Log2Sig: Frequency-Aware Insider Threat Detection via Multivariate Behavioral Signal Decomposition

Kaichuan Kong[a], Dongjie Liu[a,*], Xiaobo Jin[b], Zhiying Li[a], Guanggang Geng[a]
[a]College of Cyber Security, Jinan University, Guangzhou, China
[b]School of Advanced Technology, Xi'an Jiaotong-Liverpool University, Suzhou, China
willkkc123@gmail.com, {djliu, gggeng}@jnu.edu.cn, tz982354814@163.com, xiaobo.jin@xjtlu.edu.cn

***Abstract*—Insider threat detection presents a significant challenge due to the deceptive nature of malicious behaviors, which often resemble legitimate user operations. However, existing approaches typically model system logs as flat event sequences, thereby failing to capture the inherent frequency dynamics and multiscale disturbance patterns embedded in user behavior. To address these limitations, we propose Log2Sig, a robust anomaly detection framework that transforms user logs into multivariate behavioral frequency signals, introducing a novel representation of user behavior. Log2Sig employs Multivariate Variational Mode Decomposition (MVMD) to extract Intrinsic Mode Functions (IMFs), which reveal behavioral fluctuations across multiple temporal scales. Based on this, the model further performs joint modeling of behavioral sequences and frequency-decomposed signals: the daily behavior sequences are encoded using a Mamba-based temporal encoder to capture long-term dependencies, while the corresponding frequency components are linearly projected to match the encoder's output dimension. These dual-view representations are then fused to construct a comprehensive user behavior profile, which is fed into a multilayer perceptron for precise anomaly detection. Experimental results on the CERT r4.2 and r5.2 datasets demonstrate that Log2Sig significantly outperforms state-of-the-art baselines in both accuracy and F1 score.**

***Index Terms*—Insider Threat Detection, Signal Decomposition, MVMD, Mamba, Multivariate Log Representation, User Behavior Analysis**

I. Introduction

Insider threats have emerged as a pressing security issue in enterprise information systems due to their stealthy nature, prolonged attack cycles, and fragmented behavioral patterns. Unlike external attackers, insiders typically possess legitimate credentials and authorized access to internal systems, enabling them to bypass traditional perimeter defenses and camouflage malicious behaviors as routine operations [1]. According to the 2025 Ponemon Institute Global Cost of Insider Risk report [2], organizations encounter an average of 23 insider-related incidents annually, with most attacks taking weeks or even months to detect and contain. These low-frequency, multi-stage, and covert threats present significant challenges to detection mechanisms, particularly in terms of temporal modeling and fine-grained behavioral analysis.

In existing research, insider threat detection is primarily addressed via behavior modeling based on machine learning and deep learning techniques. Traditional machine learning methods extract statistical features, such as login frequency and file access counts, and employ classifiers such as logistic regression (LR), random forest (RF), and XGBoost [3]–[5] to identify anomalous behaviors. As the sequential and contextual nature of user activities gains increasing attention, deep learning models including LSTM, Transformer, and graph neural networks have been widely adopted for insider threat modeling [6]–[9], thereby improving the ability to capture complex behavior representations and contextual dependencies.

Despite recent advances in modeling accuracy, current approaches still face two key challenges in insider threat detection. **One challenge lies in modeling behavioral frequency perturbations across activity types**. Insider threats often involve gradual shifts between behavior types, accompanied by evolving frequency patterns. Existing methods based on event counts or discrete sequences struggle to capture such cross-type frequency dynamics, leading to missed threat cues. **Another challenge is achieving efficient detection over long behavior sequences**. As insider attacks typically span extended time windows, deep models such as Transformers and graph neural networks incur high computational costs when processing long logs, limiting their deployment in latency-sensitive or resource-constrained environments.

To address these challenges, we propose Log2Sig, a novel insider threat detection framework that integrates frequency-aware modeling with efficient sequential representation learning. Log2Sig transforms raw user activity logs into multivariate temporal signals and applies Multivariate Variational Mode Decomposition (MVMD) [10] to jointly decompose system-level signals into intrinsic mode functions (IMFs) that reveal behavioral rhythms and multi-scale perturbation patterns. In parallel, we adopt the Mamba [11] architecture, a structured state space model, as the sequence encoder to capture long-range behavioral dependencies with linear-time complexity, ensuring both strong expressive capacity and deployment efficiency. Finally, we combine the decomposed frequency features with the original event sequences to construct a joint input representation. This enables the model to simultaneously learn temporal dynamics and frequency-domain anomalies, thereby improving detection accuracy and scalability.

Our key contributions are summarized as follows:

- **Log2Sig** is proposed as the first framework to model user activity logs as multivariate frequency signals. By intro-

[*] Corresponding author.

ducing a frequency-aware representation, the framework enables the detection of subtle and multiscale anomalies that are often missed by conventional sequence-based methods.

- A frequency decomposition module based on MVMD is developed to extract IMFs across behavioral channels. This approach captures both periodic patterns and non-stationary anomalies, and addresses a core limitation in existing work, which is the inability to model cross-behavioral frequency dynamics.
- A dual-view encoding strategy is proposed to model both behavior sequences and frequency-decomposed signals. The former is modeled using a Mamba-based temporal encoder to capture long-range dependencies with linear-time complexity, while the latter is linearly projected to align with the sequence encoding. This design enhances the framework's ability to capture both temporal dynamics and frequency-aware variations over extended time windows.
- Extensive experiments on CERT r4.2 and r5.2 datasets demonstrate the superiority of Log2Sig over state-of-the-art baselines. Ablation and robustness studies further confirm the individual contribution of each module and the overall effectiveness of the proposed framework.

The remainder of this paper is organized as follows. Section II reviews related work on insider threat detection and signal-based modeling. Section III introduces the preliminaries of multivariate signal decomposition and the Mamba encoder. Section IV presents the proposed Log2Sig framework, including behavior representation, signal decomposition, encoding, and classification. Section V outlines the experimentaldataset, ssetup, baselines, and evaluation metrics. Section VI reports the empirical results and sensitivity analysis. Finally, Section VII concludes the paper and discusses future directions.

II. Related Work

In this section, we review relevant literature in two primary dimensions: (i) insider threat detection based on machine learning and deep learning architectures, and (ii) signal-based methods for behavioral sequence modeling.

