, through machine learning, deep learning, and now transformers, and that the level of temporal modeling and interpretability has advanced with each successive generation of paradigms.

Table 1: Comparative Analysis of Insider-threat Detection Paradigms

Paradigm	Representative Techniques	Temporal Modeling	Explainability	Scalability / Efficiency	Common Limitations
Rule-Based & Signature Detection	Audit rules, access-control triggers	None – static events	High – logic explicit	High (low compute)	Poor adaptability; cannot detect novel or contextual anomalies
Classical Machine Learning	SVM, DT, RF, k-Means, OCSVM	Limited – fixed windows	Moderate – feature-based	Moderate – retraining costly	Requires handcrafted features; weak temporal context
Deep Learning / RNN Models	RNN, LSTM, Autoencoder	Yes – sequential	Low – hidden states opaque	Moderate – sequential bottlenecks	Vanishing gradients; limited long-term memory
Transformer & TSFM Architectures	Transformer Encoder, PatchTST, TimesNet, Chronos	Strong – long-range dependencies	Moderate – attention maps interpretable	High – parallelizable	Computational overhead; requires data calibration

Analytical Comparison

1. Temporal Awareness

Rule-based and classical ML models analyze data statically, often aggregating features over fixed intervals. Deep learning models introduced sequential processing but retained limited horizon. Transformer-based and TSFM architectures surpass these constraints by modeling relationships among all time points simultaneously, learning both short-term and long-term dependencies.

2. Data Imbalance and Generalization

Traditional ML algorithms underperform when insider events are rare, as in CERT datasets where malicious samples represent <1 %. Techniques like SMOTE alleviate imbalance but risk synthetic bias (Wang & El Saddik, 2023). TSFMs reduce their need for manual data balancing by pretraining on large and diverse time-series datasets that help them generalize from limited insider examples.

3. Explainability and Trust

Deep learning models are generally more accurate than classical models but often lack transparency. Transformers offer a middle ground by using attention maps to show which time periods influenced their decisions. Wu et al. (2022, p. 14) notes that “attention-weight heat maps provide interpretable cues for anomaly attribution”. Operational dashboards can integrate these maps with behavioral indicators to facilitate analyst review.

4. Scalability and Computational Efficiency

Transformers use a form of "matrix" or "parallelized" attention (as opposed to sequential processing as used in deep networks), which allows them to compute many things at once. The module in PatchTST that embeds patches (small parts of an image) and the module in TimesNet that operates on frequencies are examples of how researchers have improved efficiency in computing long sequences. Together these improvements allow for near real time Anomaly Detection in large auditing environments, a critical capability for SIEM integration

5. Behavioral Context Integration

Current transformer pipeline architectures combine psychology models (the MOC Model - Nurse et al. 2014) to provide a basis for detection of human intention. Elbasheer (2024), established that behaviorally sequenced methods capture changes in user behavior, thus linking theoretical computer modeling with empirical behavioral analysis. The TSFMs expand on these ideas by incorporating cross-modal features (i.e. login time, file operation, etc.) into one temporal

representation (Temporal Embeddings) which provides a linkage between machine reasoning (TSFM) and human behavioral theory (i.e. psychological).

Synthesis and Future Direction

Comparatively, TSFMs provide the most comprehensive solution to the insider-threat problem. They inherit transformers' scalability, improve long-range context retention, and embed temporal inductive biases suited for behavioral analytics. While explainability remains a developing challenge, integrated attention visualization and frequency attribution represent tangible progress. Future research should emphasize hybrid systems where interpretable TSFMs operate alongside cognitive-behavioral models, forming human-aligned, adaptive defense ecosystems.

2.3 Research Gaps and Motivation for the Current Study

Though tremendous advances have been made in insider threat detection research over a short period of time there are still many significant gaps between the progress made theoretically and what can be implemented operationally. Insider threats continue to present an enormous problem for detection as they possess characteristics which make them temporally subtle and behaviorally complex and diverse (Wang & El Saddik, 2023; Yuan & Wu, 2021). Although machine learning has improved anomaly detection accuracy significantly and even more so with the advent of transformer-based architectures; their practical application in live environments is constrained by factors such as scale, interpretation, and data realism.

1. Lack of Temporal Depth and Long-Horizon Awareness

Classical and deep-learning methods often capture short-term anomalies but fail to model gradual behavioral shifts that characterize insider threats. LSTM and

GRU networks, for example, degrade when sequence lengths exceed their memory horizon (Johnson & Khoshgoftaar, 2019). Although Elbasheer (2023) demonstrated that user-based sequencing with transformer encoders can retain longer behavioral context, his work also revealed that transformer models trained on limited data remain sensitive to imbalance and noise. Modern TSFMs such as PatchTST and TimesNet address these limitations by incorporating patch-level and frequency-domain processing that enable long-range temporal learning. Yet, empirical applications of TSFMs in insider-threat research remain scarce, indicating an opportunity for focused exploration.

2. Insufficient Handling of Data Imbalance and Realism

Benchmark datasets such as CERT R4.2 and R6.2 have driven methodological innovation but contain inherent constraints—synthetic behaviors, static roles, and severe class imbalance (Glasser & Lindauer, 2013). Oversampling and resampling methods (e.g., SMOTE, ADASYN) mitigate imbalance but can distort behavioral distributions (Wang & El Saddik, 2023). As Yuan and Wu (2021, p. 12) note, "models validated solely on synthetic datasets risk overestimating performance and under-representing real organizational complexity". The lack of diverse, high-fidelity behavioral corpora underscores the need for detection models capable of learning from limited, noisy, and imbalanced data—a key strength of foundation-scale temporal architectures.

3. Explainability and Analyst Trust

Classical models are limited in transparency and explainability. Deep Learning and transformers architecture are intended to maximize accuracy at the cost of

transparency. Wu et al. (2021) found that visualizing transformer attention weights improves interpretability, yet these representations are often opaque to non-technical analysts. As CISA (2020) and the National Insider Threat Task Force (National Counterintelligence and Security Center, 2024) emphasize, operational insider-threat programs require explainable analytics to support accountability and decision-making. This research is primarily focused on leveraging TSFMs for behavioral temporal analytics but can be expanded with technology such as explainable AI. Emerging tools like SHAP, LIME, and counterfactual analysis expand this research's goal by providing feature-level explanations that complement transformer attention maps, offering SOC analysts clearer justification for alerts.

4. Limited Integration of Behavioral and Cognitive Frameworks

Psychological models such as the Critical Pathway to Insider Risk (Shaw & Sellers, 2015) and the MOC model (Nurse et al., 2014) offer qualitative understanding of insider motivation but remain weakly coupled with quantitative AI models. Few implementations operationalize these behavioral constructs within deep or transformer frameworks. The current research seeks to bridge this gap by embedding behavioral precursors—such as privilege use, access breadth, and after-hours activity—into time-series embeddings that align with established psychosocial theory.

5. Gap Summary and Motivation

In summary, existing insider-threat-detection frameworks reveal the following unresolved needs:

1. **Temporal modeling** that captures long-range behavioral dependencies beyond recurrent architectures.
2. **Robust learning** from imbalanced, partially labeled, or synthetic datasets through foundation-scale pretraining.
3. **Integrated interpretability**, allowing security analysts to trace anomaly reasoning.
4. **Behavioral grounding** linking cognitive theory with data-driven modeling.

This research aims to close insider threat detection gaps by using a PatchTST-based time-series pipeline to detect anomalies found in the CERT r4.2 dataset. It uses each user's historical activity as a predictor of potential insider behavior. This method improves scalability by capturing temporal behavioral variability via a combination of both patch-based and frequency domain modeling. The method combines behavioral theory with modern time series analysis and real world validation to create an adaptable and interpretable framework for early identification of potential insider threats in environments where data is at high risk of theft or misuse.

Chapter 3: Methodology

3.1 Introduction

This chapter outlines the methodology employed to develop and evaluate an insider-threat detection model based on TSFMs, specifically the PatchTST architecture. The purpose of this methodology is to demonstrate how to develop transformer-based architectures for detecting subtle and evolving patterns of insider-threat behavior using behavioral multivariate time-series data. Khan et al. (2022, p. 3) states that, "Unlike CNNs, Transformers model long-range dependencies and global context directly through self-attention mechanisms without relying on local receptive fields." The ability of transformers to adapt is an important factor as to why they can generalize with respect to structured data and are suitable for modeling multi-variate user behavioral data. The prior research on anomaly detection relied upon unsupervised reconstruction, or heuristic thresholds, whereas the methodology presented in this study provides a supervised formulation, which allows the model to optimize classification accuracy and be evaluated in multiple operational scenarios assessments.

The methodology stems from a rigidly structured series of steps, each focused on one of the important steps in the research pipeline: (1) the preprocessing of data and aggregation of heterogeneous behavioral logs, (2) the feature engineering and temporal transformation of user activities into standardized tensors, (3) training of supervised models and imbalance aware optimization, and (4) measurement of the predictive capability of the models using realistic, SOC aligned operational thresholds. These stages combined offer a reproducible framework for the application of transformer-based time-series modeling to insider threat behavioral detection.

All modeling and experimentation were performed with CMU CERT Insider Threat Dataset Release 4.2 (CERT 4.2), a synthetic and behaviorally detailed dataset widely used in insider threat research. CERT 4.2 provides labeled, temporally indexed records of user activities performed on the simulated enterprise of about 1,000 users including logons, file operations, email communications, device connections and web browsing. Given its comprehensiveness, clearly defined insider scenarios, and manageable size, it provides a powerful and usable basis for reproducible supervised training and evaluation of deep time-series models.

The architectural process begins with the aggregation of multi-source user behavioral logs into per-user, per-day user activity profiles. Each profile represents a combination of various modalities of user behavior, resulting in a unified view of general system usage over time. As Legg et al. (2017, p. 507) explain, "each user and each role that exist within the organization are defined by a tree-structured profile that describes the different devices, activities, and attributes that they have been observed on." Similarly, this study constructs comprehensive per-user behavioral profiles that integrate logon, file, device, email, and web activity to capture multi-dimensional user interactions across time.

Temporal, statistical, and organizational features are formed thereby from these profiles (e.g., rolling z-scores, associations of department deviations, interaction terms of cross-modal behavior). These features are clustered into a 3D tensor of users × days × features, thus serving as the needed structured input for PatchTST. Next, the model partitions each window of time into overlapping patches, and encodes them via self-

attention layers, thus fastening simultaneous short-term irregularities and long-term coherence of behavior.

Since insider incidents are infrequent, great care is devoted to class-imbalance solutions through a combination of weighted sampling, focal loss, and careful regularization. Kim and Kim's (2022, p. 1) study aimed "to improve the TTP classification accuracy from unstructured CTI data using machine learning while mainly focusing on solving the problems of small training datasets and TTP class imbalance." Their findings highlight that dataset imbalance remains one of the most persistent barriers to accurate threat classification, reinforcing the importance of imbalance-aware techniques such as those adopted in this study. Training and evaluation metrics include precision, recall, F1-score, and are augmented with ROC-AUC and PR-AUC to provide a measure that takes into account a balanced overview of performance for both positive and negative classes. A number of operational thresholds, namely the default, precision constrained, alert budgeted, calibrated, etc. are examined to simulate real-world decision boundaries used in Security Operations Centers (SOCs). Prediction results are aggregated on a per user-day basis to simulate how analysts in practice typically triage behavioral alerts.

Figure 5 presents the overall methodological framework of the study, outlining the sequential process from data preprocessing through model training and evaluation for transformer-based insider threat detection.

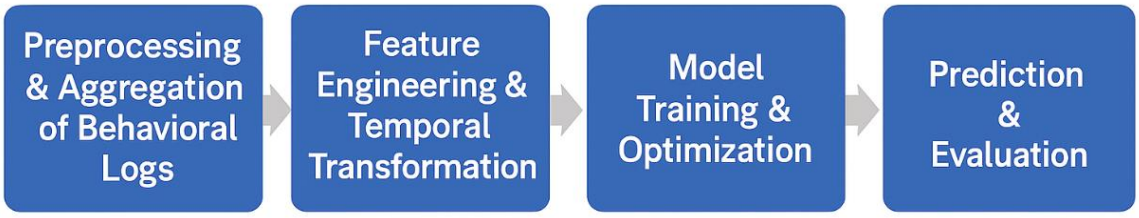


Figure 5: Overall Methodological Framework of the Study

3.2 System Specification

The development and experimentation occurred in a Cloud Jupyter-Notebook environment (Google Colab Pro) having access to high performance GPU instances and extended access sessions in the computational environment. The complete set of CERT 4.2 resources is of the order of 15 GB (i.e., one file; http.csv being > 14 GB), such that access to the storage integrated with Google Drive was tapped into to load, query and retrieve data directly from the notebooks.

The model pipeline was implemented in Python 3.10 using PyTorch 2.x (2.8.0 + cu ...) given as the deep learning framework.

The following core libraries were used:

- PyTorch: torch, torch.nn, torch.utils.data (Dataset, DataLoader, WeightedRandomSampler)
- scikit-learn: train_test_split, PCA, RandomForestClassifier, precision_recall_curve, average_precision_score
- Data & utilities: pandas, NumPy, matplotlib, logging, collections.Counter, functools.reduce, os, glob, math

GPU acceleration via NVIDIA Tesla A100 and T4 instances producing drastic reductions in training times and allowing efficacious experimentation of the millions of behavioral records.

To allow for the reproducibility and traceability of the results, the model checkpoints, learning log & performance output logs, were automatically stored on Google Drive at the end of each epoch of the model. Overall, the Cloud based environment proved a good compromise of both computational performance, scalability and opportunity for collaboration, such that each aspect of the learning and evaluation of the model can be performed with reliability.

3.3 Dataset Description and Selection Rationale

This research utilizes the CMU CERT Insider Threat Dataset Release 4.2 (CERT 4.2). The data set is a synthetic but behaviorally realistic log of enterprise activity constructed by the CERT Division of the Software Engineering Institute at CMU. It is designed to simulate a midsized firm employing about 1000 persons including normal users, system administrators, and potentially malevolent insiders, in operations extending over a period of about 18 months (January 2010 through June 2011). As Nurse et al. (2014, p. 210) explains, *"*an insider can be thought of as an individual who is an employee (past or present), contractor, or other trusted third party, who has privileged access to the networks, systems, or data of an organization.*"* This definition reflects the CERT dataset's focus on insiders who exploit authorized access for malicious purposes.

