Politecnico di Torino

Master's Degree in Data science and Engineering

Master's Degree Thesis

Multimodal Interpretation of Time-Series Discords: A Hybrid Approach using Matrix Profile and VLMs

Supervisors

Prof. LUCA CAGLIERO

Ph.D. ALI YASSINE

Candidate

MEHRBOD NOWROUZ

MARCH 2026

Summary

Time-series anomaly detection has traditionally been dominated by statistical methods and narrow Deep Learning architectures. Most well-known algorithms require extensive feature engineering and struggle with complex signals; furthermore, while certain neural models show better performance in pattern recognition, they remain "black boxes" that lack explainability and the ability to incorporate domain context. Often, these systems fall victim to an "illusion of progress" where high numerical scores mask a lack of true semantic understanding of actual system behavior.

In contrast, the emergence of Foundation Models introduces a shift towards generalized intelligence, culminating in the development of Vision-Language Models (VLMs). Unlike previous deep learning models, VLMs are natively multimodal, enabling the transformation of time-series data into visual representations. This thesis investigates whether the reasoning capabilities of VLMs can be leveraged to provide semantic context and pinpoint anomalies—such as frequency drifts or signal distortions that older models might miss.

To ensure a valid evaluation, this research utilizes the UCR Time-Series Anomaly Archive to avoid the "fundamentally flawed" pitfalls of legacy benchmarks. Because VLMs require heavy computation, the proposed pipeline is designed to be as light as possible by employing a two-part architecture: an Algorithmic Layer, where the Matrix Profile (MP) performs large-scale discord discovery, and a Neural Layer, where the VLM interprets the structure of discovered discords. This combination preserves algorithmic rigor while adding semantic interpretability.

The methodology initiates with Dominant Frequency Analysis via Fast Fourier Transform (FFT) to select an optimal window size that captures the signal motif. After discord discovery, signals are transformed into visual representations and divided into benign and suspect windows. To exploit few-shot reasoning, four windows are selected based on MP distance scores: a "Benign Baseline" representing the stable global motif, and three "Structural Variations" representing rare but non-anomalous fluctuations. These references are interleaved with textual explanations and frequency data, allowing the model to compare windows visually, semantically, and mathematically.

Evaluated using the Affiliation F1 criterion, experimental results demonstrate that combining the mathematical rigor of the Matrix Profile and FFT with the multimodal reasoning of VLMs results in a strong, affordable, and explainable solution. The resulting hybrid architecture takes advantage of the strengths of both approaches while limiting their respective weaknesses.

Acknowledgements

ACKNOWLEDGMENTS

I am profoundly grateful to every individual who has contributed to this journey. Whether through academic guidance, emotional support, or personal strength, your help has been invaluable. This work was not accomplished by me alone, but by the collective strength of everyone who stood by me.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
Artificial Intelligence

GNN
Graph Neural Network

LLM
Large Language Model

ML
Machine Learning

DL
Deep Learning

VLM
Visual Language Models

RNN
Recurrent Neural Networks

LSTM
Long-Short Term Memory

NER
Named Entity Recognition

NLU
Natural Language Understanding

BERT

Bidirectional Encoder Representations of Transformers

GPT

Generative Pre-trained Transformer

ViT

Vision Transformer

CLIP

Contrastive Language-Image Pretraining

MP

Matrix Profile

Chapter 1

Introduction

The increase in the use of high-frequency sensor data in modern industrial systems has made Time-Series Anomaly Detection (TSAD) a critical area of research. While the transition from traditional statistical methods to deep learning architectures has yielded significant advancements in pattern recognition, it has simultaneously introduced profound challenges regarding interpretability, trustworthiness, and computational overhead. To address the dichotomy between mathematical precision and semantic understanding, this thesis proposes a novel neuro-symbolic framework that bridges the deterministic rigor of algorithmic sequence matching with the zero-shot reasoning capabilities of foundational Vision-Language Models (VLMs).

1.1 Background and Motivation

In the era of Industry 4.0, the operational integrity of critical infrastructure depends fundamentally on the continuous monitoring of high-frequency sensor data. From aerospace applications to advanced manufacturing pipelines, and modern systems generate massive volumes of sequential data. Detection of critical failures and sensor degradation, or ,in general, anomalies within these large-scale time-series streams is vital to the survival of such systems that rely on constant functionality. A missed anomaly or delayed diagnosis may result in irreversible equipment damage, longer downtimes, or significant financial loss. Consequently, robust Time-Series Anomaly Detection (TSAD) has become a central area of research in the current data engineering and industrial analytics.

However, the research in this area has been undergoing growing challenges which are leading to new research areas and shifts. This scenario is better explained as follows:

1.1.1 The Limitations of Traditional and Statistical Methods

Historically, anomaly detection in sequential data has been dominated by classical statistical modeling and inflexible signal-processing techniques. Earlier approaches were mostly forecasting mathematical models such as ARIMA, seasonal decomposition methods, and statistical process control tools (e.g., Shewhart control charts). While they are computationally efficient, these techniques require extensive manual feature engineering, parameter tuning, and handcrafted thresholds which cause these models to be excessively rigid.

Despite significant human intervention, traditional methods are inflexible and struggle to adapt to non-stationary, nonlinear, and constantly changing characteristics of real industrial environments. They are often overly sensitive to benign noise and exhibit limited capability in distinguishing normal variance in the data from genuine irregularities. The more complex a system becomes, the more purely statistical techniques exhibit weakness. This inevitable characteristic of these solutions cause them to be rather impractical and lacking in scalability.

To tackle the above issues, engineers proceeded to find a more robust and dynamic solution.

1.1.2 The "Black Box" Problem in Deep Learning

To address the limitations of statistical forecasting, research shifted toward deep neural architectures, particularly Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. These models significantly improved performance in more complex pattern recognition tasks and demonstrated superior capability in modeling temporal dependencies.

However, this improvement introduced a critical operational limitation: lack of interpretability. Deep neural networks learn the pattern in the data based on their earlier guesses and the ground truth results and thus the pattern they find in the data is almost entirely hidden inside their parameters. Then, they output anomaly scores or reconstruction errors without providing human-interpretable reasoning. In an industrial context, such outputs are not sufficient. Without engineers understanding whether a detected anomaly reflects a benign calibration drift, transient environmental noise, or an impending mechanical failure, such outputs are not only unacceptable but in particular cases might be too vague and prevent the engineers from pinpointing the issue.

Furthermore, such models require extensive task-specific training, large labeled datasets, and continuous retraining under distribution shifts during runtime in case of a dynamic scenario, which is mostly the dominant scenario in a real environment. They cannot directly incorporate domain rules or constraints into their inference

mechanisms. Consequently, this separation between statistical detection and domain semantics introduces a new form of limitation which lies in their suitability for autonomous deployment in critical environments.

Recently, further research in the area of anomaly detection has further proved the above claims which not only confirms the limitations caused by lack of interpretation in real world environments but also how they have created misconceptions of success in an academic setting.

1.1.3 The "Illusion of Progress" in Anomaly Detection

The reliance on opaque neural architectures has led to what can be described as an "illusion of progress." Models are optimized to maximize benchmark metrics' precision, recall, F1-score—on datasets. Yet strong numerical performance often masks these models' inability to capture the semantics behind the data.

Many anomaly detection benchmarks fail to capture the topological and structural complexity present in real-world signals. As a result, models have achieved high quantitative scores during the research and development phase while failing miserably upon facing a distributional shift or emergence of benign noise during operation. Without interpretable reasoning, such systems remain difficult to trust and deploy in industrial settings.

1.1.4 The Paradigm Shift: Foundational Vision–Language Models

Recent advances in foundational models mark a significant paradigm shift toward generalized and multimodal intelligence. In particular, Vision–Language Models (VLMs) integrate visual perception with linguistic reasoning within a unified transformer-based architecture.

Unlike conventional neural models that are applied directly on numerical data, VLMs transform time-series data into visual representations. Furthermore, their inherent architecture allows the model to leverage large-scale pretraining on diverse visual and textual corpora. Through utilization of their visual reasoning learned during training on massive data, these models can analyze geometric and topological properties of signals—such as waveform shape, amplitude hierarchy, periodic consistency, and frequency drift.

Because foundational models are trained on countless heterogeneous data distributions, they may be flexible to detection of subtle structural deviations that traditional algorithms or narrowly trained deep models show weakness in. This thesis investigates the use of Vision Language Models not merely as anomaly detectors, but as semantic interpreters capable of providing structured, explainable

diagnostics upon deterministic algorithms utilized for efficiency but lacked semantic reasoning to support their procedure.

The motivation is to combine the mathematical strengths of deterministic mathematical methods—specifically the Matrix Profile—with the semantic reasoning capabilities of multimodal foundational models. By integrating algorithmic precision with high-level interpretability, the proposed framework aims to deliver scalable, computationally efficient, and an explainable anomaly detection pipeline that is suitable for industrial deployment and further scalable purposes.

1.2 Research Objectives

The primary objective of this research is to evaluate the performance of Vision–Language Models in time-series anomaly detection. More precisely, this thesis investigates whether the reasoning capabilities of VLMs can provide the semantic context required for explainable diagnostics in complex systems.

To achieve this objective, three core technical goals are pursued:

1. **Develop a Hybrid Framework:** Integrate deterministic mathematical algorithms with the semantic reasoning capabilities of a Vision–Language Model to construct a unified, anomaly detection pipeline.

2. **Eliminate the "Black Box" Paradigm:** Guide the VLM through strict prompt engineering so that it functions as an anomaly detector rather than a generative language model that produces rule-based diagnostic outputs in a machine-readable JSON format to be further used.

3. **Validate Against Robust Benchmarks:** Evaluate the proposed framework using the UCR Time-Series Anomaly Archive to mitigate issues such as data leakage, thereby ensuring a reliable environment for assessing both mathematical and semantic reasoning.

1.3 Motivation

Although Vision–Language Models exhibit strong reasoning capabilities, their performance on time-series anomaly detection tasks is yet under research. Furthermore, their direct application to long-duration time-series streams undermines their capabilities due to their heavy computation which is against the nature of a continuous data stream task. Passing every sliding window of a signal stream to a multi-billion-parameter foundational model is computationally and financially infeasible for real-world deployment.

The central motivation of the proposed architecture is to address this efficiency constraint while benefiting from their reasoning and explainability capabilities. A deterministic mathematical algorithm filters out clearly normal signal segments before invoking the heavy computations performed by the VLM. In this architecture, mathematical algorithms and neural models are not competitors but complementary components: the algorithm ensures scalability and efficiency, while the VLM provides semantic interpretation.

1.4 Scope

The scope of this thesis is restricted to univariate time-series data drawn from the 250 datasets contained in the UCR Archive.

Methodologically, the algorithmic discovery phase is limited to the use of the Matrix Profile (MP) and Fast Fourier Transform (FFT) for estimating baseline periodicity (m) and identifying discords. The reasoning capabilities of the Vision–Language Model are utilized for topological evaluation of the detected discord with respect to the overall topology of the signal based on directional order, amplitude hierarchy, and inter-extrema distance.

Through prompt engineering, the VLM is explicitly instructed to ignore stochastic noise, absolute scale, and visual artifacts based on the samples found in the reference windows. Its final output is constrained to a machine-readable JSON file. Real-time deployment is considered out of scope of this thesis due to current API latency constraints and the size of VLMs used.

1.5 Structure of the Thesis

The remainder of this thesis is structured as follows:

- **Chapter 2 (Related Work):** Reviews the evolution of anomaly detection researches, the limitations of existing anomaly detection baselines, and the rise of foundational Vision–Language Models and the datasets paving the way for modern evaluations.

- **Chapter 3 (Theoretical Background):** Details the mathematical concepts behind the Matrix Profile and the architecture of Transformers and the self-attention mechanisms within them.

- **Chapter 4 (Methodology):** Describes the proposed pipeline, including Phase I (Algorithmic Discovery), Phase II (Visual Transformation and 1+3 Layout Design), and Phase III (Strict Prompt Engineering).