A. Insider Threat Detection

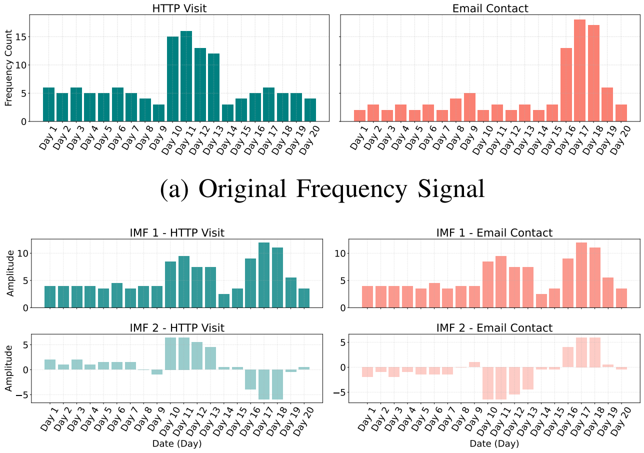
Insider threat detection has increasingly benefited from machine learning and deep learning approaches, which enable the extraction of temporal, semantic, and structural patterns from user behavior logs. This section presents representative techniques, organized into classical machine learning models and deep learning-based sequential architectures.

1) Classical Machine Learning Approaches: Traditional machine learning methods for insider threat detection primarily rely on discriminative feature extraction from structured audit logs. Liu *et al.* [3] proposed Log2vec, a hybrid framework that combines heterogeneous graph embeddings with heuristic rule modeling to capture latent user behavior in enterprise contexts. Le *et al.* [4] conducted a comparative evaluation of supervised models, including Logistic Regression (LR), Random Forest (RF), and XGBoost, demonstrating that engineered behavioral features can be effectively mapped to risk scores. Beyond model selection, feature engineering and reduction have proven critical to robustness. Bin *et al.* [5] applied Information Gain (IG) and Correlation-Based Feature Selection (CFS) to eliminate redundancy and improve interpretability. Their findings showed that RF and SVM consistently achieve strong accuracy and generalization across feature subsets. In unsupervised scenarios where labeled data is limited or unavailable, anomaly detection techniques have gained traction. Bartoszewski *et al.* [12] compared unsupervised models such as Local Outlier Factor (LOF), one-class SVMs (OCSVM), Isolation Forest (IForest), and HMM under both ensemble and single-model settings, emphasizing deployment feasibility using CERT datasets. Le *et al.* [13] further proposed an autoencoder-based reconstruction method for high-dimensional behavior vectors, while Yousef *et al.* [14] employed Isolation Forest to efficiently capture outliers in temporal user logs.

Although these approaches offer strong baselines, their reliance on handcrafted features and limited temporal modeling capacity has motivated the shift toward deep learning.

2) Deep Learning-Based Approaches: Recent advancements in deep learning have enabled more expressive representations of user behavior, improving insider threat detection through modeling of sequential, semantic, and structural dependencies. Early efforts predominantly addressed temporal patterns. He *et al.* [6] introduced an attention-augmented LSTM framework designed to highlight critical behavioral transitions. Building on this, Huang *et al.* [7] combined pre-trained BERT embeddings with a bidirectional LSTM to jointly learn contextual semantics and sequential evolution. Pal [15] employed LSTM and GRU networks for temporal representation learning, while Xiao *et al.* [16] integrated CNNs to extract statistical features and Transformers to capture long-range chronological dependencies. Beyond sequential modeling, representation enhancement techniques have emerged. Budžys *et al.* [17] proposed GAFMAT, applying Gabor filtering to transform keystroke dynamics into time–frequency representations, thereby improving CNN-based identity modeling. Concurrently, Gayathri *et al.* [18] developed SPCAGAN, a GAN-based framework that generates synthetic insider activity traces via linear manifold learning, mitigating data scarcity in security contexts. To further capture higher-order relational and structural dependencies, recent works have adopted graph-based paradigms. Xiao *et al.* [9], Roy *et al.* [8], and Cai *et al.* [19] leveraged graph neural networks (GNNs) to jointly model temporal dynamics and inter-user relationships embedded in behavior graphs.

Complementing architectural advances, large language models (LLMs) have recently emerged as a versatile paradigm for log-based anomaly detection. LogGPT [20] and LogPrompt [21] utilize handcrafted prompts for zero-shot or few-shot detection using pre-trained LLMs. In contrast, fine-tuning approaches [22] adapt LLMs to specific behavioral distribu-



(a) Original Frequency Signal

(b) MVMD-Decomposed Frequency Signal

Fig. 1. Illustrative example of MVMD-based decomposition of user behavior frequency signals.

tions, enhancing alignment and performance under domain shifts.

Despite the architectural advances and high detection accuracy, many deep learning models suffer from high computational overhead due to complex encoding and training processes.

B. Signal-Based Modeling for Detection

Signal decomposition techniques have been applied in various domains, such as wind power forecasting [23] and bearing fault diagnosis in mechanical systems [24].However, their use in cybersecurity, particularly for modeling user activity sequences from audit logs, remains underexplored.

Recent efforts have introduced wavelet-based techniques to capture behavioral anomalies. Feng *et al.* [25] combined graph-based outlier scoring with Discrete Wavelet Transform (DWT) to detect temporal deviations in user behavior on cloud-sharing platforms. Randive and Ramasundaram [26] proposed MW-CapsNet, which integrates multi-level 2D wavelet decomposition with capsule networks for image-based behavior modeling, achieving high precision on the CERT dataset. Kim *et al.* [27] applied DWT to denoise behavioral features and leveraged fuzzy clustering with OCSVM to reduce false positives. While these approaches benefit from wavelet analysis, they often rely on fixed basis functions and univariate representations.

In contrast, our work employs MVMD [10] to decompose behavior frequency signals in a data-adaptive manner, enabling fine-grained and modality-preserving anomaly detection in multichannel activity streams.

III. Preliminaries

This section briefly reviews two background techniques used in our framework: MVMD for multiscale signal decomposition and Mamba for sequence encoding. We introduce their core principles to support the design of Log2sig.

A. Multivariate Variational Mode Decomposition

MVMD [10] extends the classical Variational Mode Decomposition (VMD) framework to multichannel settings, enabling joint frequency decomposition across multiple behavioral categories.

Let $\mathbf{y}(t) = [y_1(t), \ldots, y_C(t)]^\top \in \mathbb{R}^{C\times 1}$ denote the C-channel input signal at time t, where each $y_c(t)$ corresponds to the observed activity frequency in the c-th behavior category. We adopt a column-vector convention, in which each multivariate observation is stacked channel-wise.