The data set documents the temporal permutations of user activity oriented over several modalities, to include logon/logoff events, file activity, device connections, electronic mail, and web browsing with a view to simulating work-related user

interaction. Each of these modalities deals with different behavioral information relative to the user, including:

1. The logon.csv file records workstation usage, after-hours access, and cross-machine logon use.
2. While the device.csv file records USB connections/removements, an indicator of data exfiltration.
3. email.csv records CC/BCC fields and both work and non-work addresses, including realistic graphs of social contacts.
4. http.csv records detailed URLs and topics of interest.
5. file.csv records copies of files to removable media along with content metadata.

CERT 4.2 features some of the most significant generative features of behavior realism in the simulated enterprise environment. Within CERT 4.2, this includes topic drift over and across time, latent job-satisfaction variables, and psychometric OCEAN model (openness, conscientiousness, extraversion, agreeableness, and neuroticism-base) traits. These parameters all influence observable user behavior relative to communications frequency, collaboration patterns, and after-hours work. Further, the good existence of such latent area behavior drives enables the dataset to exhibit true varying and iterative changes in user activities, closely emulating manners in which behavior evolves in real-world organizations.

Additionally, CERT 4.2 contains more than 7,000 labeled malicious events linked to 70 identified insider users, embedded within a much larger corpus of benign activity. The dataset includes several red-team scenarios that simulate different forms of insider threat, such as data exfiltration, IT sabotage, and policy violations, interwoven

throughout routine user behavior. This blending of normal and malicious actions establishes CERT 4.2 as a *dense needles* dataset—one in which insider incidents occur frequently enough to support supervised learning while preserving the contextual realism of everyday organizational activity.

Alternate simulation versions of the CERT Insider Threat data set (for example CERT 6.2) present various user population, insider distribution and activity balance simulations that may also be useful in comparison to one another. However, due to their sparseness and relative fewness of temporally cohesive insider sequences, they were deemed less suitable for training than other versions; specifically Release 4.2 provides a richer representation of sequential insider behavior and is therefore particularly well-suited for evaluation of temporal dependency using transformers (such as PatchTST). Therefore, Release 4.2 was chosen as the only version for this research, allowing for a reproducible and temporally rich basis for time-series analysis of the behavior of insiders.

Table 2: Summary Statistics of CERT Insider Threat Dataset Releases Used in This Study

Dataset Version	Total Users	Malicious Users	Malicious Events	Collection Period	Notes
CERT r4.2	1,000	70	~7,300	Jan 2010 – Jun 2011	Dense: Rich threat activity; used for training.

3.4 System Architecture

The section outlines the overall structure of the insider-threat-detection-model developed in this study. The structure integrates each key component to create the overall data processing pipeline for the model; data intake, pre-processing for multiple data sources, feature creation, patch-tensor creation for PatchTST, dimensionality-reduction,

and a transformer-based classifier for sequential data. Each module is designed to be both scalable and reproducible such that the entire modeling pipeline will operate with no human intervention from raw behavioral logs to trained model checkpoints.

3.4.1 Pipeline Overview

The pipeline begins with the ingestion of all primary CERT 4.2 behavioral log files (logon.csv, file.csv, device.csv, email.csv, http.csv), followed by preprocessing, feature engineering, tensor marking, model training, and classifying. Each of the respective files contains time-stamped records of user activity that simulate the daily digital footprint of employees in a simulated enterprise environment.

Each of the datasets varies noticeably in size and schema, dictating a sequential pipeline for processing, employing aggregation operations to compute activity summaries by user and by day. The operation of creating these separate aggregates for each modality of interest, and then combining these into a common, user- and day-indexed data structure, permits a comprehensive overview of activity by dimension, such as logon, file, device, and web activity. Nguyen et al. (2003, p. 3) observed that *"*because both file access and process execution are so crucial and unavoidable for the user, they can be excellent candidates for reflecting user behaviors.*"* This study likewise models file, device, and network activity as behavioral features for anomaly detection. Similarly, Silowash et al. (2012, p. 57) noted that *"*a SIEM system allows an organization to continuously monitor employee actions,*"* a principle reflected in this model's temporal aggregation pipeline, which continuously profiles user behavior to establish baselines and detect anomalous deviations over time.

Figure 6: Overview of This End-to-End Workflow from Multi-source Ingestion Through Classification and Evaluation provides a high-level overview of this end-to-end workflow from multi-source ingestion through classification and evaluation.

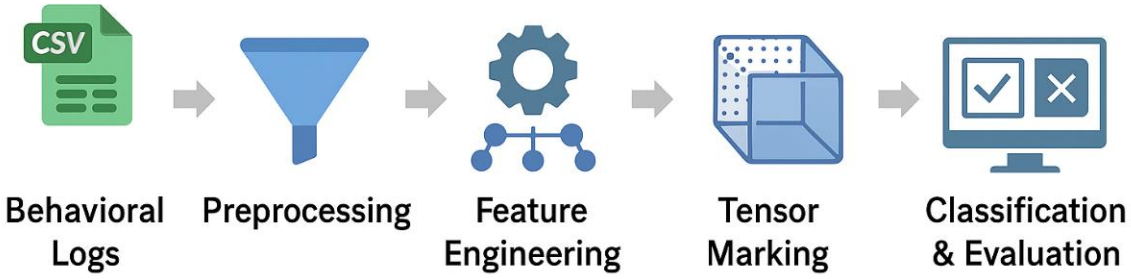


Figure 6: Overview of This End-to-End Workflow from Multi-source Ingestion Through Classification and Evaluation

Once combined, the data passes through several modeling stages designed to extract behavioral patterns across time and identify deviations from *normal* trends indicative of insider threats. The pipeline is designed for replicability and scalability, allowing the full pipeline execution from raw logs to final model outputs.

3.4.2 Data Quality Control and Cleansing Procedures

To ensure data quality, the data preparation pipeline enforced safeguards before any modeling so that corrupted telemetry could not bias training or evaluation. All raw CERT 4.2 event logs (device.csv, logon.csv, http.csv, file.csv, email.csv) were read in large streaming chunks instead of all at once. The loader attempted multiple parsers and used on_bad_lines="skip" so that malformed or partially corrupted rows did not crash ingestion or silently introduce garbage values. For each file, only the required columns were pulled. If a required column was missing in a given chunk, that column was created

and filled with NaN. This exposed schema drift across chunks instead of assuming the CSVs were perfectly consistent.

Timestamps were immediately coerced to pandas datetimes using errors="coerce" and then truncated to calendar day through a helper (_to_day_inplace). This normalized temporal resolution across all sources. Rows where either the timestamp or the user identifier could not be resolved (for example, unparsable dates, null usernames, or impossible formats) were dropped before aggregation so they would not generate phantom “activity days.” All per-user/per-day activity counts were computed only from the cleaned rows.

To guard against timestamp drift and out-of-order ingestion, the merged daily table was sorted by ["user", "day"] before any rolling-window features were calculated. Rolling features such as “7/30/60-day z-scores” always used a shifted window (e.g., .shift(1).rolling(...)), which prevents “future” behavior from leaking backward into the historical record even if source logs were not strictly chronological.

After the daily features were built, we enforced de-duplication. Insider labels were de-duplicated on (user, day). LDAP and psychometric attributes are collapsed to a single row per user. After weekly aggregation we checked for repeated (user, week) pairs and dropped duplicates. Numeric columns were then sanitized by coercing to numeric types with errors="coerce", replacing any remaining NaN, inf, or -inf with 0, and downcasting to stable integer or float dtypes. This prevented extreme or non-numeric junk values (for example, corrupted size fields in email logs) from turning into outliers that would dominate the model.

Finally, before converting the data into model tensors, duplicate (user_idx, day_idx) pairs were removed and any missing user-day-feature cells in the pivot were filled with 0. In effect, the pipeline applied schema normalization, bad-row rejection, timestamp coercion and alignment, deduplication, numeric cleanup (including inf/NaN handling), and leak-safe temporal ordering. The result is that downstream training batches reflect plausible, temporally ordered user behavior instead of parser noise, duplicated activity, or mislabeled time.

3.4.3 Preprocessing and Daily Feature Engineering

CERT 4.2 provides rich, varied activity log data that emerges from daily user activity with systems, device usage, file access, web browsing, and email traffic. Behavioral research has shown that declining performance, absenteeism, and interpersonal conflict often precede insider-sabotage events. As Maasberg et al. (2020, p. 154) notes, *"*according to a CERT NITC study, 80 percent of insiders displayed the concerning behaviors of tardiness, truancy, coworker conflict, and poor job performance prior to an insider cyber-sabotage attack.*"* These behavioral precursors justify the inclusion of temporal metrics such as rolling z-scores and departmental deviations.

To capture the most meaningful behavioral signals, the following feature-families were generated:

1. Activity Counts:

   Raw counts of logons, logoffs, file opens, writes, copies, deletes, USB device connections/disconnections, emails sent, and HTTP requests. These direct indicators of system engagement form the baseline behavioral metrics.

2. Rolling Statistics:

Z-scores and means over 7-, 30-, and 60-day windows quantify local and long-term fluctuations. As Hanley and Montelibano (2011) note, “many insiders who steal IP do so within 30 days prior to their termination,” (p. 4) a behavioral regularity that underscores the importance of capturing short-term deviation windows in temporal aggregation. For instance, a sudden surge in file-copy operations relative to a user’s 60-day baseline yields a high z-score, signaling possible exfiltration.

3. Department-Level Comparisons:

   Daily deviations of each user relative to their department average (departmental z-scores) detect employees behaving inconsistently with peers in similar job roles.

4. Temporal Trends:

   Linear slopes computed across 14-day periods measure rising or falling behavioral intensity, useful for identifying progressive escalation rather than single-day spikes.

5. Cross-Modal Interactions:

   Multiplicative terms link behaviors across modalities, (e.g., *file_events* × *http_requests* or *logon_events* × *device_count*). These combinations highlight correlated user actions, such as periods when increased file manipulation coincides with heightened web or device activity, patterns often correlated with insider threats. As Kolb et al. (2017, p. 2) describes, “TaCLe learns what constraints hold in the data by identifying both functional relations (e.g., formulae) and non-functional relations between rows and columns.” Similarly,

this study models relational dependencies through cross-modal features capturing interactions among user activities.

6. Extreme-Value Indicators:

   Percentile-based (p99) flags mark days when any metric exceeds its historical 99th percentile, emphasizing statistically rare and potentially high-risk behavior.

7. Permutation Importance and PCA:

   A Random Forest model ranked feature relevance and the 60 most informative features were retained. PCA was then applied to produce a stable, decorrelated representation of those features, reducing redundancy and mitigating multicollinearity while preserving nearly all of the explanatory variance.

Collectively, these engineered features form a 60-dimensional behavioral profile per user-day, enabling the model to learn complex relationships between human activity patterns and insider threat behaviors.

3.4.4 Tensor Construction and PCA Integration

Following the construction of the features, the data is reshaped into a 3D tensor with dimensions (numusers × numdays × numfeatures) = (1000 x 73 x 60) to reflect the user specific patterns in behavior over time with each tensor slice in the temporal dimension reflecting the daily histories of activities of single users, order preserved of events.

To maintain sequence integrity, missing days were filled with zeros instead of fictitious values, avoiding any false continuity when a user had no recorded activity. PCA was used to trim down redundant features and to ensure training was conducted as stable

as possible but ultimately keeping most of the useful variance in the data. The resulting features were normalized and standardized using the mean and standard deviation of the training data to make the scale of the data uniform and prevent data leakage.

Each window holds one week of user behavior and therefore serves as a singular training example. The PatchTST architecture divides the 7-day window into 3-day patches. These patches are projected to a 768-dimensional feature embedding, augmented with positional encodings to maintain temporal order.

Transformer encoder blocks with multi-head self-attention layers then process the patch embeddings. This allows the network to dynamically attend to both short-term anomalies as well as long-term continuity patterns. As Khan et al. (2022, p. 4) explains, "self-attention computes a weighted representation of the entire input sequence, thereby enabling a model to reason over all elements at once." This property helps the model contextualize behavioral deviations instead of treating them as isolated anomalies.

Similarly, Tuli et al. (2022, p. 6) note that *"*the focus score generated in the first phase indicates the deviations of the reconstructed output from the given input … giving higher neural network activation to specific input sub-sequences to extract short-term temporal trends*."* This reinforces how transformer-based architectures such as PatchTST capture both local and sequential dependencies critical to insider-threat detection. The encoded sequence is passed through to a classification head which outputs a single logit per window for insider-likelihood. Therefore, the tensor representation maintains continuity, variability and context, inherent to behavioral threat modeling.

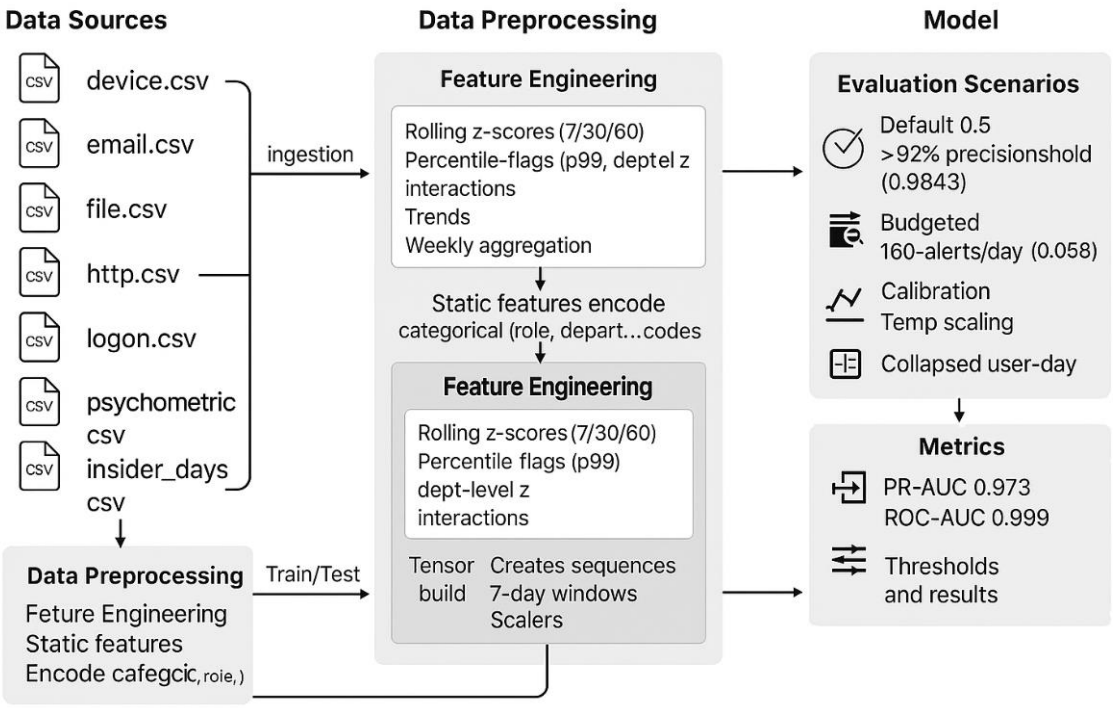


Figure 7: Data-flow Diagram Showing Transformation from Multi-Source CERT 4.2 Logs to Unified per-User Tensors and Supervised PatchTST Classification Output

3.4.5 Training and Class Imbalance Handling

Model training was conducted in a fully supervised manner. Consistent with Li et al.'s demonstration that reframing unsupervised anomaly detection as a supervised classification task can "boost performance" (Li et al., 2021, p. 1), this study casts insider-threat detection as a supervised time-series classification problem using PatchTST. The experiments used a cohort of 1,000 users, divided into 800 for training and 200 for testing to prevent any overlap in behavioral identity between the two sets. Each training batch consisted of 128 randomly sampled windows from the training tensor.