- **Chapter 5 (Experimental Results):** Presents the empirical evaluation of the pipeline using the Affiliation F1 metric, analyzing both detection performance and computational efficiency relative to baseline methods.
- **Chapter 6 (Conclusion):** Summarizes the principal findings, discusses the theoretical recall ceiling imposed by the algorithmic layer, and introduces the directions for future researche.

Chapter 2

Related work

2.1 Traditional and Distance-Based Methods

The detection of anomalies in sequential data has always been a branch of data science's main research topics. Early research in Time-Series Anomaly Detection (TSAD) was primarily grounded in statistical modeling and geometric distance-based methodologies that are characterized by deterministic inference and mathematical approaches over learned features of such data.

2.1.1 Statistical Forecasting and Thresholding

Initial approaches to TSAD relied heavily on parametric statistical forecasting models. Techniques such as Autoregressive Integrated Moving Average (ARIMA [1]) and Exponential Smoothing were widely utilized to model temporal dependencies and predict future behavior from historical trends in data. An anomaly was defined as any observation that lay beyond a predefined confidence interval or exceeded a residual error threshold calculated from the data itself.

These methods offer two principal advantages: computational efficiency and interpretability. Model parameters have clear statistical meaning, and anomaly decisions can be traced directly to forecast residuals. However, due to the core concept of such models, they inherently assume linearity, stationarity, and relatively stable variance structures.

This contradicts modern industrial environments—where sensor data are often nonlinear, non-stationary, multi-scale, and contaminated by stochastic noise and thus the assumptions behind their methodology barely hold. Therefore, statistical forecasting approaches tend to result in high false-positive rates and require frequent manual calibration. Their rigidity limits adaptability and flexibility in dynamic operational contexts.

2.1.2 Distance-Based Metrics and Subsequence Matching

To overcome the limitations of prediction-based statistical models, recent research shifted toward similarity-based approaches. Instead of forecasting future values in the time-series, these algorithms proceed to evaluate the similarity between each segment within a time series.

Earlier methods employed Euclidean Distance to measure similarity between subsequences. While simple and efficient, Euclidean Distance is sensitive to temporal misalignment. Dynamic Time Warping (DTW) was introduced to address this issue by allowing nonlinear temporal alignment between sequences. DTW is particularly effective for signals exhibiting phase shifts or variable temporal speeds.

However, the DTW algorithm has a quadratic time complexity, $O(N^2)$, with respect to sequence length which would render it impractical for large-scale industrial usage. Although pruning and approximation strategies exist, scalability remains a central limitation of classical distance-based matching methods.

2.1.3 The Matrix Profile as the State-of-the-Art

A significant advancement in sequence analysis emerged with the introduction of the Matrix Profile by Yeh et al. (2016) [2]. The Matrix Profile provides a precise yet industrially scalable method to compute the nearest-neighbor distance for every segment within a time series.

Given a subsequence of length m, the algorithm computes its minimum distance to all other non-overlapping subsequences. The resulting output of the above operation is a vector that is named the *Matrix Profile*. This representation enables direct identification of:

- **Motifs:** Frequently occurring patterns in data that are characterized by low nearest-neighbor distance values and represent the general behavior of a certain system during operation.
- **Discords:** Rare or anomalous subsequences that show highest nearest-neighbor distance values and represent the state of the system in case of an issue or out of ordinary.

Unlike machine learning models, the Matrix Profile operates purely on geometric principles and does not require labeled training data. It does not include many parameters to operate which renders it scalable to large datasets through FFT-based optimizations. As a result, it is widely regarded as the gold standard for zero-shot, deterministic anomaly discovery in univariate time-series analysis.

Nevertheless, the Matrix Profile remains fundamentally geometric. While it efficiently identifies the location of an anomaly occurrence, it lacks the semantic

reasoning necessary to explain the reason behind structural deviations of a discord. The output is a distance magnitude, not an interpretable diagnostic narrative. This limitation motivates the integration of algorithmic precision with higher-level semantic interpretation mechanisms, as explored in subsequent sections of this thesis.

2.2 The Deep Learning Era and the "Illusion of Progress"

As industrial sensors have been evolving, they are generating even more high-volume, high-dimensional data streams. However, as the challenges became more complex with this evolution, the constraints of traditional statistical and distance-based algorithms became increasingly evident. The research community was consequently forced to shift toward newer and more dynamic techniques mainly based on Deep Learning (DL) architectures with the goal of automating feature extraction and capturing complex, nonlinear temporal dependencies directly from raw sequences.

2.2.1 The Shift to Neural Architectures

The deep learning architectures used in Time-Series Anomaly Detection (TSAD) are categorized into two separate fields of forecasting-based and reconstruction-based approaches.

Forecasting Models. Forecasting-based architectures, including Long Short-Term Memory (LSTM) [3] networks and Temporal Convolutional Networks (TCNs) [4], are trained to predict future timestamps based on historical observations in the training data. An anomaly is detected when the observed value deviates significantly from the model's prediction and such deviation is usually measured with residual magnitudes. The assumption behind such architecture is that a well-trained model can accurately forecast normal system behavior, and deviations from this learned pattern indicate irregularities.

Reconstruction Models. Reconstruction-based models, particularly Autoencoders (AEs) and Variational Autoencoders (VAEs), emerged as the dominant unsupervised strategy. These architectures compress fixed-length time-series windows into a lower-dimensional latent representation and proceed to reconstruct the original input. These models are trained exclusively on benign data, and thus the model learns a compressed representation of normal dynamics. Structural

anomalies, which lie outside the learned latent pattern, result in elevated reconstruction error—commonly quantified via Mean Squared Error (MSE)—serving as an anomaly score.

Both approaches significantly improved raw detection metrics relative to classical statistical baselines, especially in the area of research benchmark settings.

2.2.2 The Black Box Problem and Lack of Interpretability

Despite quantitative improvements in the performance indicators, deep neural networks introduced a critical operational limitation that is of high importance: interpretability. In critical domains such as power grid monitoring, aerospace telemetry, or medical diagnostics, anomaly detection alone is insufficient to result in decisive action. Decision-makers require a proper and informative explanation of why a certain situation or indicator is anomalous.

Neural architectures reduce complex data with multiple dimensions into scalar outputs in form of residual magnitudes, reconstruction errors, or probability scores. These values provide no direct insight into the geometric or topological cause of the deviation. The model cannot explicitly indicate whether the anomaly resulted from which of the following scenarios:

- A phase shift in periodic structure,
- An altered amplitude hierarchy,
- A distortion in waveform symmetry,
- Or a collapse in inter-extrema temporal spacing.

This opacity creates a substantial trust deficit. Without interpretable reasoning, such systems remain difficult to validate and audit. Therefore, autonomous systems in high-stakes industrial settings become risky and impractical if the target system is of critical decisions.

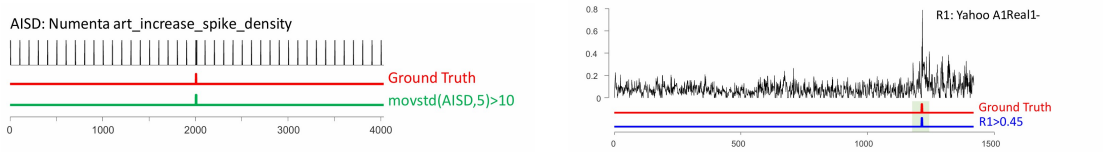
2.2.3 Flawed Benchmarks and the "Illusion of Progress"

Beyond interpretability issues, the empirical dominance of deep learning in TSAD has faced significant academic criticism. A pivotal critique by Wu and Keogh argued that the field was experiencing an "illusion of progress."

Their analysis revealed that many state-of-the-art neural architectures were evaluated on flawed legacy benchmarks, including the Yahoo and Numenta datasets. These datasets exhibited several structural deficiencies:

- **Triviality:** Many anomalies consisted of large, isolated spikes easily detectable through simple thresholding which made the usage of deep architectures unnecessary.
- **Unrealistic Anomaly Density:** Some datasets contained high anomaly frequencies. This quality distorts evaluation metrics.
- **Data Leakage:** Overlapping windows and improper splits artificially inflated F1 scores. This led to misleading conclusions regarding model superiority.

To address these limitations, the UCR Time-Series Anomaly Archive was introduced. The archive contains carefully designed datasets, each with exactly one verified, non-trivial anomaly per file. When leading neural architectures based on flawed benchmarks were re-evaluated under this stricter regime, their performance often deteriorated substantially. They even showed signs of underperformance compared to mathematical algorithms such as the Matrix Profile.



(a) Numeta **(b)** Yahoo

Figure 2.1: Examples of trivial anomalies present in the Yahoo and Numenta datasets. Some anomalies inside these datasets can be easily detected using simple "one-liner" statistical thresholds which demonstrates the flawed benchmarks from which an illusion of progress for deep learning architectures had been caused. (Image reproduced from Wu and Keogh, 2020).

This finding is foundational to the present thesis. It demonstrates that relying exclusively on neural architectures for capturing the time-series semantics is highly susceptible to overfitting and benchmark bias. The evidence motivates a methodological shift toward hybrid systems that enforce explicit structural reasoning while retaining computational scalability.

2.3 Foundational Models and Multimodal AI in Time Series

In response to the issues of narrowly specialized deep learning architectures, recent research has investigated the application of Foundational Models—particularly

Large Language Models (LLMs) and Vision Language Models (VLMs) to time-series analysis. Trained on internet-scale corpora, these models exhibit strong zero-shot generalization and cross-domain reasoning that are required for complex tasks.

2.3.1 Time-Series as Tokens

One of the earliest strategies for adapting foundational models to sequential numeric data involved treating time-series tasks as a language modeling problem. In this paradigm, continuous time-series values are discretized into tokens, which enables the direct application of transformer-based Natural Language Processing (NLP) architectures.

Models such as Chronos [5] tokenize time-series observations into discrete bins and treat the result as a language. Standard transformer architectures are then trained or prompted to predict future tokens in an autoregressive fashion. This approach enables zero-shot or few-shot forecasting without extensive domain-specific retraining.

However, discretization introduces a fundamental limitation: loss of structural detail. This is due to the fact that amplitude variations and subtle waveform distortions may be blurred by the binning procedure. Moreover, although such models may generate accurate numerical forecasts, they do not inherently resolve the interpretability problem, and in fact are making the procedure even more complex and unexplainable. The output remains a predicted value or token sequence, not a structured explanation of the underlying structural deviation. Thus, the black-box limitation persists, albeit within a larger and more generalized architecture.

2.3.2 Time-Series as Images (Visual Prompting)

A more recent and yet-to-be-explored direction in this research field proceeds to reframe time-series analysis as a computer vision task. Natively multimodal Vision–Language Models (e.g., GPT-4V, Gemini Pro) allow time-series subsequences to be rendered as visual plots and evaluated through visual prompting.

Through transformation of numeric data into graphical representations, this approach aims to overcome the discretization issues and allows the model to leverage pre-trained spatial reasoning capabilities. Instead of reasoning over textual tokens only, the model also analyzes waveform geometry, symmetry, periodicity, and structural distortion in the visual domain.

The necessity of visual prompting over raw numerical text has been recently formalized in the literature. A comprehensive 2025 study by Zhou and Yu [6] demonstrated that Large Language Models possess a distinct 'Visual Advantage' for anomaly detection, performing significantly better when processing time-series data as images rather than as text tokens. This empirical finding validates the

transition toward multimodal architectures, as rendering time-series into visual plots bypasses the severe tokenization limits and arithmetic bottlenecks that cripple text-only LLMs.

Recent baseline systems, such as VLM4TS [7], demonstrate that VLMs can perform anomaly detection by visually inspecting plotted subsequences. These frameworks validate the hypothesis that foundational multimodal models can identify structural irregularities without task-specific retraining.