The goal of MVMD is to decompose $\mathbf{y}(t)$ into K multivariate intrinsic mode functions (IMFs) $\{\mathbf{u}_k(t)\}_{k=1}^K$, each capturing a narrowband component with shared spectral structure across all channels. Formally, we have:

$$\mathbf{y}(t) = \sum_{k=1}^{K} \mathbf{u}_k(t), \quad \mathbf{u}_k(t) \in \mathbb{R}^{C\times 1}. \tag{1}$$

The decomposition is obtained by solving the following variational optimization problem:

$$\begin{aligned} \min_{\{u_{k,c}\},\{\omega_k\}} \quad & \alpha \sum_{k=1}^{K} \sum_{c=1}^{C} \left\| \partial_t \left[u_{k,c}(t) e^{-j\omega_k t} \right] \right\|_2^2 \\ \text{s.t.} \quad & y_c(t) = \sum_{k=1}^{K} u_{k,c}(t), \end{aligned} \tag{2}$$

where $u_{k,c}(t)$ is the c-th channel of the k-th mode, ω_k is the center frequency of the k-th component, and $\alpha > 0$ is a regularization parameter controlling spectral compactness.

By enforcing spectral alignment across channels, MVMD effectively decomposes multivariate time series into interpretable frequency components. This facilitates robust behavior modeling by capturing both periodic structures and frequency anomalies. The number of modes K and the bandwidth control factor α act as key hyperparameters to adjust decomposition granularity.

To illustrate this decomposition process, Fig. 1 provides an example on two behavior types: HTTP visits and Email contacts. The extracted IMFs highlight distinct oscillation patterns, separating high-frequency spikes from slower, trend-like variations. This multiscale view facilitates downstream detection of both local anomalies and global drifts.

B. Mamba Encoder

Mamba [11] is a structured state space model (SSM) designed for efficient long-range sequence modeling with linear time complexity. Unlike attention-based models, Mamba leverages selective state dynamics and content-aware gating to model temporal dependencies effectively.

Let $\mathbf{X} = [\mathbf{x}_1; \ldots; \mathbf{x}_L] \in \mathbb{R}^{L\times d}$ denote the input sequence of L behavior tokens, where each $\mathbf{x}_t \in \mathbb{R}^{1\times d}$ is a row vector representing the t-th token in d-dimensional embedding space.

Mamba computes context-aware representations $\mathbf{Z} = [\mathbf{z}_1; \ldots; \mathbf{z}_L] \in \mathbb{R}^{L\times d}$ via a selective state-space recurrence defined as:

$$\mathbf{h}_t = \mathbf{h}_{t-1}\mathbf{A} + \hat{\mathbf{x}}_t\mathbf{B}, \quad \mathbf{z}_t = \mathbf{h}_t\mathbf{C}, \tag{3}$$