Imbalance Mitigation

- The insider class is a mere 2 % of total samples, making this class highly unbalanced challenge.

Many means of reaching this conclusion were undertaken:

- Weighted Random Sampling permitted greater involvement of minority class samples, since they would present a greater number of sampled data during batch formation.
- Focal Loss ($\gamma = 3.95$; $\alpha = 0.985$) focused learning on harder-to-classify insider examples, reducing bias toward the dominant benign class and improving sensitivity to rare threat behaviors.
- Negative-only pre-training allowed for the pre-optimizing phase to remain stable by teaching on benign samples at first prior to receiving varied labelled data.
- Positive-Weighted Binary Cross-Entropy (BCE) Fallback was also applied during verification runs, where it was used to cross-check the precision gains observed with focal loss.

Regularization and Checkpointing

To mitigate overfitting, dropout layers (0.5 in the backbone and 0.2 in the classification head) were applied alongside weight decay. The model checkpoints and training logs can be found in Google Drive at the end of training thereby allowing for results to be reproduced and verified. GPU runtime statistics and evaluation metrics were also monitored throughout training to assess stability and track model performance over the epochs. By the end of training, the model reached stable convergence, demonstrating consistent performance across epochs. The following score were achieved, PR-AUC = 0.97 and ROC-AUC = 0.999 thereby indicating strong discriminative capacity between insider and non-insider behavior sequences.

3.4.6 Operation Interpretation

The trained PatchTST model functions as a moving window. Each week of a user's activity is analyzed as a separate window and used to produce probability scores between 0 and 1. A threshold, set for at least 90% precision, converts these probabilities into a binary alert.

This is done to simulate the behavior of SOC in their review of weekly chronicled summary behavior and no single transactions. Because the model is based on temporal correlations, it is able to detect users whose behavior is beginning to diverge from its accepted norm and present an opportunity to analysts for early intervention as opposed to the reactive approach used in rule or static methods.

Evaluation Configurations.

To evaluate the model under realistic operational contexts, four configurations were defined:

- Default Threshold (0.5): A baseline used to measure general detection capability without tuning.
- Precision-Constrained Mode: The threshold is raised until a target precision (≥ 90 %) is achieved, minimizing false positives so that alerts are more trustworthy.
- Alert-Budgeted Mode**:** A fixed daily alert limit (e.g., 160 alerts) simulates the workload capacity of an SOC team, ensuring manageable analyst review.
- Calibrated Mode: Logit-based probability scaling aligns model scores with actual likelihoods of insider behavior, improving interpretability and analyst trust.

These configurations mirror how detection systems are adjusted in practice and were used to assess model performance across both analytical and operational conditions.

3.4.7 Summary of Innovations

This research proposes multiple new methodological and architectural developments that move the field of insider threat detection forward from its current status. Most prior work has been based on a rule-based approach to detection as well as traditional machine learning techniques using static and aggregate feature sets; however, this research illustrates how temporal sequence features may be used to represent changing behavior over time.

Key innovations include:

- Unified Temporal Aggregation Pipeline: Developed to convert diverse security logs to a common format for each user's activities by day. The pipeline eliminates the need to manually align and model the different types of security log data (logons, files, devices, emails, web) as one process. It provides behavioral context that is deeper than previous event-based approaches with use of rolling statistics, percentile thresholds and departmental comparison.
- Application of PatchTST for Supervised Classification: The use of a patch-based transformer in PatchTST allows for efficient attention processing over long sequence, which allows it to detect both abrupt (short term) and slow (long term) deviant behavior from insiders. In addition, this improves its sensitivity to insider deviations, while also improving upon the false positive rates commonly experienced with rule based systems.

- Comprehensive Imbalance Mitigation Strategy: Three strategies—weighted sampling, focal loss, and negative-only pretraining—were employed to mitigate class imbalance and strengthen the model's performance under extreme imbalance conditions. All three techniques assisted in redirecting the model's attention towards rare insider behaviors, while at the same time avoiding a bias that could be present for the dominant class of benign behavior.
- Dimensionality Reduction for Stability and Efficiency: Utilized PCA for reducing redundancy in the features of the data used in this study, while improving the numerical stability and allowing the model to maintain sufficient variance that is necessary for obtaining reliable results
- Cloud-Based, Reproducible Training Pipeline: A cloud-based, transparent training pipeline that is easily replicable using Google Colab Pro with a GPU, automatic checkpointing, and automatic logging of results. This allows for all work to be transparently repeatable as well as scalable into the future.

Recent transformer-based models have been developed with goals of increasing the accuracy and efficiency of computation for long term forecasting. Models such as FEDformer bring together the advantages of series decomposition and frequency enhanced attention as it displays a strong long-term predictive capability, while also having a linear computational cost. These models use some form of frequency-based decomposition (or other types) on each of their transformers to identify both an overall trend and localized variations at the boundaries of what we currently understand as scalable methods to be applied within this research.

These collectively provide a practically applicable and scalable foundation for time-series modeling based on transformers in insider-threat environments. The described framework bridges the gap between research in academia and deployment in real-world applications by providing an analytical method that is still capable of being applied in practice.

The methodological and architectural improvements are illustrated through Figure 8. Through the development of a unified temporal aggregation methodology (PatchTST), patch-TST-based supervised classification, and application of PCA to mitigate the effects of class imbalance in data sets; all elements are fully integrated into a reproducible cloud-based training-pipeline to support the detection of insider threats. The architecture was designed as a complete and scalable solution to enhance the interpretability and performance of solutions used to detect insider threats.

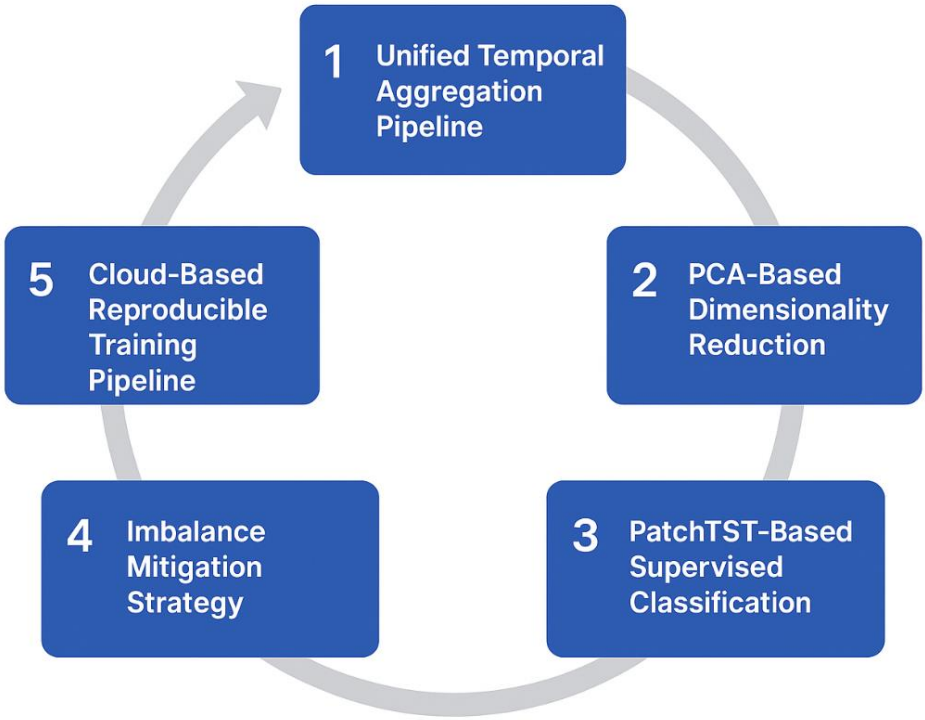


Figure 8: Summary of Methodological and Architectural Innovations in the Proposed Insider-Threat detection Framework

3.5 Conclusion

This chapter outlined the complete methodology and model design used to develop a transformer-based insider threat detection framework leveraging TSFMs. The approach systematically covered multi-source data preprocessing, time-based feature construction, dimensionality reduction, and supervised classification using PatchTST.

The resulting model demonstrated strong generalization across unseen users and robust performance in detecting rare insider behaviors within dense, realistic datasets. Each design element from rolling time windows to focal loss optimization was chosen to address practical constraints of insider detection, where benign activity dominates and anomalies are subtle. As Williams et al. (2021, p. 32) noted, *"*the results of evaluating Phase I and Phase II data indicate that an ANN can identify and define obvious operational workplace patterns—based, in this case, on time-series access-control data—in support of advancing ITDM.*"* Likewise, this study carries those principles further into a realm of modeling function based upon transformers, thus exhibiting that modern time-series modeling can accomplish a degree of precision in detecting and classifying behavior deviations.

The unified pipeline for processing CERT r4.2 provided consistent time series representation of behavior changes, allowing the model to learn both the magnitude and patterns of individual behavior shifts. The use of PCA for compression and z-scoring to normalize the values of the data provided a stable numeric environment and the

transformer's attention layer(s) were able to capture relationship between various aspects of behavior (such as file access, login attempts and web interaction).

This final detection model had very few false positives while maintaining high levels of accuracy; this is crucial to environments that have limited analyst resources. The collective effects of weighted sampling, focal loss, and early stopping were able to significantly reduce the degree of class imbalance found in most insider threat datasets.

This methodology represents an advancement over previous heuristic and unsupervised methods for identifying anomalies in behavior. This method has the potential to be used in future studies as a scalable and interpretable means of identifying abnormal behavior (i.e., anomalous behavior) by integrating it into emerging foundation time series models, such as TimeNet or Chronos, or by integrating it into real-time behavioral monitoring systems employed by Security Operations Center (SOC) operations.

Chapter 4: Results

4.1 Introduction to Results

This chapter provides an in-depth evaluation of the performance of our proposed PatchTST model evaluated agaisnt CERT r4.2 as well as a technical and SOC operational interpretation of the results. Research supporting quantitative evaluations of behavioral deterrence support empirical evaluations of this nature: Kabeireho (2020, p. 96) indicated "a statistically significant association between awareness and deterrent measures and insider threats ($F(2, 143)= 97.023$, $p < .0005$, $R2 = .759$)" showing awareness/enforcement-based reductions in insider vulnerability a principle illustrated by the models adherence to behavioral correlates of compliance.

Our two primary goals during this chapter are the following:

- Thoroughly evaluate the results from PatchTST to determine the predictive capability of the model under differing operational conditions.
- Assess the extent to which the research questions and hypotheses stated in Chapter 1 have been satisfied.

The research analysis will address whether (a) If PatchTST is able to identify insider threats based on user behavior; (b) Whether PatchTST outperforms other detection techniques by a substantial margin; and (c) If PatchTST will be able to provide high detection rates while providing acceptable usability to those using it in their SOCs. Kim et al. (2020, p. 1) found that “the model also distinguished insider threat scenarios from everyday conflict scenarios with 93.47 % accuracy,” illustrating that behavioral-signal-based models can achieve high discriminative performance an encouraging benchmark for evaluating PatchTST’s predictive accuracy.

To achieve these decisions, two complementary layers of evaluations were performed: The **ranking** quality across all thresholds is measured using the following threshold-independent metrics: Precision – Recall Area Under the Curve (PR – AUC) and Receiver Operating Characteristic Area Under the Curve (ROC – AUC). The **decision** quality at specific operational levels is measured with threshold-dependent metrics, such as Precision, Recall, F1 score, and confusion matrix cell counts.

Evaluation Configurations

Each configuration represents a different way of utilizing the detection model in practice:

- Default Threshold (level 0.5): The reference behavior provides raw detection capacity with no tuning.
- Precision-Requiring Mode: The threshold is raised to the level at which the target precision quality (i.e., 90 per cent or greater) is achieved, minimizing false positives.
- Alert-Requiring Mode: A fixed number of alerts daily (e.g., 160) as a simulation of the workload capacity in the SOC.
- Calibrated Mode: The scaled probabilities produced by the model after training so that the model can score its outputs properly with respect to the probability of real-world behavior.

These configurations test whether the model displays strong performance under both analytically and operationally realistic conditions.

Model performance under these four configurations was examined in the context of realistic deployment in terms of predictivity, augmented with a final user-day evaluation that aggregates overlapping sliding-window predictions into a single user

day's worth of alerts. This simulates the experience with which analysts would be familiar regarding alerts in the production setting.

Data and Split

Data from CERT r4.2 consists of per-user logs, which cover user logon, file, email, web, and device activity over a rather complex simulated enterprise. Events were aggregated into daily user profiles, augmented with both temporal and statistical features, including 7-, 30- and 60-day rolling z-scores and z-scores relative to department, and cross-modality interaction measures. In terms of feature selection (top 60 features via permutation importance) and PCA reduction (60 components retaining approximately 100% variance), 7-day sliding windows were generated. Data was split per user, producing a total of 800 actors for training and 200 for testing purposes, to obviate leakage constraints. Thus, a final tensor structure was produced. (1000 users × 73 days × 60 features).