However, brute-force visual scanning introduces critical limitations for industrial deployment:

- **Computationally prohibitive overhead:** Sliding a window across an entire dataset and converting every subsequence into an image results in thousands of VLM queries. Given the scale of multi-billion parameter models, this approach incurs substantial API latency and financial cost.
- **Hallucination and Noise Sensitivity:** Without strict structural constraints and proper data pre-processing, VLMs may misinterpret benign visual artifacts such as rendering noise, line thickness variations, or scaling distortions as anomalies. This susceptibility to hallucination leads to inaccurate predictions.

Thus, while visual prompting mitigates tokenization loss, it introduces scalability and reliability concerns.

2.4 Hybrid AI and the Research Gap

To utilize the advantages of Vision Language Models despite their massive hardware requirements and the cost of their inferences, recent research has increasingly turned toward hybrid architectures. This paradigm combines lighter models with larger models, aiming to integrate interpretability, logical consistency, and generalization within a unified framework.

Hybrid systems have demonstrated success in domains such as logical reasoning, program synthesis, and theorem proving. However, their application to time-series anomaly detection—particularly in conjunction with foundational Vision–Language Models—remains largely unexplored.

2.4.1 The Unaddressed Gap in the Literature

A systematic review of existing multimodal anomaly detection frameworks reveals a clear research gap. No current approach successfully integrates:

- A deterministic, mathematically exact subsequence-matching algorithm serving as a computational gatekeeper.

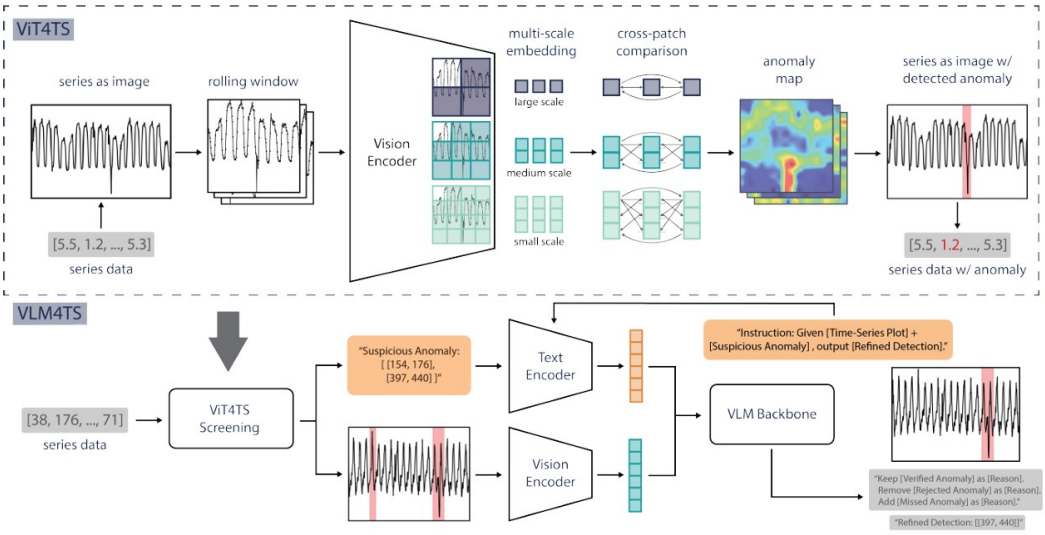


Figure 2.2: Overview of ViT4TS/VLM4TS [7] (upper/lower pane). In ViT4TS, the raw time-series is sliced into windows, and each window is transformed into an image and then embedded into multi-scale feature vectors. By comparing these features to others, ViT4TS localizes potentially anomalous regions and outputs candidate anomaly intervals. In VLM4TS, a Vision–Language Model integrates global temporal context to refine the detection process.

- A structurally constrained foundational Vision–Language Model acting as a semantic interpreter.

As highlighted in recent literature, adapting Vision–Language Models to time-series data introduces a fundamental *resolution-context dilemma*. Early VLM-based prototypes (such as TAMA [8]) relied on naïve rolling-window prompting. While these frameworks demonstrated the potential of visual reasoning, they incurred prohibitive token costs and failed to capture long-range temporal context, rendering them non-scalable for continuous industrial monitoring.

Recent advancements, such as the VLM4TS framework, have attempted to address this computational bottleneck by moving away from brute-force scanning. VLM4TS introduces a two-stage pipeline that employs a lightweight pre-trained Vision Transformer (ViT) as a first-stage visual screener to localize candidate anomalies through cross-patch embedding comparisons, before passing them to a heavier Vision–Language Model for verification.

However, while this two-stage approach successfully mitigates token inefficiency, the architecture remains fundamentally neural–neural. The initial screening stage

relies on learned attention weights and latent space embeddings to evaluate time-series windows purely as images. It therefore lacks mathematical transparency and preserves the opaque *black-box* paradigm, as anomaly candidates are selected based on visual feature dissimilarity rather than deterministic signal geometry. Furthermore, existing pipelines often fail to structurally constrain the final generative outputs of the VLM, leaving them susceptible to visual hallucinations and reintroducing the "illusion of progress."

This thesis directly addresses this gap. It proposes a truly hybrid pipeline in which the Matrix Profile functions as a symbolic, and mathematically precise filtering layer. Rather than relying on neural visual embeddings, the algorithmic layer isolates candidate discords based strictly on exact Euclidean distances. Only these structurally suspicious subsequences are passed to a strictly constrained Vision–Language Model, which operates not as a free-form generator but as a deterministic topological parser.

By integrating algorithmic exactness with semantic reasoning under explicit structural constraints, the proposed framework aims to demonstrate that industrial anomaly detection can simultaneously achieve:

- Zero-shot operational capability,
- Explicit semantic explainability,
- Computational scalability.

without reverting to the opaque black-box paradigm that characterizes legacy deep learning systems.

Chapter 3

The Theoretical Background

3.1 Traditional Deep Learning for Sequence Modeling

Historically, research in the area of time-series anomaly detection has been dominated by statistical methods and Deep Learning architectures. Before the emergence of multimodal foundation models, time-seriesanomaly detection relied heavily on recurrent architectures that are inherently designed to process chronological data which makes them perfect for fields such as natural language processing, finance, and in general any tasks based on sequential data.

3.1.1 Recurrent Neural Networks (RNNs) and LSTMs

Recurrent Neural Networks (RNNs) were initially the standard for sequential data because they maintain an internal hidden state that acts as a memory of previous inputs. However, standard RNNs suffer severely from the *vanishing gradient* problem, which makes them incapable of learning long-term dependencies in extensive time-series data scenarios.

To mitigate the limitations caused by vanishing gradient, the Long Short-Term Memory (LSTM) architecture was later introduced. LSTMs use a gating mechanism that consists of forget, input, and output gates that regulate the flow of information through a memory cell. Formally, for input x_t, hidden state h_{t-1}, and cell state c_{t-1}:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3.1}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3.2}$$
$$\tilde{c}_t = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{3.3}$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{3.4}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{3.5}$$
$$h_t = o_t \odot \tanh(c_t) \tag{3.6}$$

This architecture enables the models to keep contextual information over longer temporal windows compared to RNNs.

3.1.2 Limitations and the "Black Box" Problem

Despite improvements in temporal pattern recognition, recurrent architectures present two critical limitations for industrial anomaly detection:

Sequential Bottleneck Because LSTMs process data step-by-step, they cannot be efficiently parallelized across time. This inherent feature in the LSTM architecture creates a bottleneck during their usage in the analysis of massive and high-frequency datasets, which leads to high latency and resource consumption.

Lack of Interpretability These models remain closed systems that require task-specific training and lack explainability. They often suffer from the previously introduced "illusion of progress," where strong numerical performance on limited datasets covers the weak semantic understanding of actual system dynamics which explains their underperformance in a real-world setup. Furthermore, while they output anomaly probabilities, they do not provide topological or structural explanations for why an anomaly occurred.

Transitioning away from sequential bottlenecks and opaque reasoning naturally motivated the development of attention mechanisms and the Transformer architecture (detailed in Section 3.2), which process entire sequences in parallel and enable scalable multimodal reasoning.

3.2 Mathematical Algorithmic Discovery: The Matrix Profile

While deep learning models provide high-level representational capacity, they are computationally intensive. To establish a highly efficient, deterministic baseline for

anomaly discovery, this framework utilizes the Matrix Profile, a data structure that computes nearest-neighbor distances between all subsequences of a time-series.

3.2.1 Fast Fourier Transform (FFT) and Window Size (m)

Before computing a Matrix Profile, an optimal subsequence length (window size) m must be defined to capture exactly one cycle of the signal motif. To automate this selection without manual domain tuning, the pipeline employs the Fast Fourier Transform (FFT).

Given a time-series signal $T = \{t_1, \ldots, t_n\}$, the discrete Fourier transform is:

$$X_k = \sum_{n=0}^{N-1} t_n e^{-i2\pi kn/N} \tag{3.7}$$

The dominant frequency $f_{\max}$ corresponds to the maximum magnitude in the frequency spectrum. The estimated period is then:

$$P = \frac{1}{f_{\max}} \tag{3.8}$$

The window size is defined as:

$$m = \lfloor P \rfloor \tag{3.9}$$

This ensures that subsequences align with the intrinsic periodic structure of the signal.

3.2.2 Distance Profiles and Euclidean Geometry

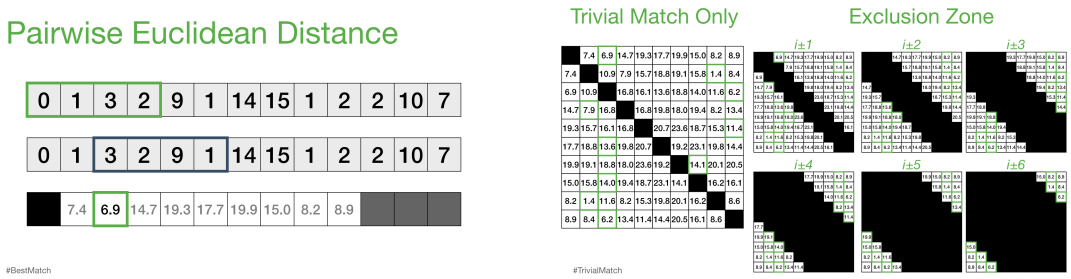
Given a time-series T and a subsequence Q of length m, the Distance Profile is a vector containing the Euclidean distances between Q and every sliding subsequence T_i of length m within T. If subsequences are z-normalized to eliminate baseline shifts, the Euclidean distance D between Q and T_i is:

$$D(Q, T_i) = \sqrt{\sum_{j=1}^{m} (q_j - t_{i+j-1})^2} \tag{3.10}$$

The Matrix Profile MP stores, for each subsequence T_i, the distance to its nearest-neighbor:

$$MP[i] = \min_{j \neq i} D(T_i, T_j) \tag{3.11}$$

According to Figure 3.1, efficient computation can be achieved in near-linear time using FFT-based convolution techniques.



(a) (b)

Figure 3.1: Image reproduced from the official STUMPY documentation [9]

3.2.3 Motifs and Discords

Analysis of the Matrix Profile reveals two fundamental structural phenomena:

Motifs (Benign Baseline) Global minima in the Matrix Profile correspond to subsequences with extremely close matches elsewhere in the time-series. These represent highly repetitive, stable, and structurally benign system behaviors.

Discords (Anomalies) Global maxima in the Matrix Profile correspond to subsequences whose nearest neighbor is significantly distant. These represent unique, non-repeating structural events—mathematically verified anomalies.