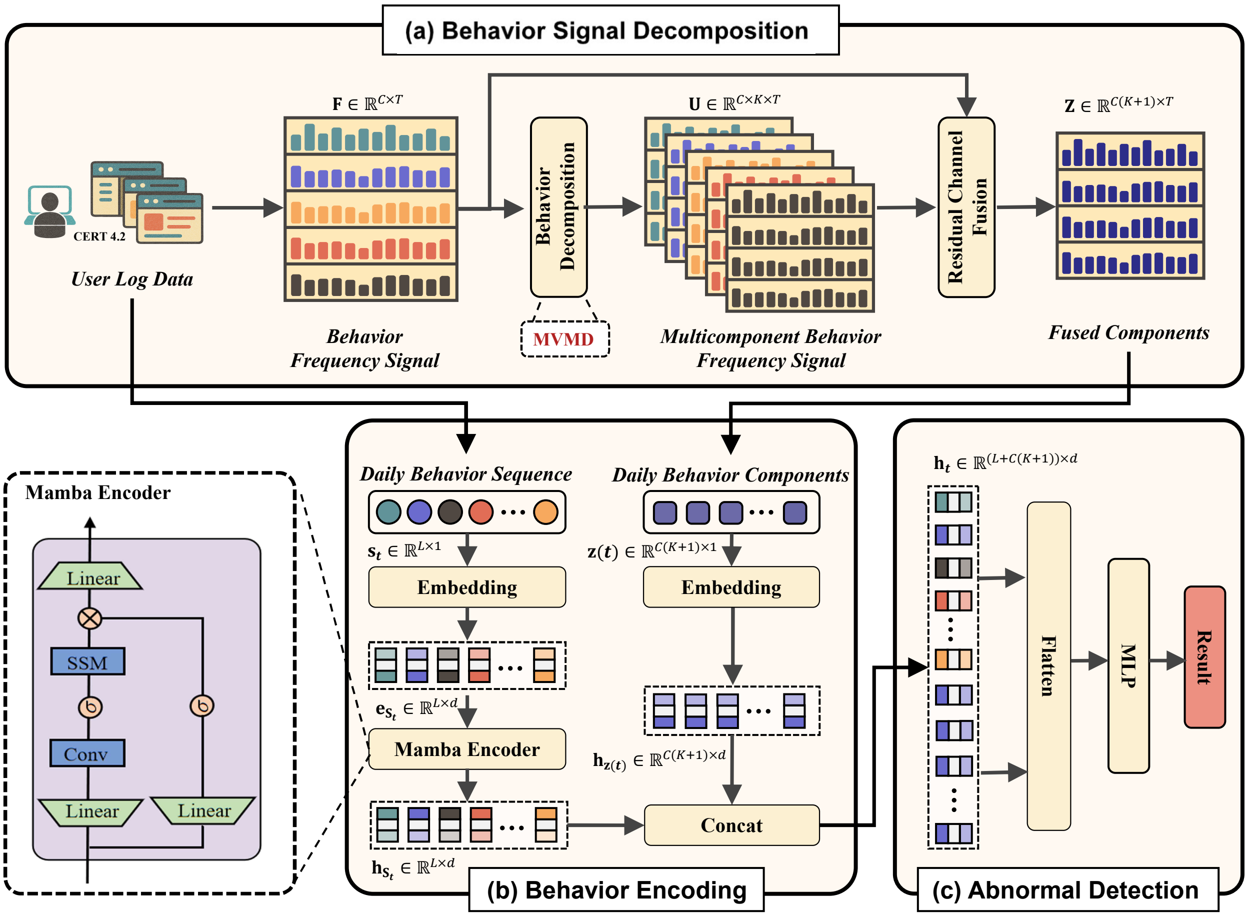

(a) Behavior Signal Decomposition
CERT 4.2
User Log Data
$\mathbf{F} \in \mathbb{R}^{C\times T}$
Behavior Frequency Signal
Behavior Decomposition
MVMD
$\mathbf{U} \in \mathbb{R}^{C\times K\times T}$
Multicomponent Behavior Frequency Signal
Residual Channel Fusion
$\mathbf{Z} \in \mathbb{R}^{C(K+1)\times T}$
Fused Components
Mamba Encoder
Linear
SSM
Conv
Linear
Linear
Daily Behavior Sequence
Daily Behavior Components
$\mathbf{s}_t \in \mathbb{R}^{L\times 1}$
$\mathbf{z}(t) \in \mathbb{R}^{C(K+1)\times 1}$
Embedding
Embedding
$\mathbf{e}_{\mathbf{s}_t} \in \mathbb{R}^{L\times d}$
Mamba Encoder
$\mathbf{h}_{\mathbf{z}(t)} \in \mathbb{R}^{C(K+1)\times d}$
Concat
$\mathbf{h}_{\mathbf{s}_t} \in \mathbb{R}^{L\times d}$
(b) Behavior Encoding
$\mathbf{h}_t \in \mathbb{R}^{(L+C(K+1))\times d}$
Flatten
MLP
Result
(c) Abnormal Detection

Fig. 2. Overview of the Log2Sig framework. (a) User activity logs are transformed into a C-channel behavior frequency signal over T days and decomposed by Multivariate Variational Mode Decomposition (MVMD) into multi-scale components. These are fused with the original signal via residual concatenation. (b) In parallel, daily behavior sequences and frequency-based components are embedded and encoded, with temporal patterns captured by a Mamba-based encoder. (c) The combined features are passed through a multi-layer classifier for anomaly detection.

where $\mathbf{h}_t \in \mathbb{R}^{1\times d}$ is the hidden state at time step t, and $\mathbf{A}, \mathbf{B}, \mathbf{C} \in \mathbb{R}^{d\times d}$ are learnable parameter matrices.

To enable content-dependent modulation, each input token is transformed through a dynamic gating mechanism:

$$\hat{\mathbf{x}}_t = (\mathbf{x}_t\mathbf{W}_u) \odot \sigma(\mathbf{x}_t\mathbf{W}_v), \tag{4}$$

where $\mathbf{W}_u, \mathbf{W}_v \in \mathbb{R}^{d\times d}$ are learnable projection matrices, $\sigma(\cdot)$ denotes the sigmoid activation function, and $\odot$ denotes element-wise multiplication. The output $\mathbf{Z} \in \mathbb{R}^{L\times d}$ maintains the original sequence length and embedding dimension.

IV. The Log2Sig Framework

This section introduces the architecture and workflow of the Log2sig framework, which consists of three main components: (1) Behavior Signal Decomposition, (2) Behavior Encoding, and (3) Anomaly Detection. Additionally, a preliminary module Behavior Representation Construction is included to standardize raw activity logs into structured behavior inputs. An overview of the full framework is depicted in Fig. 2.

A. Behavior Representation Construction

User activity logs originate from heterogeneous sources such as authentication servers, web proxies, and file access systems, each exhibiting distinct structural formats and semantic conventions. To enable consistent downstream modeling, we employ rule-based mapping strategies [28] to transform raw logs into a unified schema. Each event is assigned a high-level behavior type (e.g., login, file access), and key attributes—such as timestamps, user identifiers, and action categories—are extracted accordingly.

To preserve both the fine-grained action semantics and the aggregated behavioral statistics, we construct two complementary representations for each user on each day:

a) Behavior Sequence: Let $s_t = [b_1^{(t)}, b_2^{(t)}, \ldots, b_L^{(t)}]$ denote the ordered sequence of user u's actions on day t, where each $b_i^{(t)} \in \mathcal{B}$ is a token from the behavior vocabulary $\mathcal{B}$. This vocabulary, defined through rule-based aggregation, consolidates heterogeneous events into a compact semantic space. The sequence s_t retains both the temporal and contextual structure of user activities and is subsequently encoded using a Mamba-based sequential model.

b) Behavior Frequency single: In parallel, we compute a daily frequency vector $\mathbf{f}(t) = [f_1(t), f_2(t), \ldots, f_C(t)]^\top \in \mathbb{R}^{C\times 1}$, where $f_c(t)$ denotes the count of behavior type c observed on day t. Over a span of T days, this forms a multichannel time series:

$$\mathbf{F} = [\mathbf{f}(1), \ldots, \mathbf{f}(T)] \in \mathbb{R}^{C\times T}, \tag{5}$$

where each row traces the temporal evolution of a single behavior category. This structured signal is then passed to the signal decomposition module to extract latent frequency characteristics.

Together, the representation $(s_t, \mathbf{f}(t))$ captures both symbolic action dependencies and quantitative trends, enabling more robust and multiscale behavior modeling.

B. Behavior Signal Decomposition

While the frequency vector $\mathbf{f}(t)$ provides a compact summary of daily behavior, it lacks the capacity to capture underlying temporal rhythms and frequency-specific patterns. To enrich this representation, we apply MVMD to uncover band-limited components for each behavior type.

a) Multiscale Behavior Decomposition: As previously defined, the multichannel behavior frequency signal $\mathbf{F} \in \mathbb{R}^{C\times T}$ consists of daily frequency vectors for C behavior types over T days. We apply MVMD along the time axis to extract K intrinsic mode functions (IMFs) per behavior channel. The result is a three-dimensional tensor:

$$\mathbf{U} = \mathcal{D}_{\text{MVMD}}(\mathbf{F}) \in \mathbb{R}^{C\times K\times T}, \tag{6}$$

where $\mathbf{U}_{c,k,t}$ denotes the contribution of the k-th frequency component for behavior type c at time t.

At each time step t, the decomposed frequency components are grouped as:

$$\left\{\mathbf{m}_k(t) = \mathbf{U}_{:,k,t} \in \mathbb{R}^{C\times 1}\right\}_{k=1}^{K}, \tag{7}$$

where each $\mathbf{m}_k(t)$ represents a multichannel behavior vector oscillating at frequency level k.

b) Residual Channel Fusion: To construct a frequency-enriched representation at each time t, we concatenate the original signal $\mathbf{f}(t)$ with all K decomposed components:

$$\mathbf{z}(t) = \text{Concat}[\mathbf{f}(t), \mathbf{m}_1(t), \ldots, \mathbf{m}_K(t)] \in \mathbb{R}^{C(K+1)\times 1}. \tag{8}$$

By stacking these residual-enhanced vectors over the entire time window, we obtain:

$$\mathbf{Z} = [\mathbf{z}(1), \ldots, \mathbf{z}(T)] \in \mathbb{R}^{C(K+1)\times T}, \tag{9}$$

which retains the channel-major layout while embedding rich multiscale frequency information. This representation is then fed into the subsequent encoder for anomaly detection.