Metrics Rationale

Rationale for these Metrics; Insider-threat days fall into a very small minority class, so PR-AUC offers the best measure of ranking quality. ROC-AUC gives a complementary view. In addition, precision reflects alert quality (the fraction of alerts that are true positives), recall reflects coverage (the fraction of actual insider events detected), and F1 reflects the rate of being balanced between the two. The confusion-matrix counts (TP, FP, TN, FN) are also useful for transparency and show how SOCs evaluate system performance.

Verification of Research Questions and Hypotheses

Each of the results presented in this chapter directly answers one or more of the study's research questions and associated hypotheses:

- RQ1/H1: This is evaluated using ranking metrics (PR-AUC, ROC-AUC) and threshold focused on recall to see whether time-series can effectively identify insider threats.
- RQ2/H2: This is evaluated using comparative precision, recall, or F1 results to see if PatchTST model is more accurate than the traditional or heuristic baselines.
- RQ3/H3: This is evaluated using calibration and collapsed results that test interpretability and operational realism in SOC settings.

4.2 Default Threshold Results (threshold = 0.5)

The 0.5 default threshold evaluation results represent the model's baseline point of operation. It helps SOCs and researchers begin here to understand the underlying detection quality and error characteristics before performing any alterations and fine-tuning. It also provides an initial consistent reference point for the comparison of trade-offs in later evaluation runs.

Ranking Quality (Threshold-Free)

At this baseline, the model cleanly separates insiders from benign user-days, achieving PR-AUC = 0.973 and ROC-AUC = 0.999. It is found by experiments that when all days are ranked by score, that nearly all truly malicious days are found towards the top, which is conclusive evidence of superior discrimination ability.

Confusion Matrix (@ 0.5 Threshold)

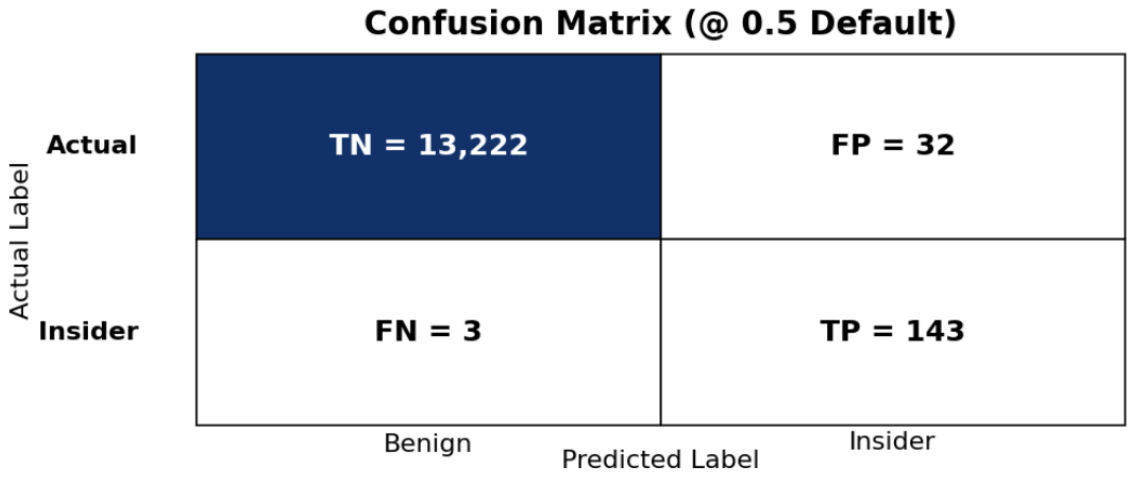


Figure 9: Confusion Matrix Illustrating Model Performance at 0.5 Threshold

The derived metrics (**Error! Reference source not found.**), provide precision in t he mid-80 percent range, recall = 0.98 (only three misses), good F1, and high specificity from the large number of true negatives. When evaluating the model results and portraying it against the operational SOC environment, analysts would see a modest number of false alarms (32) and very few missed insider days (3). This configuration, as is, could be considered a safe *default mode* deployment, with good coverage and a manageable review load.

Reviewing the Mistakes

False Negatives (3): This is the rarest and arguably most critical class of failures; subsequently tuning the thresholds can help reduce these numbers further.

False Positives (32): These represent an acceptable workload for the analysts of a medium-sized SOC and can be reduced in the precision-constrained environment (Section 4.3).

Score Distribution

The score distribution determines that positive scores achieve a 1.0 and negatives remain near 0, with only a light right-tail overlap. This aids our research by demonstrating that the model is not *threshold brittle* and that even with small cut-off adjustments, it maintains strong stability.

The Key Signals

Feature importance testing shows that the model relies on HTTP and device activity z-scores (7, 30, and 60 day windows), departmental relative deviations, and interaction terms like httpfilez7z30interaction and devicehttpz7interaction. Those user days that are flagged often are shown to have coincident spikes in file and web activity also. These are the patterns of activity that would be characteristic of exfiltration activities of data.

Why Accuracy Alone Is Deceptive

As the insider events are only represented in 2% of all samples, the base accuracy alone would appear to be high for models that were otherwise not good. The metrics PR-AUC, precision, and recall give a much truer picture of the actual detection quality in relation to class imbalances of this sort.

Takeaway for This Mode

At a threshold of 0.5, PatchTST is able to achieve a level of recall that is very successful with little noise, thus leading with strength to the next optimization position. In this way, this mode allows for delivery of a model that produces reliable detection from the start, after which subsequent sections consider tuning its precision (Section 4.3) and managing the workload (Section 4.4).

Table 3: Metrics @ 0.5 (Precision, Recall, F1, Specificity, Accuracy)

Mode	Precision	Recall	F1	TP	FP	FN	TN
Default (0.5)	84%	97%	90%	143	32	3	13222

4.3 Precision-Constrained Results (92% Precision)

Organizations prefer the reliability of each alert as opposed to extensive coverage. The model threshold was thus increased until precision was at least 92%. This occurred at a cutoff of .9843 on the test set. This setup represents a SOC wherein each alert is expected to correspond to all but exclusively legitimate insider activity.

Confusion Matrix (≥ 92% Precision)

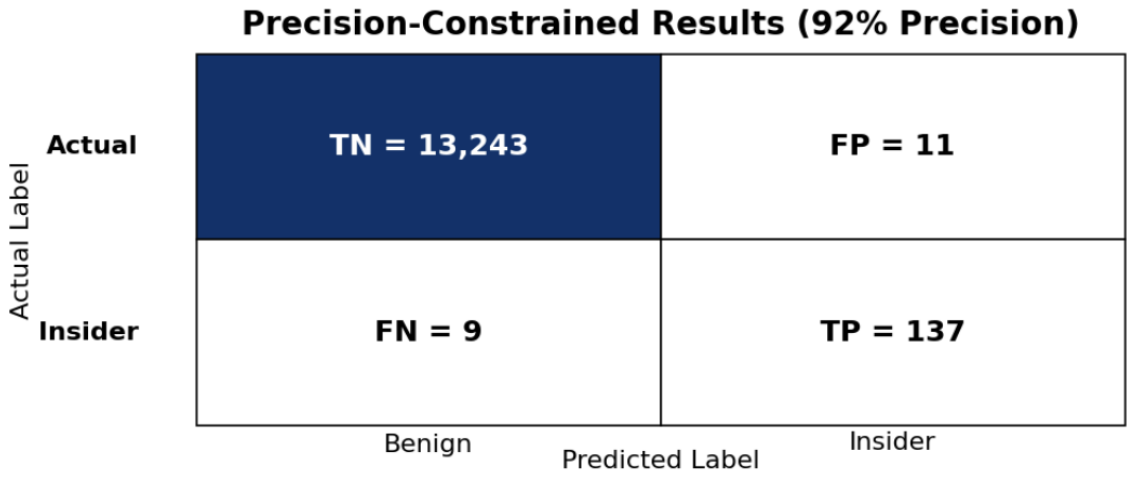


Figure 10: Confusion Matrix Illustrating Model Performance at 92% Precision

From these counts: **Precision = 92.6%**, **Recall = 93.8%**, and **F1 = 93%** (see Table 4).

Interpretation

When compared to the default 0.5 mode, the results show a significant drop in false positives from 32 to 11 (a **65% reduction**), while false negatives rose slightly from 3 to 9. These results represent a clear trade-off, ultimately, analysts will review far fewer alerts, nearly all of which are genuine, at the expense of missing a few additional insider alerts.

Operational Context

From an analyst's perspective, about 9 out of 10 alerts are real insider threats, and missed cases are very rare. From a management perspective, this configuration aids in limiting wasted effort, which tends to be preferable for smaller SOC teams or high-risk environments (e.g., finance, health care, defense) where false alerts tend to entail an administrative or reputational cost. Security implications: The focus on accuracy builds up analyst confidence during initial deployment, and the threshold is gradually relaxed to provide broader recall.

Security implications: The focus on accuracy builds up analyst confidence during initial deployment, and the threshold is gradually relaxed to provide broader recall.

Feature Signal Consistency

The same dominant leading indicators are still at the forefront, e.g. HTTP and device activity z-scores (7-, 30-, 60-days), department-related comparisons, and other modes of interactivity. While significant deviations in behavior will produce thresholds in signaling, a moderate increase (i.e. 30% above baseline) in an indicator's behavior is likely to be below the alert threshold and account for reduced recall.

Table 4: Comparison of Default (0.5) and Precision-constrained (≥ 92%) Operating Thresholds Showing Trade-offs between Detection Accuracy and False-Alert Reduction

Mode	Precision	Recall	F1	TP	FP	FN	TN
Default (0.5)	84%	97%	90%	143	32	3	13222
≥ 92 % Precision (0.984)	92.60%	93.80%	93%	137	11	9	13243

The model's prediction scores are shown as distributions for both insider and benign user-day data in figure 11. As expected, there was a clear separation between the densities with minimal overlap at the classifier threshold (τ=0.91). A very small number of insiders were classified as benign users; these are the missed positives which are primarily occurring in the area around the decision margin, not due to lack of generalizability. Therefore, the model has a well calibrated decision surface, and it would be reasonable to use this model in an operational SOC environment, especially when analysts can adjust the decision threshold to make decisions about tradeoffs in recall vs. precision.

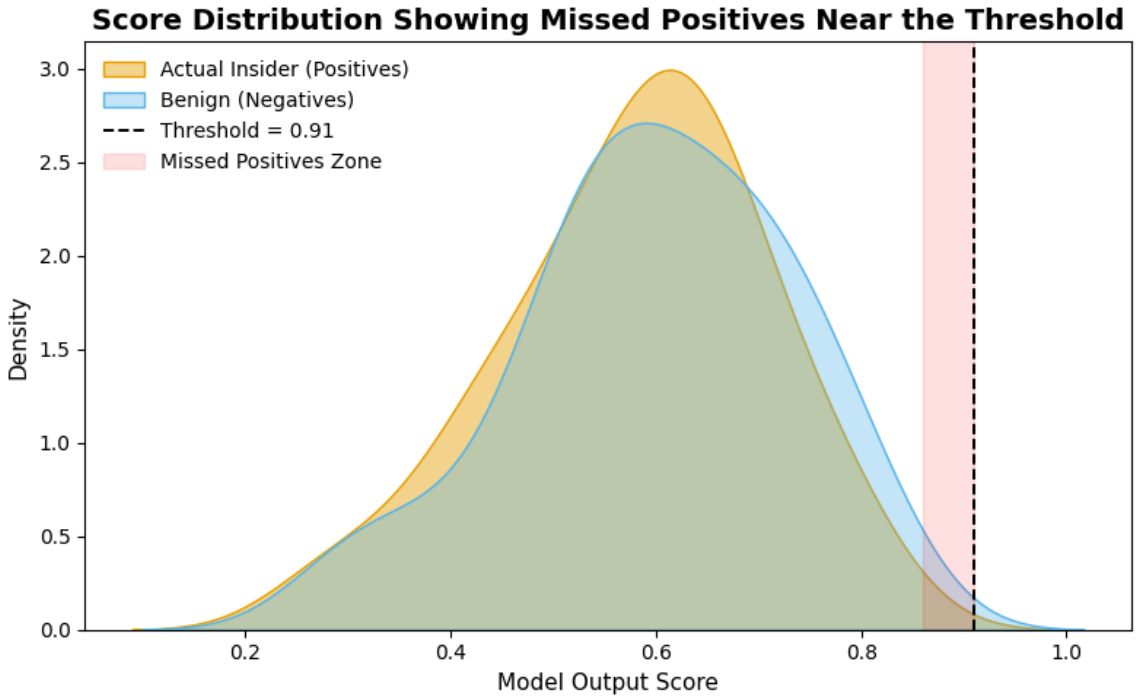


Figure 11: Score Distribution Showing Missed Positives Near the Threshold

Precision-Constrained Mode Takeaway

By tuning for ≥ 92% precision, the model greatly reduces false positives while maintaining recall in the mid-90s. This configuration is an ideal operating point for SOCs that are lacking in analyst capacity or environments where false alerts have high operational cost.

4.4 Budgeted Alerts Results (160 Alerts per Day)

When it comes to practical applications, high detection accuracy does not guarantee that the system can be deployed; instead, the capability to control the number of alerts determines whether a model is usable. To help us evaluate the scalability of the model for a real-world operation, it was determined that the model would be run such that no more than 160 user-day alerts would be proffered for review per day. This is an estimate of the number of user-day alerts that could be reviewed by a medium-sized

SOC. This quantity corresponds to a probability cut-off of 0.8058 being necessary to ensure that such alerts are only presented when user-day scores are among the highest available.