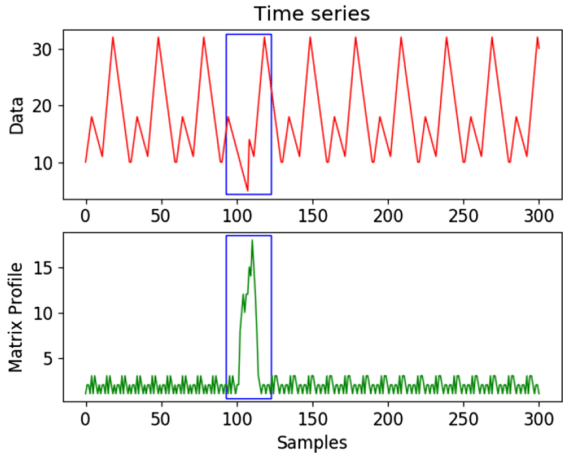


Figure 3.2: Sample matrix profile results [10]

According to Figure 3.2, the cycles with similar shapes to the rest of the time-series show lower values (Motifs) whereas the exact period of irregularity is

pinpointed with high distance values (Discords).

By leveraging this deterministic framework, the Matrix Profile enables direct discovery of the Top-K discords without requiring labeled training data or neural network overhead. It provides an exact, computationally efficient mechanism for structural anomaly detection grounded in Euclidean geometry.

3.3 The Rise of Foundation Models

As earlier deep learning models were highly dependent on the data they were trained on and would underperform in new, unseen contexts, the idea of creating a model that could perform to a certain standard in every scenario emerged. In more professional terms, a model that does not require further training in case of a domain shift.
The emergence of this need led computer scientists to develop the well-known concept of Foundation Models. The foundation models needed to be extremely large to contain information about different topics, and thus, to train such a massive structure to perform well, it required a large amount of data to push their weights to a desirable point that produced the expected results. These trainings subsequently require considerable time and resources and are therefore very costly, as they require thousands of GPU/TPU clusters running for months to make the above process possible. This transition happened during the introduction of the paper "Attention is all you need" [11] in 2017 where the Transformer architectures were introduced.
Previously, sequence modeling tasks were dominated by RNNs and LSTMs. While effective for shorter sequences they posed a bottleneck as they processed the tokens one by one. This linear computation meant that the word N could only be computed after word N-1 was computed. Such limitations prevented the practicality of their usage in parallel computing and thus GPU/TPU usage.Therefore, Such training would take too long to be considered effective.

3.3.1 Self-Attention Mechanism

The introduction of the "Self-Attention" mechanism made such training possible. For every input word it creates three vectors through multiplying the input embedding vector by the trained weight matrices. The matrices are as:

$$Q = XW^Q, \quad K = XW^K, \quad V = XW^V \tag{3.12}$$

1. **Query matrix (Q):** Represents the information a token expects from other tokens.

2. **Key matrix (K):** Represents each token in a space that allows Q to calculate the relevance.

3. **Value matrix (V):** Represents the information content associated with each key, which later is aggregated according to attention weights.

Therefore, the relationship between these tokens is calculated through:

$$QK^\top \in \mathbb{R}^{n\times n} \quad \text{(Score matrix: relevance of token } j \text{ to token } i\text{)}$$

$$\text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big) \in \mathbb{R}^{n\times n} \quad \text{(Attention weights)}$$

$$\text{Output} = \text{softmax}\Big(\frac{QK^\top}{\sqrt{d_k}}\Big)V \in \mathbb{R}^{n\times d_v} \quad \text{(Context-aware representations)}$$

The Q and K matrices are first used to calculate the relevance of tokens and are later divided by the $\sqrt{d_k}$ that is the square root of the K/Q vectors dimensionality in order to stabilize the output product for softmax calculation and lastly the softmax is multiplied by the value matrix (weighted sum of the value matrix rows) to extract the required information for each token generation.

While the self attention was the main mechanism that shaped the core of transformers, the architecture in which it was utilized is equally crucial.

3.3.2 Encoders, Decoders

In the paper [11], the architecture consisted of encoders and decoders 3.3.
Encoder: Accordingly, the input text is first transformed into an embedding and aggregated with its positional embedding. This representation is then passed to a Multi-head attention block. A Multi-head attention block consists of 8 attention heads each attending to different aspects of the contextual relationships present in the input. The input before multi-head attention block is then added to its output and normalized. The result is passed through a Feed Forward Neural Network followed by another add and normalization layer.
Decoder: In the decoder layer, the input text is first transformed into embeddings and combined with positional encodings, followed by a Masked Multi-Head Attention block. Unlike ordinary attention blocks, this one masks future tokens to prevent information leakage. While this block remains in the architecture during inference, when tokens are generated step by step, the masking plays little practical role. It is mainly effective during training or batch generation.

This attention block is followed by an add & Norm layer. Its output is then passed, together with the encoder's output, into a multi-head cross-attention layer, followed by another add & Norm layer that incorporates only the decoder's output. Similar to the encoder, the resulting vectors then pass through a position-wise

feed forward layer and its corresponding add & Norm layer. Finally, the outputs are projected through a linear layer and a softmax to produce the probability distribution over the vocabulary for each token.

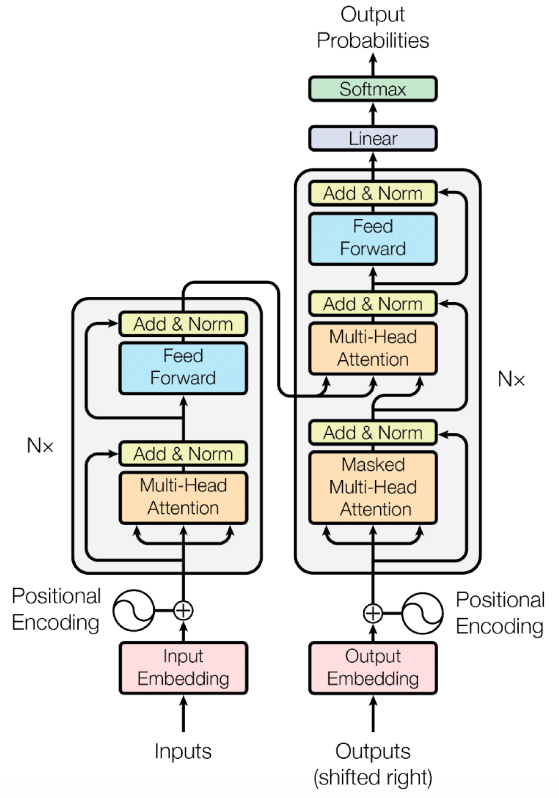


Figure 3.3: Transformer architecture [11]

3.3.3 Representative Paradigms: BERT and GPT

The divergence of transformer models lead to appearance of two dominant families of models each having their own areas of strength.

BERT [12]: It was introduced by Google in 2018 BERT (Bidirectional Encoder Representations from Transformers) as unlike the original transformer architecture, it only utilizes the encoder side and proceeds to read the entire input sequence at once which explains the bidirectionality.

It was trained through two main training strategies. One being Masked Language Modeling, in which, as seen in Figure 3.4, random words in the input sequence are masked and are to predict the masked word solely based on context. The other is the Next Sentence Prediction in which as the name suggests given an input the model is tasked with labeling a second sentence as following the first or otherwise.

The above qualities led to the model's success in NLU tasks such as sentiment analysis, NER, and Question Answering tasks.

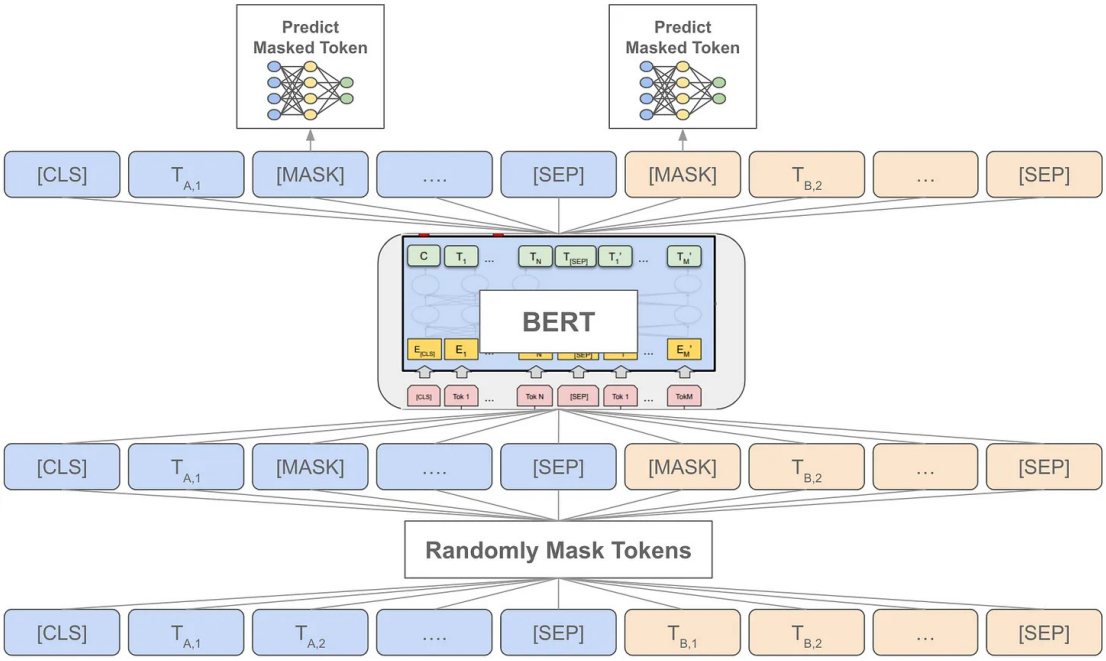


Figure 3.4: BERT's Masked Language Modeling approach [12]

GPT[13]: Unlike BERT's encoder-only structure, GPT (Generative Pre-trained Transformer) utilizes only the decoder side of the transformer architecture. It is designed to be auto-regressive which means it predicts the next token based on all previous tokens. Similarly to the transformer decoder architecture it uses masked attention that prevents GPT from seeing future tokens and can only see the left-hand context.

What set these models apart was their ability to perform tasks without finetuning through simply providing a prompt or giving them few examples of questions and answers expected (Currently known as the few-shot prompting).

These models were trained in an unsupervised setup on massive datasets without human labels as the Causal Language Modeling strategy simply trains models by providing the previous tokens and expecting the next one. This simple yet effective strategy does not require human labeling as the ground truth is already in the text itself and the model learns to mimic how documents are written.

Once the model learns how to read, it is then trained to follow instructions through Supervised Fine-Tuning which provides the model with a dataset of golden examples consisting of prompts and their perfect answers which helps with model changing roles from a text completer to an assistant.

Finally, the Reinforcement Learning from Human Feedback technique is used in which a user provided with a prompt is presented with multiple possible answers and is asked to rank them from best to worst.

The above qualities led to GPT family having the best results in the Natural Language Generation field such as Open-ended text generation (essays, stories or reports from scratch), conversational AI (chatbots such as chatGPT) and Code Synthesis (Programming bots through explanations in a prompt)

3.4 From Text to Vision: Multimodal Foundation Models

While the aforementioned models revolutionized the Natural Language Processing, they immediately initialized the enthusiasm and the need for models that could interpret and interact with non-textual information as well. This led to many proposals in how to make the usage of other types of data feasible such as images or voice data as much as the textual data.

3.4.1 ViT (Vision Transformer)

Since transformer architectures are designed fundamentally based on sequences, a strategy was to be considered to turn images to transformer-optimized input structure.

Therefore, Vision transformers were introduced. A solution that first divides images into fixed-sized squared patches (eg. 16×16)pixels. Then a linear layer transforms each the raw pixel vector into a patch embedding. These patch embeddings are later aggregated with their positional embeddings to preserve their positional information and finally just like BERT, a [CLS] label is also added to gather global features from other patches as well. This represents the entire image.

The results from above embeddings are passed into a transformer encoder with the previously explained multi-head attention block with add and normalization layer on top of it. The self attention block helps every patch to attend to all other patch which enables the model to capture dependencies over the entire image. This global receptive field is the key strength of Vision Transformers with respect to Convolutional Neural Networks that primarily depend on local convolutional filters and gradually build the global context in the next layers through depth and pooling. In contrast, ViT can model long range relationships between distant regions of the image in the very first layer.