C. Behavior Encoding

To jointly model symbolic behavior sequences and frequency-based statistical patterns, we introduce a dual-view encoding strategy. Each view independently processes one modality, and the resulting embeddings are concatenated to form a comprehensive daily representation. We denote t as the index of the current day in the behavior timeline.

a) Behavior Sequence Encoding: The discrete action sequence $s_t = \{b_1^{(t)}, \ldots, b_L^{(t)}\}$ is first mapped into a d-dimensional embedding space:

$$\mathbf{e}_{s_t} = \text{Embedding}(s_t) \in \mathbb{R}^{L\times d}, \tag{10}$$

where each token $b_i^{(t)}$ corresponds to a user action occurring on day t. The sequence is then passed through a Mamba encoder [11] to capture fine-grained temporal dynamics:

$$\mathbf{h}_{s_t} = \text{Mamba}(\mathbf{e}_{s_t}) \in \mathbb{R}^{L\times d}. \tag{11}$$

Each vector in $\mathbf{h}_{s_t}$ represents a contextualized embedding of the corresponding behavior token.

b) Behavior Components Encoding: The frequency-enriched vector $\mathbf{z}(t) \in \mathbb{R}^{C(K+1)\times 1}$, constructed from the original and MVMD-decomposed behavior signals, is treated as a set of pseudo-tokens. Each scalar is projected into the embedding space as:

$$\mathbf{h}_{\mathbf{z}(t)} = \text{Linear}(\mathbf{z}(t)) \in \mathbb{R}^{C(K+1)\times d}, \tag{12}$$

where each row reflects one behavior-frequency component at day t.

c) Representation Fusion: To construct the final daily representation, we concatenate the outputs from both encoding branches along the sequence dimension:

$$\mathbf{h}_t = \text{Concat}(\mathbf{h}_{s_t}, \mathbf{h}_{\mathbf{z}(t)}) \in \mathbb{R}^{(L+C(K+1))\times d}. \tag{13}$$

This unified representation integrates symbolic behavior dynamics and frequency-aware statistical patterns, providing a rich embedding for downstream anomaly detection.

D. Anomaly detection

Based on the fused representation $\mathbf{h}_t$, we employ a lightweight classification module to detect behavioral anomalies on a daily basis.

a) Representation Flatten: To enable standard classification, we flatten the representation into a single vector:

$$a_t = \text{Flatten}(\mathbf{h}_t) \in \mathbb{R}^{(L+C(K+1))d\times 1}. \tag{14}$$

This transformation preserves both the sequential structure and the multiscale statistics in a high-dimensional feature space. The flattened vector is then passed to a multi-layer perceptron (MLP) classifier:

$$\hat{y}_t = \text{MLP}(a_t), \quad \hat{y}_t \in (0, 1), \tag{15}$$

where $\hat{y}_t$ indicates the predicted likelihood of anomalous behavior on day t.

TABLE I
STATISTICS OF THE CERT INSIDER THREAT DATASETS

Property	CERT r4.2	CERT r5.2
Time Range	Jan 2010 – May 2011	Jan 2010 – Jun 2011
Number of Users	1,000	2,000
Anomalous Users	70	99
Total Events	32,770,222	79,856,699
Anomalous Events	7,323	10,328
Anomaly Ratio (%)	0.022%	0.013%

TABLE II
HYPERPARAMETER CONFIGURATION OF LOG2SIG

Component	Configuration
MVMD Decomposition	Bandwidth α: `500` Initialization: `0` Number of Modes K: `3` Tolerance: `1e-3`
Mamba Encoder	Number of Layers: `2` Embedding Dimension: `64` Normalization: `RMSNorm`
MLP Classifier	Number of Layers: `3` Hidden Units: `256-128-32` Activation: `LeakyReLU` Dropout: `0.3` Optimizer: `Adam` Learning Rate: `5e-4` Epochs: `200` Batch Size: `32`

b) Training Objective: Given a labeled training set $\{(a_t, y_t)\}_{t=1}^{T}$, where $y_t \in \{0, 1\}$ denotes the ground-truth anomaly label, we optimize the binary cross-entropy loss:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{T}\sum_{t=1}^{T}\left[y_t \log \hat{y}_t + (1 - y_t)\log(1 - \hat{y}_t)\right]. \tag{16}$$

All parameters in the encoder and classifier are trained end-to-end using the Adam optimizer with appropriate learning rate scheduling.

V. EXPERIMENTAL CONFIGURATION

This section describes the datasets, implementation settings, baseline models, and evaluation metrics used in the experiments.

A. Datasets

We evaluate our method on the publicly available CERT Insider Threat Datasets [29], a widely used benchmark for insider threat detection. The r4.2 and r5.2 datasets contain detailed time-stamped logs from 1,000 and 2,000 users respectively, spanning activities such as logon, file access, email, and web usage. Each record is annotated with user IDs and threat labels covering scenarios like data theft and privilege misuse. Dataset statistics are summarized in Table I.

B. Implementation Settings

Experiments are conducted on the CERT Insider Threat datasets (r4.2 and r5.2, Scenario 2), focusing on 30 users with verified anomalous behaviors. Daily logs are segmented into behavior sessions comprising behavior sequences and multivariate frequency signals, which are jointly fed into the proposed model. An 80/20 train-test split is employed for evaluation. To mitigate class imbalance, Synthetic Minority Over-sampling Technique (SMOTE) with a sampling ratio of 0.5 and adaptive neighbor selection is applied, followed by Tomek Links to eliminate borderline instances.

Log2Sig employs a dual-path architecture consisting of a Mamba-based sequence encoder and an MVMD-based frequency decomposition branch. The fused representation is passed through a multi-layer perceptron (MLP) classifier. All modules are trained in an end-to-end manner. Hyperparameter settings are detailed in Table II, with tuning procedures discussed in Section VI-C.