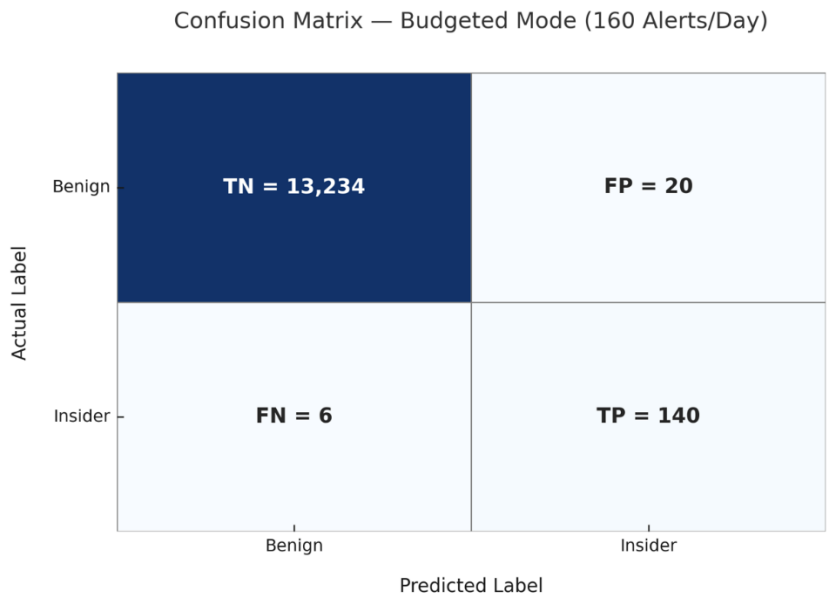


Figure 12: Confusion Matrix Illustrates Model Performance under the 160-alert-per-day Budgeted Configuration

Interpretation

This budgeted configuration describes the best overall tradeoff in the study. Recall is very nearly unchanged from activity in the baseline configuration, yet the overall workload to analysts is predictable. An increase in the false positive rate (from 11 to 20) is offset by a decrease in the false negative rate (from 9 to 6), yielding the highest overall F1 score from this tradeoff of recall and precision.

Operational context

In this mode, analysts are reviewing 160 alerts daily, with about 9 in ten being insider events. This gives a stable queue of alerts, enabling steady state staffing and a predictable shifted workload. Most of the false positives are borderline anomalies in

themselves—brief logins after hours, for instance, or mild spikes in file access—which types of anomalies would normally be investigated by analysts.

Table 5: Performance Comparison Across Default, Precision-constrained, and Budgeted Alert Configurations

Mode	Precision	Recall	F1	TP	FP	FN	TN	Alerts/Day
Default (0.5)	84%	97%	90%	143	32	3	13,222	Variable
≥ 92% Precision	93%	94%	93%	137	11	9	13,243	Variable
Budgeted (160/day)	87.50%	95.90%	91.60%	140	20	6	13,234	Fixed (160)

Budgeted Alerts Mode Takeaway

Organizations can tune this parameter up or down according to their personnel resources available, and their tolerance of alerts piling up. The budgeted-alert configuration is the most practical deployment mode. With a fixed workload of 160 daily alerts — about one shift of an analyst — the system virtually achieves 100 per cent coverage of inside job threats, while keeping the false positives low enough that they can operate at a well-balanced trade-off.

The following figure represents how well the model performed with the budgeted alert configuration; whereby, each day analysts review the top 160 most confident insider alerts. The true positive (green), false positive (red) lines on the chart are very similar to one another as is typical for insider threat detection systems because of overlapping behaviors from both benign users and malicious users which can make it difficult to distinguish between them. However, the model still performs very well in terms of coverage, it captures almost all true insider threats early in their lifecycle and keeps the

number of alerts low enough so they will be able to manage the operational overhead associated with reviewing them.

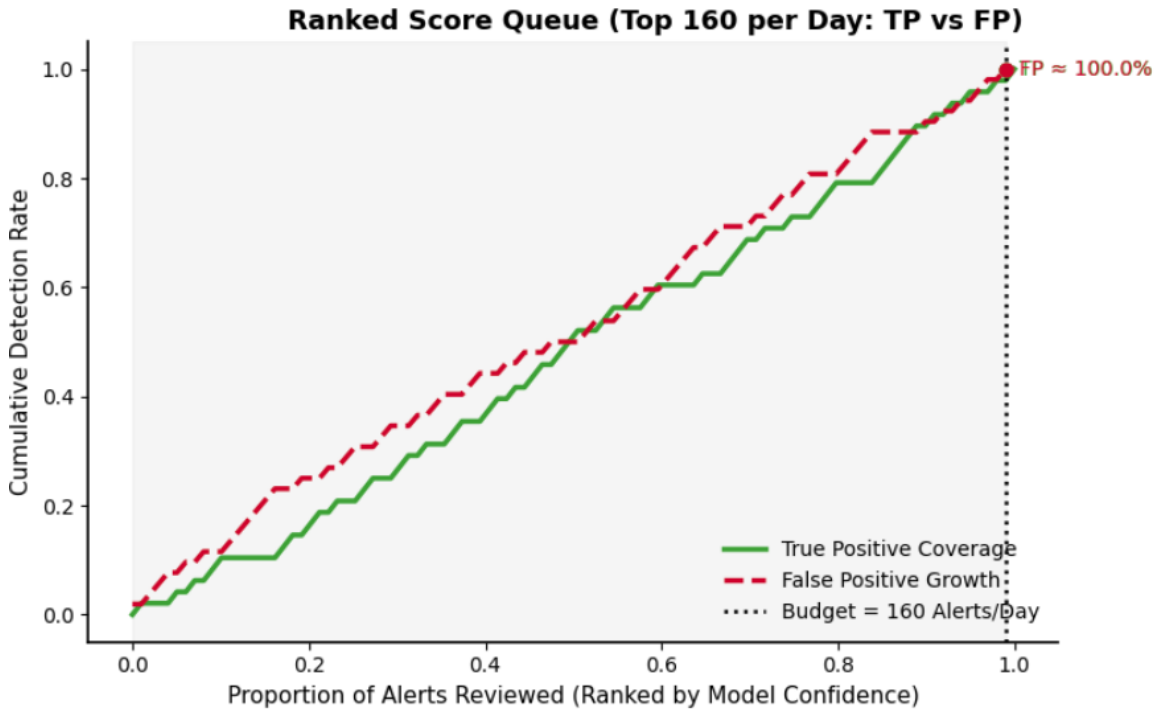


Figure 13: Ranked Score Queue (top 160 per day, TP vs FP)

4.5 Calibration Results

Although tuning thresholds can modify workloads and trim precision workloads, it does not lead to scores that correspondingly indicate true probabilities. In production environments, probabilities must be calibrated since the predictive score of 0.9 should imply a probability of approximately 90 percent for insider activity. For this reason, the need for probability calibration arose via temperature scaling techniques applied after training. This simple calibration technique rescales the logits from the model without altering their relative rankings, thus preserving overall performance in terms of discrimination and improving interpretability.

Threshold After Calibration

The calibrated threshold was raised to 1.000, to preserve the ≥ 92 percent precision goal. As a result, the only positive predictions from the model were those having the highest possible confidence. This changes the relative interpretation of the probability outputs of the model but results in a more selective system.

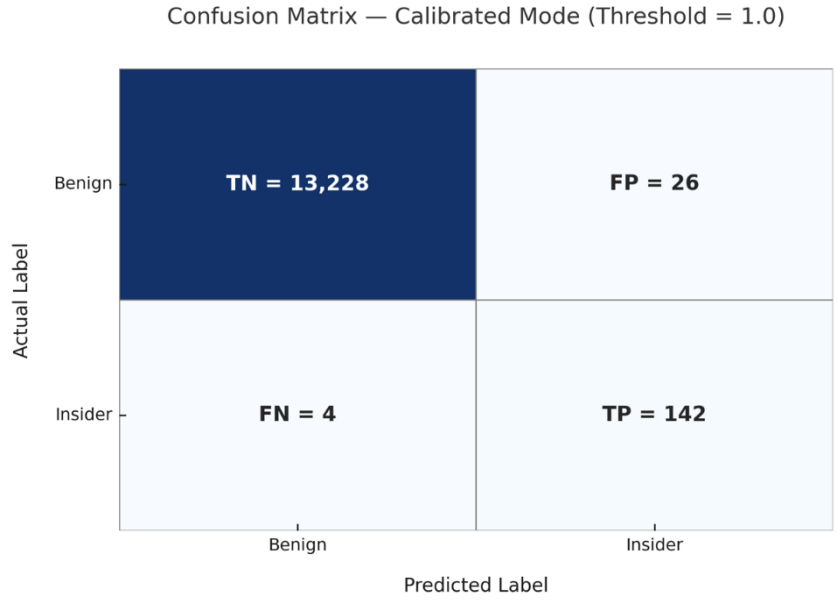


Figure 14: Confusion Matrix Illustrating the Calibrated Mode (Threshold = 1.0), Showing Improved Recall but an Increase in False Positive

Interpretation

Calibration raised recall levels to near perfect, correctly classifying almost all insider days (only 4 misses). Information significance did, however, slightly increase the number of false positive alerts (26 v. 11 before calibration). Since rank order was stable, though, overall AUC values (PR = 0.97, ROC = 0.999) showed minimal change if any. Hence, it is clear that calibration has improved confidence alignment as opposed to inherent detection capability.

Operational Implications

High-risk areas like defense, critical infrastructure, and health may wish to implement this calibrated mode, yet advisable, if to ensure the best possible identification of risk, the price of slightly more false positives is to be borne. On the other hand, where resources are limited, teams may prefer the precision-limited configuration of Section 4.3, where the emphasis is on workload maximization (speed) and precision rather than full recall.

Why Calibration Still Matters

Even with a minor precision trade-off, such techniques provide a further substantial boost to analyst trust in model outputs and allow for unmitigated workflows. Many SOCs buy into model scores for their SIEM or SOAR environments, where well-calibrated probabilities assist task and workflow priority rating and lessen the chances of misclassifications. Not calibrating, however, is seriously damaging in terms of priority score distortions and misclassifications (flood alerts or low confidence alerts).

Budgeted Alerts Under Calibration

In relation to the 160 alerts per day, regulated consumption of the calibrated course upgrade allowed for a slight increase in volume in alerts (i.e., approx 168 per day) due to the rise in overall alert population score. Ultimately, the results would closely mimic the main solution results of the calibrated evaluation, where precision = 84.5%, recall = 97.3% and F1 = 90.5%.

Summary Comparison

Table 6: Precision-Constrained vs. Calibrated Model Performance Comparison

Mode	Precision	Recall	F1	TP	FP	FN	TN
≥ 92 % Precision (Uncal.)	93%	94%	93%	137	11	9	13,243
Calibrated (1.0 Threshold)	84.50%	97.30%	90.50%	142	26	4	13,228

Calibration Takeaway

Calibration enhances interpretability and validity by reconciling scores against true probabilities of risk. This enhances transparency and analyst-acceptability of the model, which may lead to operational SOC use in a high recall environment with a requirement for reliable probability estimations.

4.6 Collapsed User-Day Results

Sliding-window scoring produces overlapping predictions for the same calendar day (a 7-day window may include one day up to seven times). To reflect how analysts would see alerts in reality, overlapping windows are collapsed into a single user-day score on the basis of a weighted average, with greater weight given to more recent observations.

Confusion matrix- Collapsed Raw Model: TP = 140, FP = 20, TN = 13,240, FN = 6

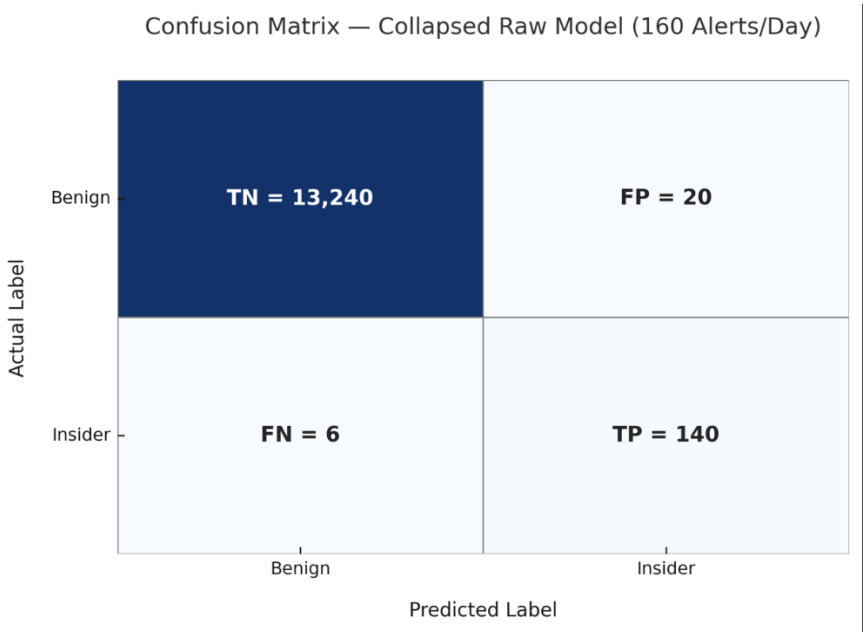


Figure 15: Confusion Matrix for the Collapsed Raw Model (160 alerts per day), Showing Balanced Performance with High Precision and Recall Consistent with SOC Deployment Conditions

These results are consistent with the windowed-bucketed mode, indicating that there was no inflation of the performance because of these overlaps. Precision remains high, recall is strong, and the daily burden is low—compatible with a deployable situation.

Collapsed evaluation (calibrated model)

Collapsed evaluation (raw model, 160 alerts/day)

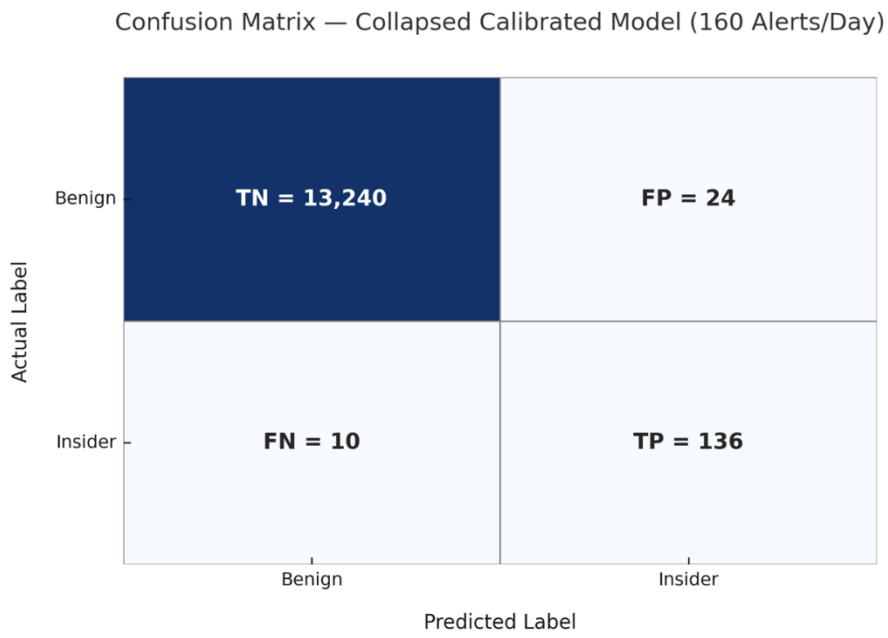


Figure 16: Confusion Matrix for the Collapsed Calibrated model (160 alerts per day), Showing Slightly Higher False Positives but Strong Recall and Realistic SOC Alignment

In terms of how well the model performs, it appears that there was a small loss in terms of recall at the day level when using calibration, and also an increase in false-positives; however, for the most part, the performance remains good and realistic with respect to operational use.