ViT like every transformer is data hungry and requires a massive amount of data to train while CNNs already have built-in understanding of the images as the structure already uses locality assumption, kernels that are designed to find edge,

kernels that use the same weights in the entire image. However,ViT, after being trained enough it almost always outperforms the traditional approaches. Although,

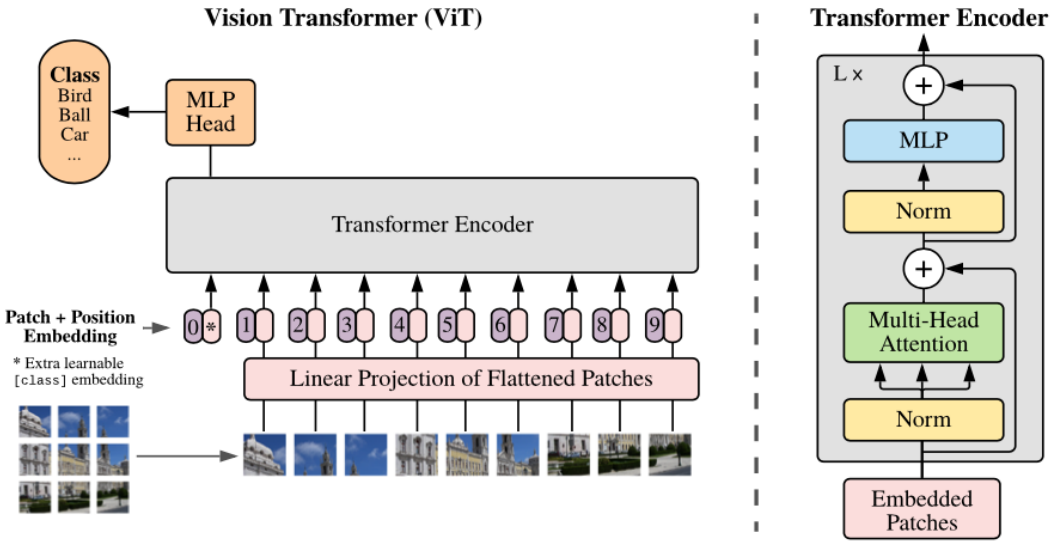


Figure 3.5: Vision Transformer Architecture in image classification [14].

Vision Transformers are successful at learning visual representations, they are purely operating in the visual domain and have not yet been connected to the natural language domain. This limits the capabilities of the model to follow tasks and instructions. To enable these models to understand images through language a mechanism is required to connect these two domains. The vector representation of the word "dog" should lie close to the vector representation of the image "dog" within a shared embedding space.

3.4.2 CLIP (Contrastive Language-Image Pretraining)

To perform such alignment, OpenAI introduced CLIP, a Multimodal model designed to align visual and textual representation within a shared embedding space. Despite classical approaches in which a range of classes were defined and the model was to classify the image in the finite dataset, this approach utilizes what foundation models are best at. The pipeline is designed so that it can train of vast amount of image-text pair data from the internet.

It consists of two encoders as seen in Figure 3.6, an image encoder that is often a ViT and a text encoder that is a transformer based language model. Given a batch of images and their corresponding captions, encoders map their corresponding inputs into a vector embedding and then these embeddings are compared for

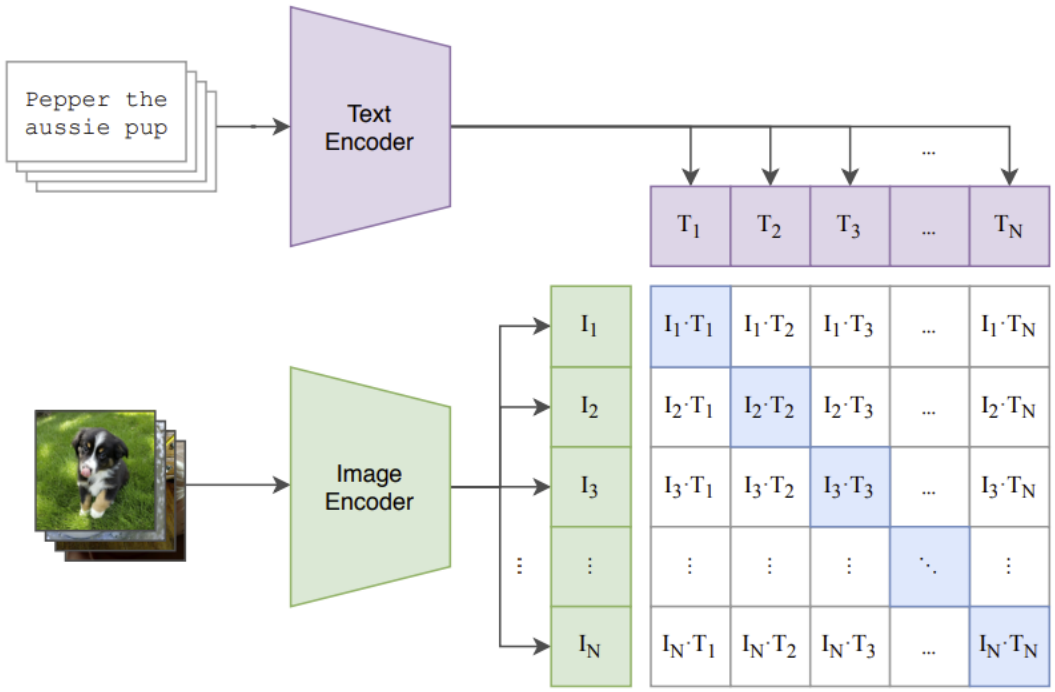


Figure 3.6: Contrastive Language-Image Pretraining Architecture [15]

similarity through vector multiplication. The embeddings of similar image and text pairs are pushed closer to each other and distant ones are pulled apart. This encourages the model to organize visual and linguistic concepts in a shared semantic space. As a result, CLIP can perform zero-shot classification by comparing an image with the prompt given such as "a photo of a running dog" or "A yellow car" without the required task-specific retraining. By grounding visual representations in natural language, CLIP enables interaction between images and text, allowing models to generalize to new categories and instructions simply through prompts.

This training strategy, as previously mentioned:

- overcomes the issue of finite classes,
- enables self-supervised training over vast amounts of naturally paired data from the internet,
- provides information-rich supervision that encourages embeddings to represent the semantics behind images in addition to the objects themselves.

and thus has been one of the most prominent strategies.

However, due to the text and image encoders operating independently and their result being the product of a late fusion, proper interaction between these data cannot be fulfilled by encoders. This issue could potentially limit the performance of our models as there is a lack of attention between text and visual data.

3.4.3 Native Multimodal Models

To permit each representation to influence the other during encoding, multimodal models such as Gemini and GPT-4o were introduced. As shown in Figure 3.7, Gemini's architecture, first presented in their official paper [16], demonstrates how such co-influences are made possible.

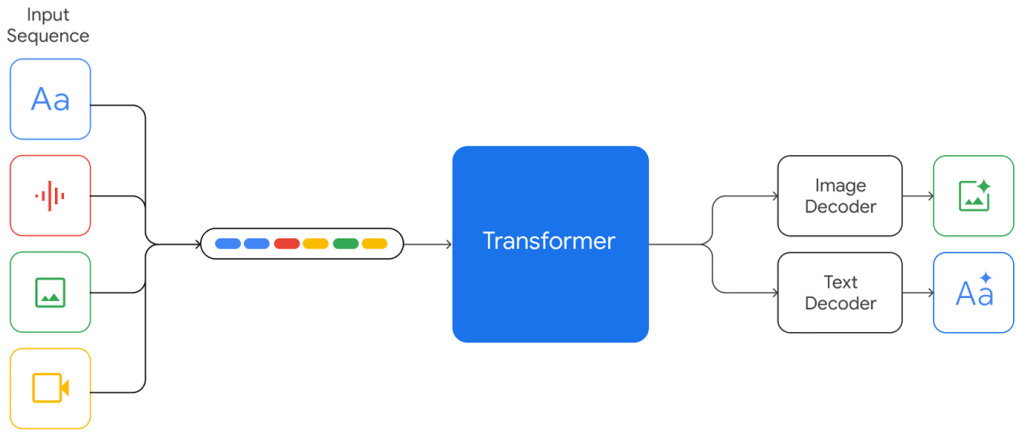


Figure 3.7: Native Multimodal Architecture [16]

Since transformers are designed to work with linear sequences, each of the data types should be preprocessed to match the transformer's requirements. Therefore, each of the input data types such as text, audio, image and video are first tokenized. However, these data types are rather unique and require their own tokenization procedure:

- **Textual Data:** Text input is processed through a subword tokenizer. These tokens are mapped into the same high-dimensional embedding space as the other modalities.
- **Image Data (Spatial Tokenization):** Images are considered as discrete visual entities within the input data. As Gemini supports variable input dimensions, the model is able to allocate more computational resources to tasks that require better understanding, such as identifying microscopic glitches in a signal plot. And thus images are separated into patches that are later encoded as a sequence of visual tokens.

- **Audio Data (Direct Signal Ingestion):** An important technical aspect of the Gemini architecture is its ability to directly ingest audio signals at 16kHz using features derived from the Universal Speech Model (USM) [17] developed by Google itself.
- **Video Data (Spatio-Temporal Sequences):** Video is handled as a temporally indexed sequence of image frames. This allows the model to capture the evolution of a signal over time, which is critical for identifying non-stationary anomalies. By default, the architecture samples video at a rate of 1 frame per second (FPS), though this can be adjusted for "fine-grained" tasks. Each frame is tokenized and placed sequentially within the large context window (up to 1M+ tokens), enabling the model to "remember" events from the beginning of a long recording and compare them to current states.

After having tokenized the input data, they are then placed in the sequence in the exact order they appear in the "conversation" or document which means Gemini allows interleaving. For example, if provided a manual and a sensor plot, the input sequence would be as follows:

[Text Token 1] | [Image Token 1] | [Video Token A] | [Audio Token 1]

Once these inputs are projected into a unified embedding space, they are fed as a single token sequence. Then they are fed to the Transformer backbone. This enables the Self-Attention mechanism to operate across the entire sequence and thus every data type in at the same time. It can attend to the textual instructions, the audio cues, video frames, and image pixels to determine the answer to the user's prompts. Then the output generated by the transformer backbone is given to a textual and an image decoder to provide the desired output.

In summary, the transition from sequential RNNs to natively multimodal Transformers like Gemini represents a paradigm shift in data processing. By encoding diverse inputs into a unified token stream, these models can perform complex cross-modal reasoning within a massive context window. Having established the architectural foundations of these models, the following chapter will detail the specific methodology utilized to leverage these capabilities for zero-shot anomaly detection in high-frequency signals.

Chapter 4

Methodology

The proposed methodology utilizes a multi-step pipeline that combines the mathematical power of the Matrix Profile (MP) with the semantic reasoning of a Vision Language Model (VLM). The pipeline is designed as a combination of approaches that address the interpretability gap found in traditional unsupervised methods.

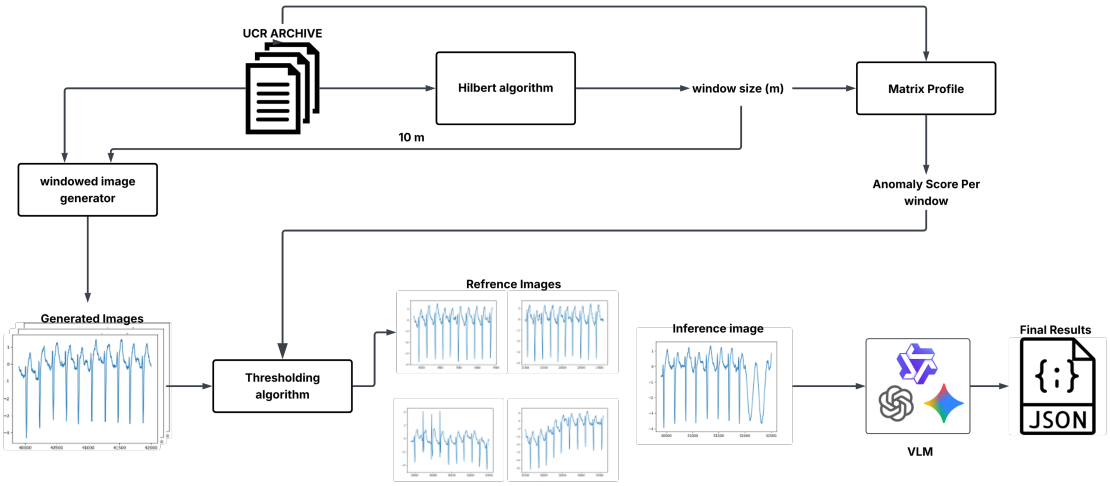


Figure 4.1: The architecture of the proposed hybrid pipeline begins with signal preprocessing and window size calculations (Phase 0), followed by Matrix Profile-based discord discovery (Phase I) to filter out simpler windows and select the 1 + 3 references. These windows are then transformed into visual representations (Phase II). Subsequently, suspicious windows containing the top-K discords are provided to the VLM along with the references. The final output is extracted from the VLM in JSON format.