C. Baselines Methods

Log2Sig is evaluated against a diverse set of baseline methods, categorized into three major groups: **(1) Traditional models**, including IForest [12], OCSVM [12], and XGBoost [13], which serve as representative unsupervised and supervised learning approaches, respectively; **(2) Deep learning models**, such as ITDBERT [7] and CATE [16], which employ Transformer-based or graph-enhanced architectures to capture semantic and structural properties in user logs; **(3) Large Language Model (LLM)-based methods**, including LogGPT [20] and ITDLM [22], which leverage prompt-driven inference with pretrained LLMs to perform log anomaly detection under zero-shot or few-shot settings.

D. Evaluation Metrics

Detection performance is assessed using four standard evaluation metrics derived from the confusion matrix: Recall, Precision, Accuracy, and F1-score, defined as follows:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{18}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{20}$$

Here, TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively. **Recall** quantifies the model's ability to detect actual threats, while **Precision** measures the proportion of true threats among all positive predictions. **Accuracy** reflects the overall correctness of predictions. **F1-score** combines Precision and Recall into a single measure, providing a balanced evaluation.

TABLE III
BASELINE PERFORMANCE COMPARISON

Method	CERT r4.2				CERT r5.2			
	Rec	Prec	Acc	F1	Rec	Prec	Acc	F1
IForest	0.818	0.905	0.964	0.846	0.789	0.943	0.966	0.843
XGBoost	0.827	0.957	0.973	0.871	0.854	0.973	0.978	0.899
OCSVM	0.928	0.507	0.861	0.639	0.912	0.557	0.887	0.677
ITDBERT	0.884	0.912	0.960	0.898	0.889	0.914	0.961	0.901
CATE	0.904	0.936	0.980	0.911	0.893	0.972	0.983	0.926
LogGPT	0.920	0.880	0.959	0.899	0.925	0.891	0.963	0.907
ITDLM	0.852	0.906	0.950	0.879	**0.930**	0.843	0.951	0.884
Log2Sig (Ours)	**0.929**	**0.990**	**0.990**	**0.956**	0.918	**0.986**	**0.988**	**0.946**

TABLE IV
IMPACT OF DECOMPOSITION AND SEQUENCE ENCODING

Variant	Rec	Prec	Acc	F1
w/o MVMD & Mamba	0.822	0.915	0.967	0.856
w/o MVMD	0.833	0.979	0.977	0.890
w/o Mamba	0.916	0.963	0.984	0.929
Full Model	**0.929**	**0.990**	**0.990**	**0.956**

VI. RESULTS AND DISCUSSIONS

This section reports the experimental results of Log2Sig. We first compare its overall performance against baseline models, then conduct ablation studies to assess key component contributions. We further analyze core hyperparameter sensitivity and evaluate the efficiency of different decomposition and encoding strategies.

A. Baseline Comparison

As shown in Table III, Log2Sig achieves the highest overall performance across both CERT r4.2 and r5.2 datasets. On r4.2, it outperforms all baselines in every metric, achieving an F1-score of 0.956. On r5.2, it maintains strong results with a leading F1-score of 0.946 and slightly lower recall.

Traditional models like IForest and XGBoost perform reasonably but lack adaptability to evolving behavioral dynamics. OCSVM exhibits high recall yet suffers from very low precision, indicating excessive sensitivity to benign anomalies. This stems from its static feature assumptions, which fail to model temporal or structured user behavior. Deep learning models such as ITDBERT and CATE offer more balanced performance, with CATE benefiting from enhanced structural modeling. However, both rely on static training paradigms that may limit generalization. LLM-based models (e.g., LogGPT and ITDLM) exhibit improved adaptability to unseen logs. While LogGPT offers balanced precision-recall, its effectiveness diminishes under dynamic conditions.

In contrast, Log2Sig integrates multiscale frequency decomposition with sequence modeling, enabling robust, high-precision detection of subtle threats. These results demonstrate its superiority in both static and dynamic insider threat scenarios.

B. Ablation Study

As shown in Table IV, we conduct an ablation study on the CERT r4.2 dataset to evaluate the individual contributions of each core component in the Log2Sig framework. The full model integrates behavior encoding via the Mamba sequence encoder and multiscale decomposition through MVMD. This configuration achieves the best overall performance, with an F1-score of 0.956 across all evaluation metrics.

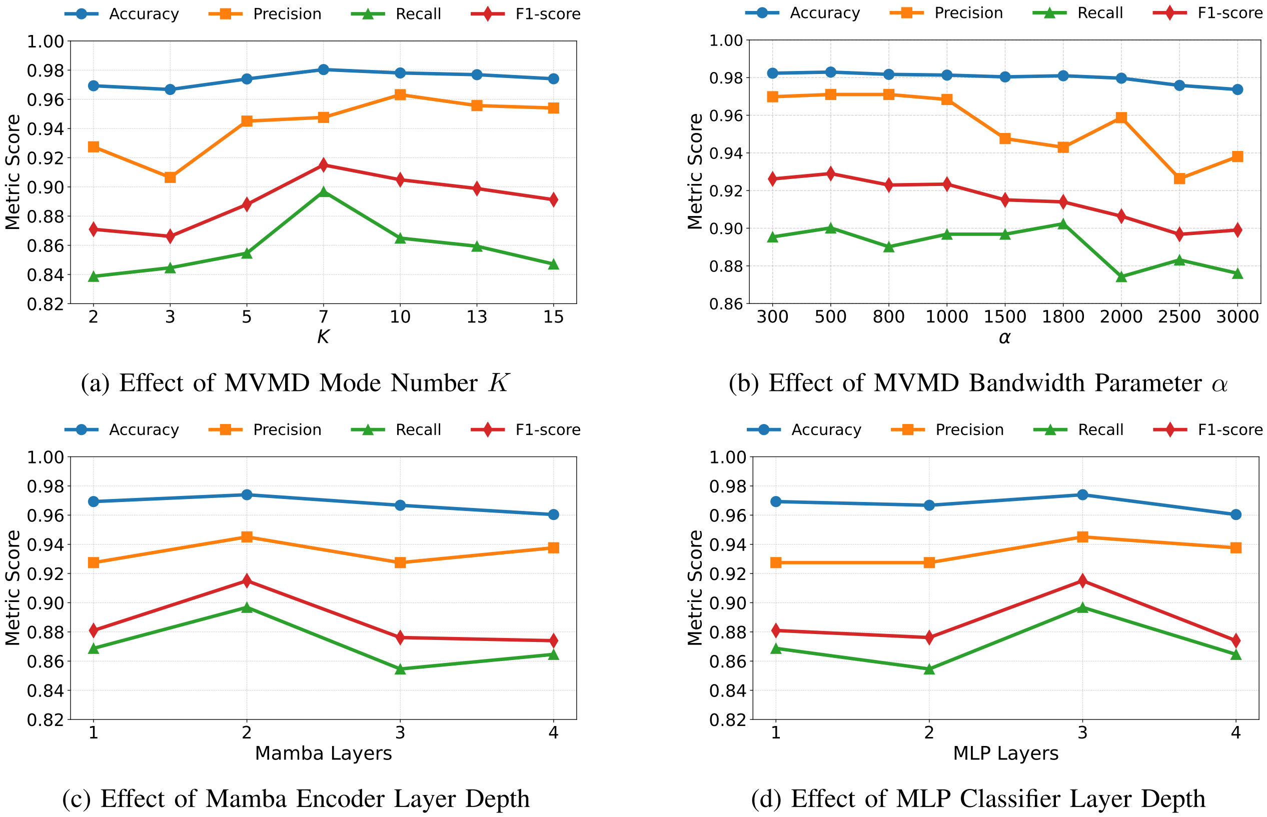
To assess component-wise impact, we test several reduced variants. Removing the Mamba encoder (w/o Mamba) while retaining MVMD leads to a moderate decline in performance (F1 = 0.929), indicating the importance of intra-day sequential modeling. Excluding the frequency decomposition module (w/o MVMD) results in a larger decrease in recall and F1-score, highlighting the role of multiscale signal modeling. When both components are omitted (w/o MVMD and Mamba), the model exhibits the weakest performance (F1 = 0.856), confirming that both components are essential for robust anomaly detection.