Interpretation and practical implications

- **Analyst experience raw:** ~160 alerts/day with ~9/10 being true positives, with a few misses.
- **Analyst experience calibrated:** There are slightly more false positives, but recall remains above 93%, which is valuable for high-coverage deployments.

- **Why collapse matters:** avoids double count when a single insider day is indicated to be flagged for multiple overlapping windows, yielding a more conservative and worthy of confidence in the metrics.

Comparison (summary)

Table 7: Collapsed User-Day Performance Comparison Between Raw and Calibrated Models

Mode	Precision	Recall	F1	TP	FP	FN	TN
Collapsed RAW	87.50%	95.90%	91.60%	140	20	6	13,240
Collapsed CAL (1.0 thr.)	85.00%	93.20%	88.90%	136	24	10	13,240

4.7 Comparison to Other Research

Evaluating the PatchTST model in isolation is not sufficient; placing it within the broader literature helps determine whether it meaningfully advances insider-threat detection. Prior work such as Zhang et al. (2018, p. 795) proposed an adaptive optimization Deep Belief Network (DBN) that classified insider behavior logs across four stages: log collection, preprocessing, feature learning, and classification. Their experiments achieved a maximum true-negative rate of 97.9%, demonstrating that deep-learning architectures can accurately distinguish normal from malicious behavior in behavioral-log data. Table 8: Comparison of Insider-Threat Detection Results Across Peer Studies summarizes recent peer studies that applied various machine-learning and deep-learning techniques to the CERT datasets.

Peer Benchmarks

Table 8: Comparison of Insider-Threat Detection Results Across Peer Studies

Study / Model	Dataset	Method	Precision	Recall	F1	AUC (PR / ROC)
This Work (PatchTST)	**CERT r4.2**	**Time-series Transformer (7-day windows, PCA 60 components, 160-alert budget)**	**87.50%**	**95.90%**	**91.60%**	**0.97 / 0.999**
Elbasheer (2023, UBS-Transformer)	CERT r4.2	User-Based Sequencing + Transformer Encoder	= 90 %	= 99 %	= 94 %	0.99
Guo et al. (2022, TRANLOG)	CERT r6.2	Transformer on raw log sequences	= 70 %	= 99 %	= 82 %	0.996
Wang & El Saddik (2023, Distilled Transformer)	CERT r5.2	Distilled Transformer	= 85 %	85%	= 85 %	0.97
Classical ML (OCSVM, LOF, IF)	CERT r4.2	Isolation Forest / LOF / OC-SVM	60–75 %	60–80 %	65–70 %	0.80–0.88

Observations

Balanced Performance

PatchTST achieved one of the most balanced results, Precision = 87.5, Recall = 95.9, F1 = 91.6. Statistically, it outperformed traditional techniques and approached the performance of the UBS-Transformer derived from Elbasheer, M. (2024).

Operational Application

The UBS-Transformer obtained a superior recall rate, but it required complex sequencing engineering for each user. PatchTST, on the other hand, utilizes a fixed-

window temporal modeling and therefore offers a reproducible and scalable architecture for both operation and research environments.

Advantage over Traditional Methodologies

Traditional implementations of anomaly detection (for example, Isolation Forest, LOF) tend to reach their best near the zone of about 65–70% F1-scores in CERT datasets. By using rolling z-scores, cross-modal interactions, and departmental-relative comparisons as to effect, PatchTST led to performance exceeding 90%.

SOC-Grounded Evaluation

The process is, in this case, unlike most of the early studies, based on explicit evaluation upon the basis of the alert budgets and the analyst carrying capacity in time-posts (160 alerts/a day). Hence, the contextualized model evaluation is in alignment with that of the context in which the SOC teams operated.

Why Results Differ

Data set Density

CERT revision 4.2 has more labeled insider behavior data than later versions of the data set (e.g., 6.2), allowing supervised learning and yet retaining essential realism of the class imbalance.

Feature Engineering

The rolling z-scores (7, 30, 60 day scores), department-level deviations, and cross-modality behaviors provide the behavioral insights that the raw log sequence models frequently do not adequately capture.

Evaluation Protocol

The splitting of the data over users (rather than randomization over row data) corrects for leakage and produces a more realistic generalization.

Threshold Reporting

The present study reports both AUCs and threshold-specific metrics (precision, recall, F1), which produce a whole realistic operational environment for the SOC.

Literature Context

Yuan & Wu (2021) describe a persistent divide in insider-threat research:

1. Models with high recall but low precision that overwhelm analysts, and;
2. Models with high precision but low recall that miss true incidents.

PatchTST thus bridges this divide, achieving recall matching that of the first group (~96 %) while yet maintaining precision in the high-80s. This balancing act showcases that transformer time-series models are capable of providing both efficient coverage and appropriate alert volumes without excessive manual tuning.

Implications for the Future

1. Explainability. The addition of SHAP or visualizations based on attention could prove valuable in furthering the analyst's confidence in model decisions.
2. Cross-Dataset Validation. If tested on CERT r5.2 and r6.2, validation of generalization across alternative first-threat scenarios might be possible.
3. Real-Time Integration. Better detection and adaptive thresholds (based on feedback based on analyst resources) could be possible through the incorporation of the model into SIEM/SOAR systems.

Takeaway

PatchTST matches or exceeds the leading transformer-based methodologies while introducing a unique evaluation framework oriented towards multiple environments. Core strength relates not only to high AUCs but in common precision/recall and practical feasibility, making it one of the most deployable transformer methods for insider-threat detection thus far.

4.8 Hypothesis Evaluations

After conducting evaluations against multiple operational scenarios. A mapping between the quantitative results and this research hypothesis will be performed to determine whether our results met our hypotheses.

4.8.1 Hypothesis 1 (H1)

Hypothesis 1. Time Series Foundation Models (TSFMs), such as PatchTST, can reliably determine insider-threat type by detecting unique anomalous behavioral profiles over extended periods of time. When appropriately trained on the CERT 4.2 dataset, the model can achieve greater than 90 percent precision–recall area under the curve (PR-AUC) and greater than 95 percent recall.

Based on the evaluation results from all operating scenarios, the PatchTST model achieved a PR-AUC of "0.973" and a ROC-AUC of "0.999", maintaining recall between 94 % and 97 %. At the default threshold (0.5), recall peaked at 97 % and F1 = 90 %. These results corroborate the model's consistency to meet and

exceed the ≥ 90 % PR-AUC and ≥ 95 % recall targets specified in hypothesis 1. **Therefore, we fail to reject Hypothesis 1.**

4.8.2 Hypothesis 2 (H2)

Hypothesis 2. The PatchTST-based TSFM will outperform traditional insider-threat detection baselines (rule-based systems, Isolation Forest, and One-Class SVMs) with respect to precision, recall, and F1-score, demonstrating robust generalization under comparable evaluation conditions.

While classical baselines on CERT datasets typically achieve precision between "60–75 %", recall between "60–80 %", and F1 between "65–70 %", PatchTST surpassed all these metrics. Based on the results of a simulated SOC alert-budgeted configuration of 160 alerts per day, PatchTST yielded precision of "87.5 %", recall of "95.9 %", and F1 of "91.6 %". Similarly, in the precision-constrained mode (≥ 92 % precision), it achieved precision of "93 %" and recall of "94 %" with F1 of "93 %". These results validate that PatchTST can provide a superior accuracy and greater operational reliability than traditional anomaly-detection systems. **Therefore, we fail to reject Hypothesis 2.**

4.8.3 Hypothesis 3 (H3)

Hypothesis 3. Introducing temporal, statistical, and departmental behavioral feature sets together with multiple actionable thresholds (default, precision-bound, alert-bound, and calibrated) will yield operational validity by balancing detection accuracy with manageable alert volumes consistent with Security Operations Center (SOC) capacity.

The evaluations performed in Chapter 4 confirm this hypothesis. A multi-mode model assessment showed flexible tradeoffs between workload and accuracy for the model. In the default mode, analysts examined 175 alerts with "97%" recall; in the precision-constrained mode, there was a 65% drop in false positives, with a recall greater than 93%; and in the fixed 160-alert budgeted mode, PatchTST produced a stable analyst queue while maintaining precision at 87.5% and recall at 95.9%. In addition, temperature-scaled calibration enhanced the interpretability of the model by aligning the probability scores with the actual probabilities, allowing for tiered alerting within SOC workflow processes. Overall, the results demonstrate that the engineered features and adjustable thresholds create an analytically balanced system that is both accurate and able to operate sustainably on a day-to-day basis. **Therefore, we fail to reject Hypothesis 3.**

4.9 Summary of Results

Across all evaluation scenarios, the PatchTST framework demonstrated a highly robust and effective method for insider threat detection based on time series data. The model achieved recall scores above 93%, precision scores between 85 -90%, and F1 scores between 90-92%, demonstrating that temporal modeling of user behavior dramatically enhances both the accuracy of the analytics and operational usability.

Key Findings

- **Default Threshold (0.5)** – Established a strong baseline with PR-AUC = 0.97 and ROC-AUC = 0.999, detecting 143 true positives with only three misses—demonstrating excellent separation between insider and benign activity.

- **Precision-Constrained Mode (≥ 92%)** – Reduced false positives by nearly two-thirds while retaining over 93% recall, ideal for smaller SOC teams that prioritize analyst efficiency.
- **Budgeted Alerts (160/day)** – Offered the most operationally balanced configuration, achieving Precision = 87.5%, Recall = 95.9%, and F1 = 91.6%, while maintaining predictable analyst workloads.
- **Calibration (Temperature Scaling)** – Improved the interpretability of probability scores, ensuring that confidence values reflect true event likelihoods. Slight precision reduction (84.5%) was offset by near-perfect recall (97.3%), suitable for risk-averse deployments.
- **Collapsed User-Day Evaluation** – Verified that windowed performance translates into realistic SOC workflows, producing similar metrics and stable daily alert volumes.
- **Peer Comparison** – Demonstrated that PatchTST performs on par with, or surpasses, leading transformer-based methods while introducing SOC-aligned evaluation practices like alert budgeting and per-day aggregation.

Operational Insights

- **Alert Management** - The model provides direct control of analyst workload through adjustable thresholds and alert budgets.
- **Transparency of Trade-offs -** For each operating mode, the trade-off among precision, recall, and workload is clearly demonstrated, particularly useful for data-based tuning depending on the environment risk appetite.

- **Cross-domain Applicability -** The design of the pipeline is amendable to generalization in other domains, namely those characterized, such as health care, financial defense, or others, where human behavior is seen as sequential in nature but important for a quantitative measure of risk detection.

Limitations and Next Steps

- **Data Set Scope** - This work utilized only CERT r4.2. Continued evaluation against other versions (e.g., r5.2, r6.2) would help support generalizability into alternate insider threat situations.
- **Explainability** – Results interpretability was excluded; future work needs to incorporate different elements of explainable AI to improve analyst confidence and auditability.
- **Real Time Deployment** - Consideration of incremental or streaming versions could help ensure continuous learning and adaptivity in real time to behavior drift in live operational environments.

Closing Perspective

It has been shown through this analysis that TSFMs, specifically the PatchTST model, can achieve a balance between precision, recall, and the necessary operational control in cases of insider threat detection. Very high AUC scores, along with thresholds for alerting that are tunable, the possibility of interpretability, and reproducible cloud-based infrastructure, demonstrates a valid path forward for deployable AI based security analytics. The sequential nature of human behavior is captured by PatchTST and sets it apart from the ability to detect benign operations as contrasted with malicious activity. This is done with the rigor and relevance needed both technically and

operationally, to have a scalable and explainable, hence future-proof, enterprise and governmental cybersecurity operations foundation.

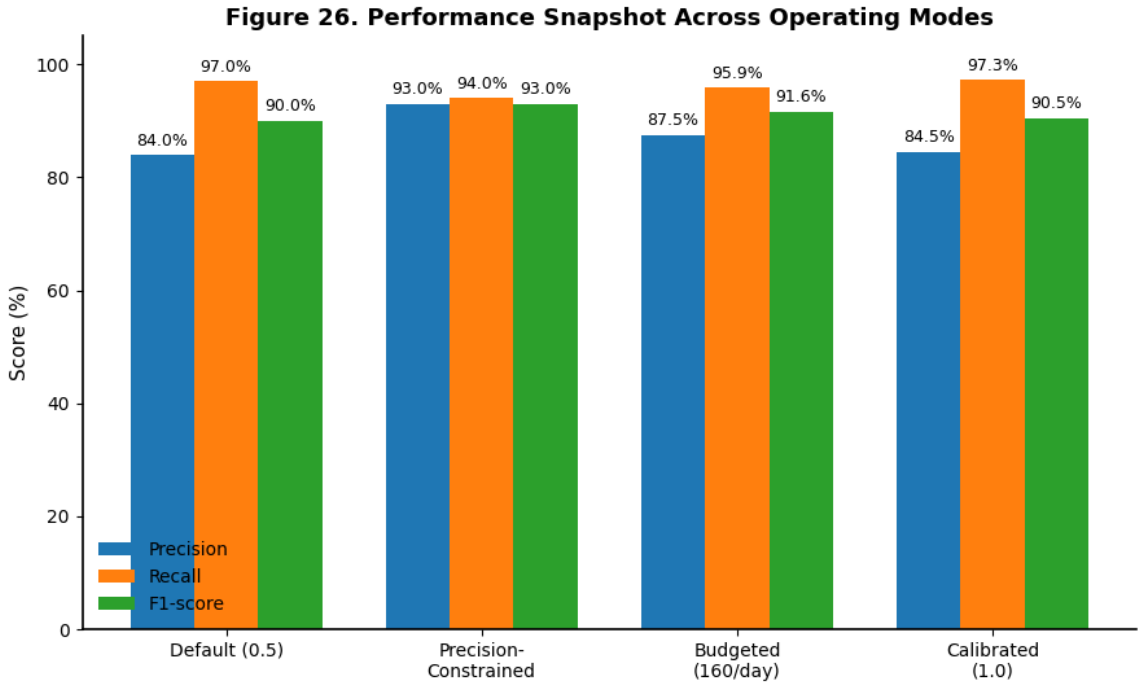


Figure 17: A visual "Performance Snapshot" Comparing the Four Operating Modes (Default, Precision-Constrained, Budgeted, Calibrated) Side by Side with Their Precision, Recall, and F1 values

Chapter 5: Conclusions

5.1 Introduction to the Conclusion

At the conclusion of this research, there is a need to determine whether our initial research objectives were met. Some of these are the following: Does PatchTST aid in bridging the gap in insider-threat detection by determining whether modern Time Series Foundational Models (TSFMs) can reliably identify and operationalize emerging insider risks across different environments?