- **Algorithmic Feature Extraction via the Matrix Profile:** the Matrix Profile is specialized in identifying discords that are distinctive compared to the rest of the time-series. While the MP is effective at finding these mathematical anomalies, it lacks the domain-specific context to distinguish between benign, yet rare events and critical system failures. To tackle this, our pipeline utilizes the Matrix Profile as a filter that goes through the dataset to mark suspicious candidates. This significantly reduces the computational payload for the VLM, so that it only processes highly suspicious segments.
- **Visual Conditioning and Preprocessing:** After the candidates are pinpointed, they are preprocessed into window segments. These segments are not simply images; they are visualized through adaptable segmenting algorithms to maximize the VLM's perception of the target suspicious segment. This step is the bridge between raw data and visual tokens that change a signal processing problem into a visual reasoning task.
- **Multimodal Reasoning:** These visualized windows are then presented to the multimodal model, which leverages its native multimodal foundations to provide a qualitative interpretation of the anomaly. By providing a text-based system prompt alongside the visual data, the model utilizes its reasoning power. It evaluates the shape of the signals, their frequency and amplitude then compares it to the learned patterns through few-shot examples provided.
- **Computational Efficiency and Explainability:** This integrated pipeline allows for a detection system. A system that while computationally efficient through filtering easily distinguished segments from being passed to the VLM is also inherently explainable. Unlike black-box neural networks that provide a simple anomaly score, this framework allows the model to output an explanation in natural language that provides the reasoning behind each detected discord.

In the following sections, the above processes are explained more thoroughly but before diving into the pipeline properties, the dataset and its characteristics will be explained.

4.1 Phase 0: Signal Pre-processing and Mathematical Grounding

Before the data can be processed by the Matrix Profile or the VLM, the continuous time-series must be discretized into meaningful units due to the architecture of these models and how they rely on windowed data rather than signal numerical data itself. This process, that is widely known as windowing is the most critical preprocessing step, as it defines the resolution at which an anomaly can be detected.

4.1.1 Defining the Window Length (m)

The window length, usually referred to as m, represents the length of the window the model analyzes at each time step. Selecting m is not a trivial task and thus it requires domain knowledge of the signal's periodicity.
The Motif Principle: In an ideal setup, m should be long enough to capture at least one full cycle of the normal repeating pattern (the motif) as this represents the true shape that is under analysis. For example in case of an ECG signal, m should at least span one heartbeat which shows the shape of the signal and carries the essential information about heart pumping. However, to capture the relationship and distance between each heartbeat, a window size of multiple ms could also be suggested.
Sensitivity: If m is too small, the system becomes overly sensitive to noise and thus would lead to high false positives. If m is too large, a short-duration anomaly may be averaged out or appear too small, which would make it invisible to the model and consequently lead to high false negatives.

To identify the most robust windowing strategy, four candidate signal processing algorithms were evaluated:

- **Hilbert Burst Detection:** This method utilizes the Hilbert Transform to calculate the amplitude and frequency of the signal. It was evaluated for its ability to isolate transient energy "bursts" that could define the boundaries of a window.

- **Autocorrelation Function (ACF):** A classic temporal approach where m is determined by the first significant peak in the autocorrelation plot. This identifies the lag at which the signal most strongly repeats itself.

- **Energy-Based Segmentation:** This approach calculates the short-time energy of the signal, setting m based on the duration of high-energy events. This is particularly useful for impulsive anomalies but proved less effective for subtle structural drifts.

- **Dominant Frequency Analysis (Selected Strategy):** As seen in Algorithm 1 the cycle length is calculated through Fast Fourier Transform (FFT) with which we can identify the primary frequency component (f_{dom}). The window size is then set as the inverse of this frequency ($m = 1/f_{dom}$) and it effectively calculates one full cycle of the system's normal behavior if applied to the benign segment. However, since the VLM's large context can analyze more than just one cycle, the window size was chosen as ten cycle lengths to keep the model's perception accurate enough and yet include as much information as possible inside the windows.

The selection of the Dominant Frequency heuristic is due to its adoption in the paper DAMP (Discord Awareness the Matrix Profile)[18] that is written by the creators of the UCR Archive and the Matrix Profile themselves. This ensures that the windowing on the Matrix Profile matches that of the VLM as well. But most importantly, it showed a greater flexibility towards change required per data in window sizes.

Algorithm 1 FFT-Based Cycle Length Estimation

Require: Signal x
Ensure: Estimated cycle length m
$N \leftarrow \text{length}(x)$
$x \leftarrow \text{GaussianSmooth}(x, \sigma = 2)$
$F \leftarrow \text{FFT}(x)$
$f \leftarrow \text{FrequencyBins}(N)$
Keep only positive frequencies of F and f
$M \leftarrow |F|$ ▷ Magnitude spectrum
$M[0] \leftarrow 0$ ▷ Remove DC component
$S \leftarrow M$
$m \leftarrow N$
while $10m > N/8$ **do**
    $i \leftarrow \arg\max(S)$
    $f_0 \leftarrow f[i]$
    if $f_0 = 0$ or $|f_0| < 10^{-10}$ **then**
        $S[i] \leftarrow 0$
        continue
    end if
    $m \leftarrow \lfloor 1/f_0 \rfloor$
    $S[i] \leftarrow 0$
end while
return $10m$

Contextual Scaling and Few-Shot Anchoring While the Matrix Profile utilizes the window size of m to preserve high sensitivity to discords present in the dataset, VLMs can benefit from a broader contextual field. Consequently, the expansion of the window size to $10m$ during the interpretation phase could potentially lead to improvements in performance.

Furthermore, this $10\times$ expansion is critical for the few-shot learning stage. The larger window enables extraction of more information from segments that are given to the model as a reference. These references, that present the model with

$10m$ of normal system behavior allow the model to observe the stability of the dominant frequency and motif across multiple cycles. This extended baseline makes deviations within the $10m$ test window significantly more salient, as the model can visually and semantically contrast the anomalous event against a high-capacity reference of benign behavior.

4.1.2 The Sliding Window Mechanism

To prepare the data, we utilize a sliding window approach across the dataset. Given a time-series T of length n, a window size m, and a stride s, the total number of subsequences produced would be:

$$N_{\text{windows}} = \left\lfloor \frac{n-m}{s} \right\rfloor + 1.$$

Thus each subsequence A_i is defined as

$$A_i = \{t_i, t_{i+1}, \ldots, t_{i+m-1}\}.$$

This sliding mechanism guarantees that no part of the signal is missed. While the Matrix Profile uses the stride $s = 1$, which produces

$$N_{\text{MP}} = n - m + 1$$

subsequences, the windows for the VLM are constructed in a manner to avoid overlap. Therefore, it has a stride equal to the window size, $s = m$, and thus the VLM sees only non-overlapping windows, and subsequently, the number of windows becomes

$$N_{\text{VLM}} = \left\lfloor \frac{n-m}{m} \right\rfloor + 1 = \left\lfloor \frac{n}{m} \right\rfloor.$$

This construction ensures that multiple windows of the same segment are not created, which naturally would prevent unnecessary increases in the number of windows processed by the VLM and eventually reducing runtime and operational cost.

4.1.3 Pre-processing for the Matrix Profile Discovery

While the VLM receives visualized raw signal data to preserve visual and magnitude context, the data stream that is fed to the Matrix Profile algorithm requires relatively extra preprocessing steps to maximize discord sensitivity. These steps ensure that the Matrix Profile is not underperforming due to baseline drifts or low-level noise that are usually patterns that the Matrix Profile algorithm cannot overcome.

A. Baseline-Anchored Global Normalization To make sure the Euclidean distance metric inside the Matrix Profile algorithm is not confused by baseline drifts, we need to perform global normalization on our data. However, To ensure the pipeline remains causal and free from data leakage, normalization parameters should be calculated from the training (benign) segment. The mean (μ_{train}) and standard deviation (σ_{train}) are calculated from the first k points of the archive that is set by the dataset's publisher. Therefore, every subsequent data point in the Matrix Profile is then transformed with normalization as below:

$$\hat{t}_i = \frac{t_i - \mu_{\text{train}}}{\sigma_{\text{train}}}.$$

This, as desired, centers the signal values at zero and scales its variance to a unit which allows the Euclidean distance metric to only capture the cycle shape deviations rather than raw magnitude shifts.

B. Magnitude Sharpening (P^{10}) Following the initial calculation of the Distance Profile, as the paper [18] suggests, we apply a sharpening transformation of magnitude 10. This step is intended to amplify the difference between the signal's motifs and its discords:

$$P_{\text{sharp},i} = (P_i)^{10}.$$

By raising the distance values to the tenth power, we effectively punish deviations and thus increase the sensitivity of the Matrix Profile depending on the criticality of small but important deviations. This sharpening is critical for Phase II, as it allows the pipeline to pinpoint the exact center of a discord, ensuring more precise instructions to the VLM.

4.2 Phase I: Algorithmic Discord Discovery

Having prepared the windows for both models, the pipeline can initiate the discovery phase. The objective of this phase is to reduce the thousands of windows (N_{VLM}) created in the previous section to a highly targeted set of candidate windows containing discords.

4.2.1 the Matrix Profile as a Discovery Tool

As described above, the primary objective of Phase I is to filter the previously created windows and extract the most informative windows as references while for the inference phase, it keeps the highest-probability anomalous windows to be further analyzed by the VLM. To achieve this, we utilize the Matrix Profile (MP).

Distance Profile Computation For every window A_i of length m in the time-series, the Matrix Profile performs an all-pairs similarity search. The resulting distance profile P stores the distance between A_i and its most similar match. Formally, for a distance function dist, the Matrix Profile value at index i is

$$P_i = \min_{j \in [1, n-m+1], j \neq i} \text{dist}(A_i, A_j). \tag{4.1}$$

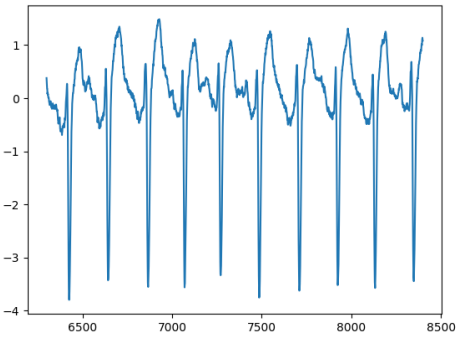
In this research, we use the Euclidean distance in the Matrix Profile to calculate the difference between z-normalized subsequences. As mentioned in the preprocessing section, Z-normalization is essential because it makes the Matrix Profile invariant to differences in offset and amplitude, and helps it focus on structural discords ,rather than shifts in signal magnitude.

Interpretation of Discords In the Matrix Profile theory, low values indicate motifs (repeating, normal patterns), whereas high values represent discords. In this paper, discord is defined as a cycle that is different from all other cycles in the data. Through identification of the cycles that correspond to these maximum differences, the pipeline flags timestamps, representing the most unique cycles in the UCR dataset. These flagged indices serve as an indicator that pinpoints the windows containing discords. Windows that will later be fed into our VLM.