C. Parameter Sensitivity Analysis

a) MVMD Mode Number K: As shown in Fig. 3(a), increasing the number of decomposition modes generally improves accuracy and precision, with performance peaking at $K = 7$. This indicates that moderate multiscale resolution enhances the model's ability to capture meaningful frequency components. However, when K exceeds 10 (e.g., $K = 13$ or 15), performance degrades, particularly in recall and F1-score, likely due to the introduction of redundant or noisy modes. Based on this observation, $K = 7$ is selected as the optimal setting.

b) MVMD Bandwidth α: Fig. 3(b) presents the sensitivity to the bandwidth parameter α. The model remains relatively stable in the range of 300 to 1000, but larger values (e.g., $\alpha \geq 2000$) cause noticeable drops in recall. This degradation may stem from excessive smoothing, which reduces decomposition fidelity. A value of $\alpha = 500$ is therefore adopted as it offers a robust trade-off between precision and generalization.

c) Mamba Encoder Layers: According to Fig. 3(c), using 2 layers in the Mamba encoder achieves the most balanced performance. This depth is sufficient to model temporal dependencies while avoiding overfitting. Deeper configurations



(a) Effect of MVMD Mode Number K

(b) Effect of MVMD Bandwidth Parameter α

(c) Effect of Mamba Encoder Layer Depth

(d) Effect of MLP Classifier Layer Depth

Fig. 3. Impact of key hyperparameters on model performance: MVMD decomposition settings (K, α), Mamba encoder depth, and MLP classifier depth.

TABLE V
COMPARISON OF MULTICHANNEL DECOMPOSITION METHODS (PER USER AVERAGE).

Method	Acc	F1	Memory (MB)	Time (s)
MEMD	0.965	0.911	**1.17**	6.37
MVMD	**0.984**	**0.928**	9.89	**0.49**

TABLE VI
COMPARISON OF SEQUENCE ENCODER METHODS (PER USER AVERAGE).

Method	Acc	F1	GPU(MB)	Time (s)
LSTM	0.947	0.897	11.24	6.67
Transformer	0.963	0.912	8.89	**6.11**
Mamba	**0.981**	**0.931**	**6.33**	**4.67**

(3 or 4 layers) lead to performance declines, suggesting that additional layers may introduce overparameterization or vanishing gradients. Thus, a 2-layer encoder is used in our final setup.

d) MLP Classifier Layers: As depicted in Fig. 3(d), increasing the depth of the MLP classifier enhances performance up to 3 layers, particularly in terms of recall and F1-score. However, further deepening to 4 layers results in performance drops, likely due to training instability or overfitting in the final classification stage. A 3-layer MLP is therefore adopted for the final architecture.

D. Comparison of Decomposition and Encoder Strategies

Table V and Table VI summarize the performance and efficiency of different multichannel decomposition and sequence encoding strategies under a unified classification pipeline. All methods are evaluated with consistent preprocessing and training configurations to ensure fair comparison.

For decomposition, MVMD [10] outperforms MEMD [30] with higher accuracy (0.984 vs. 0.965), F1-score (0.928 vs. 0.911), and dramatically lower computation time (0.49s vs. 6.37s), at the cost of slightly increased memory usage (9.89 MB vs. 1.17 MB). Note that both memory and time measurements refer solely to the decomposition stage, excluding downstream processing.

In terms of sequence encoding, Mamba achieves the best results across all metrics, with the highest accuracy (0.981) and F1-score (0.931), while also being the most efficient—consuming the least GPU memory (6.33 MB) and achieving the fastest inference speed (4.67s per user). These measurements are isolated to the encoder stage during per-user sequence modeling, excluding input preprocessing or classification layers.

In summary, the experimental results consistently demonstrate that MVMD and Mamba are the most effective and efficient components within their respective modules. MVMD significantly accelerates the multichannel decomposition process while improving classification performance, making it well-suited for real-time applications. Similarly, the Mamba encoder not only surpasses LSTM and Transformer in predictive accuracy but also offers superior computational efficiency with reduced GPU memory usage and inference time.

VII. Conclusion

In this work, we proposed Log2Sig, a novel frequency-aware framework for insider threat detection that combines multivariate signal decomposition with deep sequence modeling. Unlike traditional methods that treat user logs as flat event sequences, Log2Sig transforms multichannel behavioral data into temporal signals and applies MVMD to extract frequency-localized intrinsic mode functions. These decomposed components, when fused with daily behavior statistics, reveal subtle and multiscale variations often overlooked by conventional models. We further design a hybrid encoding architecture, where the daily behavior sequence is processed by a lightweight Mamba-based temporal encoder to capture long-range dependencies, while frequency-derived components are embedded and directly fused at the feature level. This enables efficient integration of semantic and spectral behavior cues for accurate anomaly detection. Experiments on the CERT r4.2 and r5.2 datasets show that Log2Sig achieves consistently strong performance across different versions of the dataset.

References

[1] F. R. Alzaabi and A. Mehmood, "A review of recent advances, challenges, and opportunities in malicious insider threat detection using machine learning methods," *IEEE Access*, vol. 12, pp. 30907–30927, 2024.