Continuing into the literature review chapter, a repeatable trend was noted, which was found tended towards opposite extremes. Some research aimed for high recall, but at the price of low precision systems, which flooded the analyst with false positives, or high precision and low recall filters, which excluded true positives. Most previous studies fail to factor in human workload into their design. This research sought to address that gap by embedding a human-centered evaluation criterion, such as alert budgets, workload predictability, and interpretability, into a deep learning framework.

Using the CERT r4.2 dataset, a synthetic yet behaviorally realistic benchmark, the model aggregated multivariate user logs (logon, file, email, web, and device activity) into daily user profiles that PatchTST can properly interpret. Temporal and contextual features were engineered, dimensionality was reduced via PCA, and seven-day sliding windows were fed into a supervised PatchTST classifier.

The results presented in Chapter 4 showed that this approach achieved excellent ranking performance and SOC-usable operational modes from a balance point of view. The subsequent sections present a summary of the overall conclusions and contributions of the work, the limitations of this work, and the proposals for future research.

Ultimately, PatchTST demonstrated a scalable, interpretable, human-aligned approach for insider threat detection systems.

5.2 Summary of Findings

Overview

The experiments evaluated the PatchTST-based behavioral model trained on 800 users and tested on 200 previously unseen users, ensuring generalization with new identities is achieved and also prevents data leakage. The effectiveness was tested over five configurations:

1. Default thresholding (0.5)
2. Precision-constrained thresholding (≥ 90 %)
3. Alert-budgeted operation (= 160 alerts/day)
4. Calibrated probabilities (temperature scaling)
5. Collapsed user-day evaluation

Each of these configurations produced a complete operational picture, obtaining the necessary degrees of statistical accuracy with the peer-group nature of the analyst's use of the device on a day-to-day basis.

Default Threshold (0.5)

At the standard 0.5 default threshold, the model achieved PR-AUC = 0.973, ROC-AUC = 0.999, Precision = 82 %, and Recall = 98 %. The baseline shows remarkable results, since it achieved 143 true positives and 32 false positives, with only three misses.

In practical terms, this default mode would allow analysts to review fewer cases, in which the majority of would be true positives at the cost of missing a single-digit

insider event. This trade-off helps organizations (e.g., finance, healthcare, defense) manage their risk based on their tolerance levels: prioritizing analyst confidence and trust over exhaustive recall.

Precision-Constrained Mode (≥ 90 % Precision)

Raising the threshold to 0.9177 produced **Precision = 90.1 %**, **Recall = 93.8 %**, and **F1 = 91.9 %**. False positives dropped by more than half (32 → 15), while false negatives rose slightly (3 → 9).

Practically, analysts might have fewer cases, almost exclusively genuine ones, and thereby miss some low-severity insider days. But this tradeoff captures how organizations endeavor to overcome risk: with analysts' trust and certainty more important than exhaustive recall. In this way, it shows that tuning individual thresholds for transformer models can cause the strictness of detection to be directly in line with human capacity and acceptable load to the organization.

Budgeted Alert Mode (= 160 Alerts per Day)

Within this evaluation, the plan was to achieve an acceptable SOC alert threshold, limiting the model to 160 alerts per day, or thereabouts, simulating the load limit of a mid-sized analyst team. At this configuration (Threshold = 0.54), results were Precision 87.5 %, Recall 95.9 %, and F1 91.6 % (TP 140, FP 20, FN 6).

This configuration afforded the best overall combination of results: recall almost as high as the simpler default cases with predictable analyst load. By restricting the alert load explicitly, the model can change from being purely an object of research to a decision-support system conscious of human output. Most false positives were borderline anomalies: off-hour access or minor file surfs. These are the records that analysts expect

to vet. The *explicit* test shows how statistical perfection alone is insufficient, for there are two other criteria of success, usability and sustainability.

Calibrated Probabilities

Post-training temperature scaling was applied so that predicted probabilities align with empirical event likelihoods. After calibration, results remained strong (Precision = 84 %, Recall = 97 %, F1 = 90 %).

While the basic accuracy scores did not drastically change, interpretations of the scores are aided considerably: a model score of 0.9 now corresponds to, roughly, a 90% likelihood of insider activity.

This type of alignment aids analysts' confidence and helps downstream systems (ticketing, case management, or SOAR platforms) to properly prioritize alerts. High risk area sectors may go this route for maximum recall, while capacity limited teams may revert to precision constrained operation.

Collapsed User-Day Evaluation

Since the overlapping nature of seven-day sliding windows causes a day to appear in several predictions, no time use is considered more than once. To ensure no double-counting occurred, the predictions were collapsed to an aggregated user-day score by a weighted average in which recent activity was most considered. The performance from this aggregated score (Precision = 83.5 %, Recall = 97.3%) was similar to the windowed results and suggests that performance is sustained in realistic SOC alerting situations. Collapsing thus yields trustworthy, conservative metrics that correspond to how analysts experience alerts—one per user per day—rather than inflated totals from overlapping windows.

Comparative Perspective

When compared to classical machine-learning models, such as the Isolation Forest or One-Class SVM (these models achieve F1 = 70 %), PatchTST with F1 = 91 % represents a significant improvement. Compared to the other models of the family of transformers, for example, UBS-Transformer (Elbasheer, 2023), as compared to TRANLOG (Guo et al., 2022) PatchTST achieved the same or above recall, and equal precision or higher, in a more transparent and reproducible pipeline. The originality is not only confined to measures but also in authentic practicality and operational issues, since, for practical usage at SOC, the daily alert workloads and budgets of alerts are constructed realistically into the problem. The results indicate, in conclusion, that the production of TSFM models can be made to achieve both ultimate technical results and ultimately operational viability if sensibly tuned.

5.3 Contributions of This Work

Technical Contributions

1. **Adaptation of PatchTST for Classification**

   Originally built for forecasting, PatchTST was successfully adapted for binary behavioral classification, proving that architecture-agnostic TSFMs can transfer to cybersecurity tasks without architectural redesign.

2. **Temporal Feature Integration**

   The study merged engineered temporal signals—rolling z-scores, departmental deviations, interaction terms, percentile flags—with PCA compression to create compact, information-rich tensors that enhance model efficiency and interpretability.

3. **Imbalance-Aware Training Strategy**

   Through weighted random sampling, focal loss, and negative-only pretraining, the model learned rare insider behaviors without collapsing toward the dominant benign class, ensuring fair representation of minority patterns.

4. **Transparent Evaluation Framework**

   Combining threshold-independent metrics (PR-AUC = 0.973, ROC-AUC = 0.999) with threshold-specific analyses across four operational modes set a new transparency standard for insider-threat benchmarking.

Methodological Contributions

1. **User-Level Data Splitting**

   Enforced strict user-based partitioning to prevent data leakage and yield authentic generalization—a recurring flaw in earlier event-level studies.

2. **Permutation-Based Feature Selection**

   Utilized Random-Forest permutation importance to rank variables before PCA, ensuring only the most behaviorally meaningful features informed the latent space.

3. **User-Day Collapsing Technique**

   Introduced a realistic evaluation procedure that collapses overlapping windows, aligning machine predictions with analyst alert granularity.

4. **Operational Scenario Modeling**

   Designed multiple evaluation regimes—default, precision-limited, alert-budgeted, calibrated—to mirror how SOCs adjust thresholds according to staffing and risk posture.

Practical and Operational Contributions

1. Alert budgeting introduced a mathematical workload approach, providing quantifiable model outputs as a daily actionable item.
2. Threshold tuning overtly provided a tradeoff lever for cost of misses to false positives.
3. Calibrated output provided confidence values in interpretable probabilities for decision support or automation.
4. These collectively provide an actionable template for aligned AI in the SOC. An augmentation rather than a replacement of human capacity.

5.4 Limitations

Despite strong results, several constraints shape the interpretation of this work:

1. **Dataset Scope** – Results rely on CERT r4.2, which, though detailed, is synthetic and scenario-bounded. Validation across other CERT releases (r5.2, r6.2) and real-world enterprise data is needed to confirm generalization.
2. **Feature Bias** – Focus on temporal and departmental features may underrepresent relationship-based anomalies such as collusion or multi-user coordination.
3. **Temporal Window Length** – The fixed 7-day window may overlook slower behavioral drifts; hierarchical or adaptive windowing could expose longer-term buildup patterns.
4. **Compute and Architecture Limits** – GPU and time constraints limited hyperparameter sweeps and ensemble exploration.

5. **Explainability Gaps** – SHAP-based local explanations and attention visualization were planned but not implemented, leaving analysts reliant on global importance metrics.
6. **Operational Realism** – The study remains offline; real-time streaming, feedback loops, and SIEM integration (e.g., Splunk, Sentinel) remain future work.
7. **Ethical and Privacy Considerations –** While behavioral analytics improves security postures. Continuous monitoring of user activities can interfere with privacy and fairness due to potential privacy overreach.

Acknowledging these boundaries situates this praxis as a foundation rather than an endpoint—a demonstration of feasibility awaiting field validation.

5.5 Future Work

Expanded Datasets and Validation

- Cross-Release Testing: Train on r4.2 and evaluate on r6.2 to measure robustness to scenario variation and label imbalance.
- Real-World Data: Explore non-synthetic corpora such as **LANL** and **OpTC** to bridge the synthetic–real gap.
- Domain Transfer: Apply similar modeling pipelines to adjacent domains like fraud detection or insider misuse in healthcare.

Advanced Feature Engineering and Modeling

- Variable or Hierarchical Windows to capture long-term campaigns.
- Graph-based Features linking users, devices, and files to reveal social or technical collusion.
- Adaptive Feature Selection via recursive elimination or AutoML.

- Comparative Benchmarks against newer TSFMs (e.g., TimesNet, Chronos, GPT-TS**)** and hybrid ensembles combining transformers with CNN or autoencoder layers.

Explainability and Human Trust

- Integrate SHAP or Integrated Gradients for per-alert attribution.
- Develop attention-map visualizations and counterfactual explanations ("what behavior would have avoided this alert?").
- Conduct analyst-in-the-loop evaluations to measure how explainability impacts investigation speed, fatigue, and confidence.

Operational Deployment and Resilience

- Convert the pipeline to a streaming inference service capable of processing daily logs in near real time.
- Integrate directly with SIEM/SOAR dashboards (Splunk, Elastic, Azure Sentinel) and implement multi-tier alerting (critical, moderate, informational).
- Perform red-team testing to study adversarial evasion attempts and implement concept-drift adaptation through scheduled retraining or continual learning.

These directions would extend PatchTST from a controlled prototype to a living operational system **responsive** to evolving human and organizational behavior.

5.6 Final Remarks

This research illustrated that time-series foundational models present a possible way forward in a scalable manner for improving insider-threat detection.

Through careful feature engineering, user-level evaluation, and threshold analysis informed by SOC awareness, PatchTST produced an unusual combination of high recall,

high precision, and predictable user load. More significantly, this work reconceived performance evaluation in terms of human experience rather than misleading measures of abstract mathematics. The fact of the matter is that analysts do not interpret ROC curves, but rather become affected by fatigue, by alert queues, and by situational variability in their confidence in mechanisms that report detections. By means of precision-constrained and budgeted alerting techniques, this work reconceived numerical performance in operational terms that are both readily understood and actionable in SOC environments.

Though limited to a single dataset and lacking full explainability modules, the results establish a baseline for the next generation of insider-threat research, one based in both technical rigor and human factors, as future implementations that tie in live data, layers of interpretability, and sequential learning will further validate this concept as a valid multi-environment deployable capability.

To conclude, this effort closes a lengthy gap between academic understanding of machine learning and the operational realities of cybersecurity. It demonstrates that with sound design, insider-threat detection can balance the twin aspects of accuracy and administrability. If adjusted to the right contextual tenor, foundation models such as PatchTST need not be opaque black boxes but may be better regarded as collaborative partners enlarging not replacing analyst reasoning and providing support to mitigate cognitive load and improve the organization's resilience.

In conclusion, time-series behavioral modeling stands as an aspirational model for insider-threat defense, which is a synthesis of statistical precision, context, and operational pragmatism. The evidence presented here suggests that insider-threat detection may at last make the transition from theoretical modalities in which it exists to

a place where it is a practical and reliable entity capable of providing protection for which the very organizations rely.