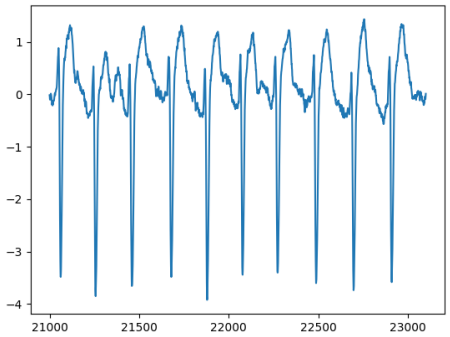
4.2.2 Phase I, Part A:Curation of the Few-Shot Baseline

To prepare the VLM, so that it can distinguish between benign fluctuations and true failures, the pipeline must provide a representation of normal behavior. However, the context which can be provided is limited. The more visual context given, the more computation heavy and expensive the process would become. Therefore, instead of using the entire training sequence as a reference to the VLM, we need to carefully select the most informative ones. Consequently, rather than selecting training windows at random, we use the sharpened the Matrix Profile of the training set as a statistical filter to curate a Few-Shot Reference Set. To establish this set, we first calculate the full the Matrix Profile on the training set and store the distance profile values. Then using this result, the below windows are chosen:

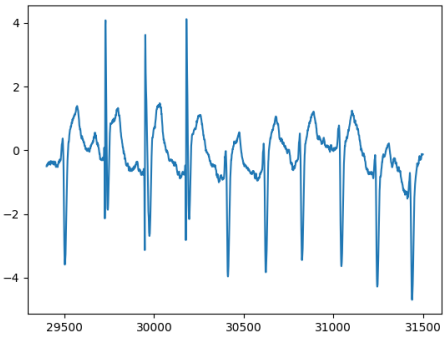
A. The "Golden Motif" (Global Minimum) The window i where the training distance profile P_{train} reaches its global minimum. This represents the most consistent and repeating pattern in the benign history that represents the "ideal" state of the system. This window serves as the Anchor of Normalcy, that provides the VLM with a clear visual definition of perfect cycles.



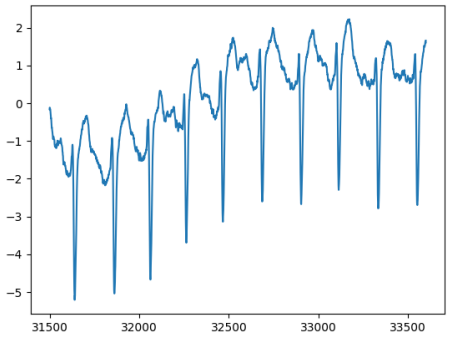
(a) Least anomaly score



(b) Top3 highest anomaly score



(c) Top3 highest anomaly score



(d) Top3 highest anomaly score

Figure 4.2: The training samples chosen based on their anomaly scores.

B. The "Boundary Motifs" (Top-3 Benign Peaks) To establish the limits of acceptable variance and system behavior that without a previous encounter could be mistaken with an anomaly, the pipeline selects three windows corresponding to the highest sharpened peaks within the training set. While these are technically "benign" (occurring within the verified training segment), they represent the upper limit of system noise.

By presenting these four mathematically chosen windows (1 Golden + 3 Boundary samples Figure **??**) to the VLM, we create a Comparative Reasoning Environment. This demonstrates for the model what normal behavior is and identifies minor structural fluctuations that do not exceed the variance seen in the Boundary Motifs and should be ignored. This approach effectively reduces false positives during the interpretation phase.

4.2.3 Phase I, Part B: Inference and Discord Discovery

Once the baseline references are established, the pipeline begins the discovery process on the inference (test) segment. This phase is designed to be selective, ensuring that only the most unique events are passed to the VLM.

Implementation of the Left Matrix Profile

To simulate a real-world monitoring environment, the pipeline utilizes a Left Matrix Profile. Unlike the training phase in which searches happened globally, the inference search for any window A_i is restricted to the "seen" history:

$$P_i^{(L)} = \min\Big(\text{dist}(A_i, A_j)\Big) \quad \text{where } j < i.$$

This ensures that a high distance score truly represents a novelty. If a pattern was previously seen in the training set or test set, the distance score will remain low, preventing the VLM from being alerted to recurring, known patterns. However, depending on the task this algorithm could be adjusted. In a real-world scenario this left-the Matrix Profile should have limited memory so that it does not miss an already seen anomaly simply due to the same anomaly happening a second time. But in the current scenario such adjustment would not create any difference in the behavior of the system as only one anomaly exists in the dataset.

Ranking and Dynamic Thresholding

To choose candidate windows for the VLM, the pipeline, rather than applying a fixed value, it proceeds to apply a statistical filter to the distance profile. This percentile-based Thresholding strategy accounts for varying noise floors across different UCR datasets.

A. Threshold Calculation The threshold (τ) is defined as the 99.55^{th} percentile of all values within the inference Matrix Profile (P_{inf}):

$$\tau = \text{percentile}(P_{\text{inf}}, 99.55).$$

B. Suspect Window Identification A window is flagged as "suspicious" if its maximum the Matrix Profile value exceeds the previously calculated threshold ($\max(W) > \tau$).

This strategy ensures that the pipeline only considers the most extreme 1.5% of deviations as potential anomalies which significantly reduces the computational load on the VLM and filters out noise.

Precision Boundary Identification and Padding

Once a window is flagged, the pipeline proceeds to localize the period inside the window that contains the anomaly. This ensures that the image sent to the VLM is not just a random slice, but is also accompanied by information from the Matrix Profile.

A. Anomalous Period Delineation The pipeline identifies all indices within the window where the Matrix Profile values exceed τ. We define the Core Anomaly Boundaries as the span between the first and last points above the threshold:

$$t_{\text{start}} = \text{first}(i \ : \ P_i > \tau), \qquad t_{\text{end}} = \text{last}(i \ : \ P_i > \tau).$$

B. Padding Extension To ensure the VLM receives the complete geometric profile of the failure, a buffer is applied to both boundaries of the MP's flagged discord. While MP pinpoints the mathematical deviations from normality, its sliding window can occasionally cause a slight offset relative to the actual cycle period. To compensate for this misalignment and to ensure the window contains the entire anomalous period, the boundaries are extended by $0.25m$ (one-quarter of the fundamental cycle length).

$$\text{Suspicious\_Period} = [t_{\text{start}} - 0.25m, \ t_{\text{end}} + 0.25m]$$

This padding logic helps with geometric centering. By appending a quarter-cycle to each end of the period, the pipeline guarantees that even when the Matrix Profile flags a subsequence that is slightly out of phase with the actual anomaly, the resulting frame will still contain the full structural "signature" of the event. This ensures that the features are not truncated and provide the VLM with a complete view of the discord.

4.3 Phase II: Visual Transformation and Multimodal Interleaving

4.3.1 Signal-to-Pixel Rendering Protocol

The transition from time-series data points to a visual representation serves as a critical step. To ensure that the VLM can interpret the data, the rendering process follows a protocol that is designed to provide clear numerical context without introducing unnecessary visual complexity.

A. Coordinate Retention (Axes and Ticks)

Unlike the traditional computer vision tasks that require completely clean images, this pipeline deliberately retains the X and Y axes, including their respective tick marks and numerical labels. By providing the numerical scale, the model is guided to reason about the specific magnitude and temporal location of deviations relative to the established baseline. Furthermore, it also opens the possibility of further interaction of the textual prompts with the image. Papers such as [19] also support the performance of VLMs such as Gemini on plot reasoning tasks.

B. Grid Suppression for Feature Clarity

While the axes are maintained, all background grid lines are suppressed. This decision was made to maximize the signal-to-noise ratio for the VLM. In a time-series context, grid lines can often intersect with high-frequency signal components, and despite the grid's color intensity with respect to the plot, it could create visual artifacts that may confuse the model's attention heads. By removing the grid, the signal itself remains the primary high-contrast feature in the image, while the axes remain at the sides to provide the necessary information.

C.Resolution Standardization and Structural Fidelity

All visualizations are rendered at a fixed resolution of 640×480 pixels, corresponding to a 100 DPI density over a 6.4×4.8 inch canvas. This specific resolution was selected to optimize the *pixel-to-information* ratio for the VLM visual encoder while keeping the balance between cost and performance at a reasonable point.

VGA Standard Alignment The 640×480 resolution provides a standardized aspect ratio that prevents the VLM from performing non-linear stretching or warping during the preprocessing stage. Maintaining a consistent frame size across all datasets leads to the spatial frequency of the signal remaining comparable across different inference windows.

Feature Resolution at 100 DPI This pixel density is sufficient for the critical geometric markers that are to be identified by the VLM must remain visible without introduction of the computational weight that is usually resulting from high resolution images.

Lossless PNG Integrity While the resolution adheres to a standardized scale, the use of the PNG file type, it is ensured that the signal topology and numerical tick marks remain pixel-perfect. This eliminates the risk of JPEG compression which

could lead to blurring of sharp edges of a discord and consequently misinterpretation of compression noise as signal jitteriness by the VLM.

4.3.2 Comparative Layout Construction

After rendering high-fidelity, grid-less 640×480 PNG frames, the pipeline organizes these windows into a structured prompt designed to enable relational reasoning. A central methodological innovation is the transition from isolated anomaly detection to comparative few-shot prompting.

Instead of presenting a single inference window in isolation, the system assembles five distinct visual inputs to establish a comparative *ground truth*:

- **The Golden Motif (Reference 1).** The global minimum of the training distance profile, representing the *ideal operational state* of the signal and serving as a baseline for perfect periodicity.
- **The Boundary Motifs (References 2–4).** Three windows corresponding to the highest benign anomaly scores identified during the Matrix Profile discovery. These capture rare yet non-pathological fluctuations and define the upper bound of structural variance.
- **The Inference Candidate (Discord).** The target window selected via the Top-K the Matrix Profile scan and subjected to semantic evaluation.

By arranging these frames sequentially within a single prompt, the Vision Language Model (VLM) is compelled to perform contrastive analysis. It must distinguish structural variance (as illustrated by the boundary motifs) from genuine anomalies (e.g., frequency drift or waveform distortion). This configuration of images is the fundamental basis on which the model's reasoning process is exposed to the signal data itself during inference and allows comparison with the specific signal during its benign period.

4.3.3 Metadata Augmentation and Temporal Localization

While visual representations provide essential topological context, relying solely on computer vision introduces vulnerability to scale misinterpretation, phase ambiguity, or perceptual bias. To fully leverage the native multimodal capabilities of the Vision-Language Model (VLM), the visual layout is augmented with structured mathematical and spatial metadata.

Each of the four reference windows and the inference candidate is interleaved with concise textual descriptors. Critically, this augmentation introduces two anchors derived directly from the Phase I algorithmic discovery process:

Frequency Grounding. The dominant frequency (m), computed via the Fast Fourier Transform (FFT), is provided in the prompt. This strategy of providing the expected periodicity as text alongside the image of the signal, requires the VLM to reconcile geometric shape with frequency-domain constraints. This cross-referencing enforces consistency between the visual topology of the signal and its mathematical characterization.

Temporal Localization. The prompt explicitly mentions the precise time interval at which the Matrix Profile identified a discord. Instead of evaluating a context-free image, the VLM is informed of when the event occurred within the broader dataset. This localization enables reasoning about progression, drift, or transitions relative to past behavior of the signal.

Multimodal Cross-Validation. The multimodal nature of this strategy enables comparison across three sources of information:

- Visual topology (waveform geometry),
- Semantic interpretation (textual reasoning),
- Mathematical grounding (frequency metadata and temporal index).

With the incorporation of explicit frequency expectations and precise temporal coordinates, the VLM is provided with sufficient information regarding normality to reject benign fluctuations while confirming real anomalous behavior. The result is a robust evaluation process that minimizes ambiguity and strengthens interpretability.