[2] Ponemon Institute, "2025 Global cost of insider risk report," 2025. [Online]. Available: https://www.dtexsystems.com/blog/2025-cost-insider-risks-takeaways/.

[3] F. Liu, Y. Wen, D. Zhang, X. Jiang, X. Xing, and D. Meng, "Log2vec: A heterogeneous graph embedding based approach for detecting cyber threats within enterprise," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1777–1794, 2019.

[4] D. C. Le, N. Zincir-Heywood, and M. I. Heywood, "Analyzing data granularity levels for insider threat detection using machine learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 30–44, 2020.

[5] B. Bin Sarhan and N. Altwaijry, "Insider threat detection using machine learning approach," *Applied Sciences*, vol. 13, no. 1, p. 259, 2022.

[6] W. He, X. Wu, J. Wu, X. Xie, L. Qiu, and L. Sun, "Insider threat detection based on user historical behavior and attention mechanism," in *Proc. IEEE 6th Int. Conf. Data Sci. Cyberspace (DSC)*, 2021, pp. 564–569.

[7] W. Huang, H. Zhu, C. Li, Q. Lv, Y. Wang, and H. Yang, "ITDBERT: Temporal-semantic representation for insider threat detection," in *Proc. IEEE Symp. Computers and Communications (ISCC)*, Athens, Greece, 2021, pp. 1–7.

[8] K. C. Roy and G. Chen, "GraphCH: A deep framework for assessing cyber-human aspects in insider threat detection," *IEEE Trans. Dependable Secure Comput.*, early access, 2024.

[9] J. Xiao, L. Yang, F. Zhong, X. Wang, H. Chen, and D. Li, "Robust anomaly-based insider threat detection using graph neural network," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3717–3733, 2022.

[10] N. Ur Rehman and H. Aftab, "Multivariate variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 6039–6052, 2019.

[11] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[12] F. W. Bartoszewski, M. Just, M. A. Lones, and O. Mandrychenko, "Anomaly detection for insider threats: An objective comparison of machine learning models and ensembles," in *Proceedings of the IFIP International Conference on ICT Systems Security and Privacy Protection*, pp. 367–381, Springer, 2021.

[13] D. C. Le and N. Zincir-Heywood, "Anomaly detection for insider threats using unsupervised ensembles," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1152–1164, 2021.

[14] R. Yousef, M. Jazzar, A. Eleyan, and T. Bejaoui, "A machine learning framework & development for insider cyber-crime threats detection," in *Proceedings of the 2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*, pp. 1–6, IEEE, 2023.

[15] P. Pal, P. Chattopadhyay, and M. Swarnkar, "Temporal feature aggregation with attention for insider threat detection from activity logs," *Expert Syst. Appl.*, vol. 224, p. 119925, 2023.

[16] H. Xiao, Y. Zhu, B. Zhang, Z. Lu, D. Du, and Y. Liu, "Unveiling shadows: A comprehensive framework for insider threat detection based on statistical and sequential analysis," *Computers & Security*, vol. 138, p. 103665, 2024.

[17] A. Budžys, O. Kurasova, and V. Medvedev, "Deep learning-based authentication for insider threat detection in critical infrastructure," *Artificial Intelligence Review*, vol. 57, no. 10, p. 272, 2024.

[18] R. G. Gayathri, A. Sajjanhar, and Y. Xiang, "Hybrid deep learning model using SPCAGAN augmentation for insider threat analysis," *Expert Systems with Applications*, vol. 249, p. 123533, 2024.

[19] X. Cai, Y. Wang, S. Xu, H. Li, Y. Zhang, Z. Liu, and X. Yuan, "LAN: Learning adaptive neighbors for real-time insider threat detection," *IEEE Transactions on Information Forensics and Security*, 2024.

[20] J. Qi, S. Huang, Z. Luan, S. Yang, C. Fung, H. Yang, D. Qian, J. Shang, Z. Xiao, and Z. Wu, "LogGPT: Exploring ChatGPT for log-based anomaly detection," in *Proc. IEEE Int. Conf. High Performance Computing & Communications, Data Science & Systems, Smart City & Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, 2023, pp. 273–280.

[21] Y. Liu, S. Tao, W. Meng, J. Wang, W. Ma, Y. Chen, Y. Zhao, H. Yang, and Y. Jiang, "Interpretable online log analysis using large language models with prompt strategies," in *Proc. 32nd IEEE/ACM Int. Conf. Program Comprehension (ICPC)*, 2024, pp. 35–46.

[22] S. Song, Y. Zhang, and N. Gao, "Confront insider threat: Precise anomaly detection in behavior logs based on LLM fine-tuning," in *Proc. 31st Int. Conf. Computational Linguistics (COLING)*, 2025, pp. 8589–8601.

[23] T. Yang, Z. Yang, F. Li, and H. Wang, "A short-term wind power forecasting method based on multivariate signal decomposition and variable selection," *Applied Energy*, vol. 360, p. 122759, 2024.

[24] Q. Song, X. Jiang, G. Du, J. Liu, and Z. Zhu, "Smart multichannel mode extraction for enhanced bearing fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 189, p. 110107, 2023.

[25] W. Feng, W. Yan, S. Wu, and N. Liu, "Wavelet transform and unsupervised machine learning to detect insider threat on cloud file-sharing," in *Proc. IEEE Int. Conf. Intelligence and Security Informatics (ISI)*, Beijing, China, Jul. 2017, pp. 155–157.

[26] K. D. Randive and M. Ramasundaram, "MWCapsNet: A novel multi-level wavelet capsule network for insider threat detection using image representations," *Neurocomputing*, vol. 553, p. 126588, 2023.

[27] D.-W. Kim, G.-Y. Shin, and M.-M. Han, "Anomaly detection based on discrete wavelet transformation for insider threat classification," *Computer Systems Science and Engineering*, vol. 46, no. 1, pp. 153–164, 2023.

[28] K. Kong, X. Jin, D. Liu, S. Xu, Z. Liu, and G. Geng, "DPI-ITD: A dual-perspective information-driven framework for insider threat detection in IoT systems," *IEEE Internet of Things Journal*, 2025.

[29] B. Lindauer, "Insider threat test dataset," *doi:10.1184/R1/12841247.v1*, 2020. [Online]. Available: https://doi.org/10.1184/R1/12841247.v1.

[30] N. Rehman and D. P. Mandic, "Multivariate empirical mode decomposition," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 466, no. 2117, pp. 1291–1302, 2010.