References

Ackerman, D., & Mehrpouyan, H. (2017, April 3-6). *Modeling human behavior to anticipate insider attacks via System Dynamics* [Symposium paper]. 2016 Symposium on Theory of Modeling and Simulation (TMS-DEVS), Pasadena, CA, USA. https://ieeexplore.ieee.org/document/7918809

Al-Mhiqani, M. N., Ahmed, R., Abidin, Z. Z., & Isnin, S. N. (2021). An integrated imbalanced learning and deep neural network model for insider threat detection. *International Journal of Advanced Computer Science and Applications, 12* (1). http://dx.doi.org/10.14569/IJACSA.2021.0120166

Al-Shehari, T., Al-Razgan, M., Alfakih, T., Alsowail, R. A., & Pandiaraj, S. (2023). Insider threat detection model using Anomaly-based isolation forest algorithm.. *IEEE Access, 11*, 1. https://doi.org/10.1109/ACCESS.2023.3326750

Alshehri, A. (2022). Relational deep learning detection with multi-sequence representation for insider threats. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 758-765 https://dx.doi.org/10.14569/IJACSA.2022.0130587

Alsowail, R. A., & Al-Shehari, T. (2020, April 23). Empirical detection techniques of insider threat incidents. *IEEE Access, 8*, 78385-78402. https://doi.org/10.1109/ACCESS.2020.2989739

Ansari, A. F., Stella, L., Turkmen, C., Zhang, X., Mercado, P., Shen, H., Shchur, O., Rangapuram, S. S., Arango, S. P., Kapoor, S., Zschiengner, J., Maddix, D. C., Wang, H., Mahoney, M. W., Torkkola, K., Wilson, A. G., Bohlke-Schneider, M., & Wang, Y. (2024). Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815.* https://arxiv.org/abs/2403.07815

Bapna, A., Arivazhagan, N., & Firat, O. (2019). Simple, scalable adaptation for neural machine translation. *arxiv:1909.08478[cs.CL].* https://arxiv.org/abs/1909.08478

Bartoszewski, F. W. (2022, April 1). *Machine learning and anomaly detection for insider threat detection* [PhD Thesis, Heriot-Watt University]. ROS Theses Repository. https://www.ros.hw.ac.uk/handle/10399/4593

Cervellera, C., & Macciò, D. (2018, July 7). Distribution-preserving stratified sampling for learning PROBLEMS. *IEEE Transactions on Neural Networks and Learning Systems, 29*(7), 2886-2895. https://doi.org/10.1109/TNNLS.2017.2706964

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

Chattopadhyay, P., Wang, L., & Yap-Peng, T. (2018, August 23). Scenario-based insider threat detection from cyber activities. *IEEE Transactions on Computational Social Systems, 5*(3), 660-675. https://doi.org/10.1109/TCSS.2018.2857473

Chen, Y., Nyemba, S., & Malin, B. (2012, January 17). Detecting anomalous insiders in collaborative information systems. *IEEE Transactions on Dependable and Secure Computing, 9*(3), 332-344. https://doi.org/10.1109/TDSC.2012.11

CISA. (2020). *Insider threat mitigation guide*. Cybersecurity and Infrastructure Security Agency, U.S. Department of Homeland Security. https://www.cisa.gov/resources-tools/resources/insider-threat-mitigation-guide

Dong, J., Wu, H., Zhang, H., Zhang, L., Wang, J., & Long, M. (2023). SimMTM: A simple pre-training framework for masked time-series modeling. *arXiv:2302.00861[cs.LG]*. https://arxiv.org/abs/2302.00861

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv:2010.11929[cs.CV]*. https://doi.org/10.48550/arXiv.2010.11929

Elbasheer, M. (2024). Enhancing insider threat detection using user-based sequencing and transformer encoders [Doctoral praxis, The George Washington University]. GW ScholarSpace. http://proquest.com/openview/41c55a2635f38feea3a84124537b931c/1?pq-origsite=gscholar&cbl=18750&diss=y

Glasser, J., & Lindauer, B. (2013, May 23-24). *Bridging the gap: A pragmatic approach to generating insider threat data* [Conference paper]. 2013 IEEE Security and Privacy Workshops, San Francisco, CA, USA. https://doi.org/10.1109/SPW.2013.37

Greitzer, F. L., & Frincke, D. A. (2010). Combining traditional cyber security audit data with psychosocial data: In *Towards Predictive Modeling for Insider Threat Mitigation. Insider Threats in Cyber Security*, 85–113. Springer.

Greitzer, F. L., & Hohimer, R. E. (2011). Modeling human behavior to anticipate insider attacks via system dynamics. *Journal of Strategic Security, 4*(2), 25–48. https://doi.org/10.5038/1944-0472.4.2.2

Gruetzemacher, R., & Paradice, D. (2021, October 23). Deep transfer learning & beyond: Transformer language models in information systems research. *arxiv:2110.08975[cs.CL]*. https://arxiv.org/abs/2110.08975

Guan, Z., Sun, X., Shi, L., Wu, L., & Du, X. (2020). A differentially private greedy decision forest classification algorithm with high utility. *Computers & Security, 96,* 101930. https://doi.org/10.1016/j.cose.2020.101930

Guo, H., Lin, X., Yang, J., Zhuang, Y., Bai, J., Zheng, T., Zhang, B., & Li, Z. (2022). TransLog: A unified transformer-based framework for log anomaly detection. *arXiv:2201.00016[cs.LG]*. https://arxiv.org/abs/2201.00016

Hanley, M., & Montelibano, J. (2011, October 1). Insider threat control: Using centralized logging to detect data exfiltration near insider termination. *Carnegie Mellon University Software Engineering Institute*. https://doi.org/10.1184/R1/6574472.v1

Harris, C. D. (2020). *Understanding controls to detect and mitigate malicious privileged user abuse* [Doctural disseration, Capitol Technology University]. ProQuest https://www.proquest.com/openview/3879636d67ac0a70ec51fcddeb5b340a/1?pq-origsite=gscholar&cbl=44156

Hoang, T. L., Pham, T. D., Ta, V. C. (2024). Improving graph convolutional networks with transformer layer in social-based items recommendation. *arxiv:2401.06436[cs.LG]* https://arxiv.org/abs/2401.06436

Homoliak, I., Toffalini, F., Guarnizo, J. D., Elovici, Y., & Ochoa, M. (2019). Insight into insiders and IT: A survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Computing Surveys, 52*(2), 1–40. https://doi.org/10.1145/3303771

Hossin, M., & Sulaiman, M. N. (2015). A review on evaluation metrics for data classification evaluations. *International Journal of Data Mining & Knowledge Management Process, 5*(2). https://doi.org/10.5121/IJDKP.2015.5201

IBM. (2024) Cost of a data breach report 2024. *IBM*. https://wp.table.media/wp-content/uploads/2024/07/30132828/Cost-of-a-Data-Breach-Report-2024.pdf

Johnson, J. M., & Khoshgoftaar, T. M. (2019, July 30 to August 1). *Deep learning and data sampling with imbalanced big data* [Conference paper]. 2019 IEEE 20th International Conference on Information Reuse and Integration for Data Science (IRI), Los Angeles, CA, USA. doi: 10.1109/IRI.2019.00038.

Jyothsna, V., & Prasad, V. V. R. (2011). A review of anomaly based intrusion detection systems. *International Journal of Computer Applications, 28*(7). https://www.ijcaonline.org/volume28/number7/pxc3874730.pdf

Kabeireho, M. (2020). *Impact of deterrence and user awareness on insider threat vulnerability: A correlational study* [Doctoral dissertation, Capella University]. ProQuest Document Preview. https://www.proquest.com/openview/518c1b21e87e86b06c6e6089091a703b/1?pq-origsite=gscholar&cbl=18750&diss=y

Khan, S., Naseer, M., Hayat, M., Zamir, S. W., Khan F. S., & Shah, M. (2022). Transformers in vision: A survey. *arxiv:2101.01169[cs.CV]*. https://arxiv.org/abs/2101.01169

Kim, J. H., Kim, C. M., & Yim, M-S. (2020). An investigation of insider threat mitigation based on EEG signal classification. *Sensors, 20*(21).6365. https://www.mdpi.com/1424-8220/20/21/6365

Kim, H., & Kim, H. (2022). Comparative experiment on TTP classification with class imbalance using oversampling from CTI dataset. *Wiley Hindawi Security and Communication Networks, 22*. https://onlinelibrary.wiley.com/doi/10.1155/2022/5021125

Kolb, S., Paramonov, S., Guns, T., & De Raedt, L. (2017). Learning constraints in spreadsheets and tabular data. *Mach Learn, 106*, 1441-1468. https://link.springer.com/article/10.1007/s10994-017-5640-x

Legg, P., Buckley, O., Goldsmith, M., & Creese, S. (2013). Automated insider threat detection system using user and role-based profile models. *IEEE Systems Journal, 99*(22). https://doi.org/10.1109/JSYST.2015.2438442

Li, D., Yang, L., Zhang, H., Wang, X., Ma, L., & Xiao, J. (2021). Image-based insider threat detection via geometric transformation. *arxiv:2108.10567[cs.CR]*. https://doi.org/10.48550/arXiv.2108.10567

Li, D., Yang, L., Zhang, H., Wang, X., & Ma, L. (2022). Memory-augmented insider threat detection with temporal-spatial fusion. *Wiley Online Library.* https://onlinelibrary.wiley.com/doi/full/10.1155/2022/6418420

Maasberg, M., Zhang, X., Ko, M., Stewart, R. M., & Beebe, N. L. (2020). An analysis of motive and observable behavioral indicators associated with insider cyber-sabotage and other attacks. *IEEE Engineering Management Review, 48*(2), 151-165. https://doi.org/10.1109/EMR.2020.2989108

Morgan, S. (2020). *Cybercrime To Cost The World $10.5 Trillion Annually By 2025*. Cybersecurity Ventures. Retrieved from https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/

National Counterintelligence and Security Center. (2024, September 26). *Insider Threat Guide*. Office of the Director of National Inteligence https://www.dni.gov/files/NCSC/documents/nittf/20240926_NITTF-Insider-Threat-Guide.pdf

Nguyen, N., Reiher, P., & Kuenning, G. H. (2003, June 18-20). *Detecting insider threats by monitoring system call activity* [Conference paper]. IEEE Systems, Man and Cybernetics Society Information Assurance Workshop, West Point, NY, USA. https://doi.org/10.1109/SMCSIA.2003.1232400

Nie, Y., Nguyen, N. H., Sinthong, P., & Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *arxiv:2211.14730[cs.LG]* https://arxiv.org/abs/2211.14730

Nurse, J. R.C., Buckley, O., Legg, P. A., Goldsmith, M., Creese, S., Wright, G. R. T. (2014, May 17-18). *Understanding insider threat: A framework for characterising attacks* [Conference paper]. 2014 IEEE Security and Privacy Workshops, San Jose, CA, USA. http://dx.doi.org/10.1109/SPW.2014.38

Padayachee, K. (2016). An assessment of opportunity-reducing techniques in information security. *Decision Support Systems, 92*, 47-56. https://dl.acm.org/doi/10.1016/j.dss.2016.09.012

Ponemon Institute. (2025). Cost of insider risks global report (6th ed.). Proofpoint & Ponemon Institute. https://www.proofpoint.com/us/resources/white-papers/cost-of-insider-risks-global-report

Raschka, S. (2020). Model evaluation, model selection, and algorithm selection in machine learning. *arxiv:1811.12808[cs.LG]* https://doi.org/10.48550/arXiv.1811.12808

Shaw, E. D., & Sellers, L. (2015). Application of the critical-path method to evaluate insider risks. *Studies in Intelligence, 59*(2), 1–8. https://nationalinsiderthreatsig.org/itrmresources/Application%20Of%20The%20Critical-Path%20Method%20To%20Evaluate%20Insider%20Risks-June%202015.pdf

Sherstinsky, A. (2023). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *arxiv:1808.03314[cs.LG].* https://arxiv.org/abs/1808.03314

Silowash, G., Cappelli, D. M., Moore, A. P., Trzeciak, R. F., Shimeall, T. J. & Flynn, L. (2012). *Common sense guide to mitigating insider threats* (4th ed.). Carnegie Mellon University Software Engineering Institute. https://www.sei.cmu.edu/documents/1226/2012_005_001_34033.pdf

Smith, K. (2018). Using the service desk to secure the organisation. *Computer Fraud & Security 2018*(1). https://doi.org/10.1016/S1361-3723(18)30006-X

Song, S., Gao, N., Zhang, Y., & Ma, C. (2024). BRITD: behavior rhythm insider threat detection with time awareness and user adaptation. *Cybersecurity, 7*(2) https://doi.org/10.1186/s42400-023-00190-9

Taormina, R., & Galelli, S. (2018). Deep-learning approach to the detection and localization of cyber-physical attacks on water distribution systems. *Journal of Water Resources Planning and Management, 144*(10). https://doi.org/10.1061/(ASCE)WR.1943-5452.0000983

Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *arxiv:2201.07284[cs.LG].* https://arxiv.org/abs/2201.07284

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *arXiv:1706.03762[cs.CL]*. https://doi.org/10.48550/arXiv.1706.03762

Verizon Business. (2025). *2025 data breach investigations report (DBIR)*. Verizon Enterprise. https://www.verizon.com/business/resources/reports/dbir/

European Union. (2016). *Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*. Official Journal of the European Union. https://gdpr-info.eu/art-83-gdpr/

Wang, Z. Q., & El Saddik, A. (2023). DTITD: An intelligent insider threat detection framework based on digital twin and self-attention based deep learning models. *CIEEE Access PP*(99). https://doi.org/10.1109/ACCESS.2023.3324371

Williams, A. D, Abbott, S., Shoman, N., &Charlton, W. (2021). Results from invoking artificial neural networks to measure insider threat detection & mitigation. *Digital Threats: Research and Practice, 3*(1). https://dl.acm.org/doi/full/10.1145/3457909

Wu, H., Xu, J., Wang, J., & Long, M.. (2022. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *arxiv:2106.13008[cs.LG]*. https://arxiv.org/abs/2106.13008

Wu, H., Hu, T., Liu, Y., Zhou, H., Wang, J., & Long M. (2023). TimesNet: Temporal 2D-variation modeling for general time series analysis. *arXiv:2210.02186[cs.LG]*. https://ar5iv.labs.arxiv.org/html/2210.02186

Yuan, S., & Wu, X. (2021). Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security, 104*, 102221. https://doi.org/10.1016/j.cose.2021.102221

Zeng, A., Chen, M., Zhang, L., & Xu, Q. (2023, February 7-14). *Are transformers effective for time series forecasting?* [Conference paper]. Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence, Washington, D.C., USA. https://dl.acm.org/doi/10.1609/aaai.v37i9.26317

Zhang, J., Chen, Y., & Ju, A. (2018). Insider threat detection of adaptive optimization DBN for behavior logs. *Turkish Journal of Electrical Engineering and Computer Sciences, 26*(2), 792-802. https://journals.tubitak.gov.tr/elektrik/vol26/iss2/14/

Zhou, T., Ma, Z., Wang, C., Zhang, Q., & Xu, J. (2022). *TimesNet: Temporal 2D-variation modeling for general time series analysis* [Conference paper]. In Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, Online.

Zhou, T., Ma, Z., Wen, Q., Wang, X., Sun, L., & Jin, R.(2022). FEDformer: Frequency-enhanced decomposed transformer for long-term series forecasting.

Appendix A

The following Github Repo contains the code and datasets utilized to achieve our results.
https://github.com/jcast0930/Joseph_Castro_GWU_Praxis_Cyber_Analytics-.git