4.3.4 Visual Prompting Scenarios and Color Ablation

To determine the optimal visual input for the Vision-Language Model (VLM), the visual transformation phase was engineered to generate two structurally identical representations with a single controlled variable: the use of color to indicate algorithmic suspicion.

Scenario A (Clean Representation) The candidate subsequence identified by the Matrix Profile is plotted as a standard, unannotated line graph. The entire signal is rendered in a uniform color which forces the VLM to evaluate the raw topological geometry of the sequence without external visual cues.

Scenario B (Red Highlight Representation) The structural layout is identical to Scenario A, but the exact interval identified as a discord by the Matrix Profile is explicitly highlighted in red. The idea behind this approach was to visually guide the VLM's attention towards the suspicious region.

By isolating the red highlight as the sole independent variable, these scenarios enable a controlled evaluation of whether explicit visual guidance enhances the VLM's semantic reasoning or instead bypasses the model's geometric interpretation of the signal.

4.3.5 Structural Prompt Engineering and Persona Definition

To ensure consistency across all 1,186 inference windows, the prompt architecture deliberately suppresses conversational behaviors and additional text. (The complete, unedited multimodal prompt is provided in Annex A.)

A. Persona and Scope Assignment. The system prompt initializes the VLM with a narrowly defined analytical role: as *Signal Morphology Analysis Expert specialized at detecting topological anomalies.* This explicit role conditioning anchors the model's internal attention mechanisms toward structural physics rather than generic image recognition or narrative description.

B. Intentional Feature Suppression. A key innovation of the prompt design is the explicit declaration of features to ignore. Time-series visualizations frequently contain superficial variations that can mislead deep models. The prompt instructs the VLM to disregard:

- Absolute amplitude values and baseline offsets,
- Absolute time duration and sampling rate differences,
- Stochastic noise, jitter, and general visual smoothness.

This suppression forces evaluation based strictly on relational geometry and therefore, it prevents false positives induced by benign scaling, shifting, or sensor calibration artifacts.

4.3.6 The Topological Evaluation Protocol

Because the prompt prohibits natural language explanations, the model does not emit a visible Chain-of-Thought. Instead, it processes an internalized algorithmic checklist prior to producing its final classification.

The inference candidate is evaluated against the reference motifs using three structural hierarchies:

1. **Directional Order.** The ordered sequence of directional transitions (upward versus downward movements) must match the reference structure. Missing, additional, or reordered transitions result in anomaly classification.

2. **Relative Amplitude Hierarchy.** Successive extrema (peaks and valleys) are extracted and ranked. The relative ordering (e.g., primary peak > secondary peak) must remain invariant, independent of absolute scale.

3. **Inter-Extrema Distance Topology.** The relative spacing between extrema (e.g., peak-to-valley, peak-to-peak) is ranked from longest to shortest. Temporal compression or expansion is permissible only if this ranking hierarchy is preserved.

Violation of any of the above structural hierarchy results in an anomalous classification. Consequently, preservation of all the above rules leads to a benign classification instead.

4.3.7 Strict JSON Output and Programmatic Integration

The use of large language models in automated pipelines introduces possibility of uncontrolled natural language generation that makes post-processing of the output impossible or error-heavy. To eliminate this variability, the prompt enforces a zero-tolerance policy for unstructured output.

The model is explicitly instructed to respond *only* with a valid JSON object, containing no markdown or extraneous text. The response schema is constrained to:

- `"is_anomalous"`: Boolean (`true` or `false`),
- `"confidence"`: Categorical string (`"high"`, `"medium"`, or `"low"`).

Dual Utility of the Output Format. This strict machine-readable format achieves two objectives:

1. **Deterministic Integration.** The Python evaluation script can directly compare the resuls from the VLM with the ground truth to compute the Affiliation F1 score without reliance on regular expressions or any other post processing techniques as the output is already in the correct format.

2. **Operational Introspection.** The confidence attribute provides a means for internal model certainty that could be of high value in the future extensions such as dynamic thresholding, abstention policies, or active learning feedback loops.

Through structural constraint, hierarchical topology enforcement, and strict output formatting, Phase III transforms the VLM from a generative model into an analytical component within a reproducible industrial anomaly detection pipeline that is capable of explaining the reasoning that is applied inside.

4.3.8 The Diagnostic Variant for Industrial Deployment

Previous variant was utilized during research for the performance analysis to avoid further token generation and costs that could be caused by VLM's reasoning generation for the user.

However in a real world setup, a VLM's strength includes its ability to explain its reasoning process behind the decisions and thus important to provide a version for real-world usage.

To make this possible prompts regarding strict decision-only outputs are replaced with a request for explanation of the logic and the final JSON output was altered to the following version:

- `"is_anomalous"`: Boolean (`true` or `false`),
- `"confidence"`: Categorical string (`"high"`, `"medium"`, or `"low"`).
- `"Reasoning"`: String (the logic behind the decision)

Chapter 5

Results

5.1 Experimental Setup and Dataset: The UCR Archive

As any other methodology, the foundation of anomaly detection algorithms also lies in the integrity of its evaluation data. However, as noted by Wu and Keogh [20], most of the recent research in the field are potentially an illusion caused by as the title puts it flawed benchmarks. This paper introduces three critical issues in current famous benchmark datasets such as NAB and Yahoo:

- **Triviality:** Many anomalies are so simple and don't require complex AI.
- **Unrealistic Labels:** Ground truth labels are often misplaced or subjective.
- **Data Leakage:** Some models cheat by seeing patterns in the test set that exist in the training set.

To avoid such illusions and ensure a correct evaluation of the pipeline, this research utilizes the UCR Time Series Anomaly Archive. This dataset was specifically developed to correct the flaws identified in earlier datasets, and provides 250 diverse, non-trivial sequences from different datasets where anomalies are clearly defined and contextually complex.

So to overcome the three issues existing in previous benchmarks this dataset is designed as follows to ensure its integrity:

- **Single but Expert-Verified Anomaly:** To eliminate the Anomaly Density problem (where a model might look good just by marking a huge block of data as bad), each sequence in the archive contains exactly one carefully created anomaly. This forces the pipeline to be precise; the model either identifies the correct timestamp or it fails.

- **Meaningful Training/Test Split:** To prevent data leakage, each file is split into a training segment (Anomaly-Free segment) and a test segment (Segment that contains a single anomaly). This data structure allows the model to learn benign yet rare events before running on the entire dataset.

- **The One-Liner Baseline Requirement:** Wu and Keogh introduced ,as they call it, one-liner methods (simple statistical tests such as moving mean or differencing). For an anomaly to be considered non-trivial, it must be difficult for simple scripts to detect it. This ensures VLMs are required to perform this task; if a one-line script could find the anomaly, multimodal reasoning would be too heavy and thus unnecessary to use.

- **Run-to-Failure Prevention:** Many older datasets placed anomalies at the very end of the dataset. This allowed biased models to perform well through simply flagging the final data points. The archive ensures the occurrence of anomalies at varied and unpredictable times.

As the figure 5.1 demonstrates, the anomalies in UCR dataset are carefully engineered and carry meaningful and critical information. However, this is only an example of the dataset and all data inside this archive comes with its own unique and specific type of anomaly. While this anomaly is about amplitude change, the next data could include an anomaly in the ordering of peaks, the frequencies, etc.

The training, testing and ground truth are all present in the name of the files. For example in the file name "`UCR_Anomaly_tiltAPB1_100000_114283_114350.txt`", the 100000 represents the point at which the training segment finishes, which means this point onwards represents the inference data. Furthermore, the ground truth period is [114283,114350].

5.2 Evaluation Metric: f1 score and f1 Affiliation

Anomaly detection models are divided into algorithms that pinpoint the exact location of an anomaly and those that provide an interval for the anomalous segment. While the well-known F1-score is used for the former, the latter is utilized to better evaluate interval-based models.

In this scenario, since the anomaly detection task is based on interval-based ground truth and the VLMs are approximating the interval start and end points based on visuals rather than direct numerical data, F1-Affiliation is currently the most suitable criterion for this evaluation. The algorithm behind F1-Affiliation is provided in the appendix.

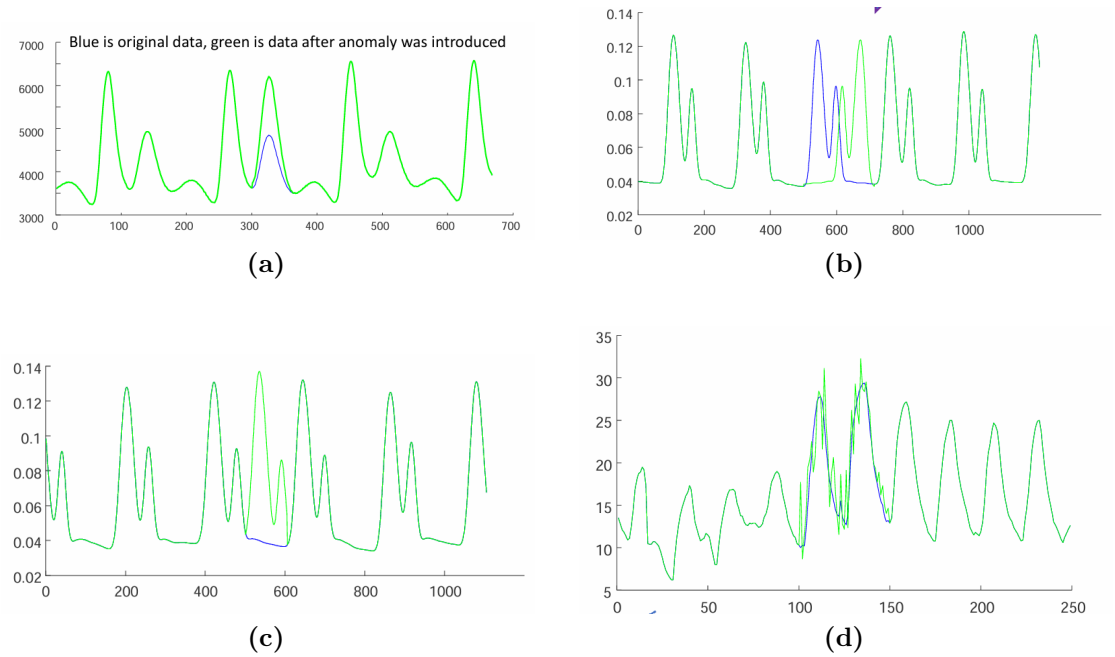


Figure 5.1: Varied types of anomalies in the UCR dataset [20].

5.3 Comparative Performance of Vision-Language Models

To determine the optimal Foundation Model for the semantic interpretation phase of the pipeline, an evaluation of multiple state-of-the-art Vision-Language Models (VLMs) was conducted. The comparison was measured by Precision, Recall, F1 Score, and Accuracy parameters that are demonstrated in Table 5.1.

Table 5.1: Model Performance Leaderboard

Model Name	Precision	Recall	F1 Score	Accuracy
gemini-3-pro-preview	**0.6667**	0.6957	**0.6809**	**0.8864**
gemini-3-pro-preview-highlighted	0.5758	**0.8261**	0.6786	0.8636
gpt-5	0.6000	0.7407	0.6630	0.8320
gpt-5-highlighted	0.5000	0.8000	0.6154	0.7727
gemini-3-flash-preview	0.3636	0.8000	0.5000	0.6364
qwen-vl-max	0.3333	0.4000	0.3636	0.4615

As demonstrated by the empirical results, the choice of the foundational model

and its visual input format significantly impacts the system's diagnostic capabilities. The analysis of these results yields several critical insights regarding multimodal topological reasoning.

The Optimal Configuration: Gemini-3-Pro

The `google_gemini-3-pro-preview` model that was evaluated on the plots that were not highlighted and maintained the same color across all signals achieved the highest overall F1 Score (0.6809) and Accuracy (0.8864). Most notably, this model achieved the highest Precision (0.6667) across all tested variants. By forcing the model to evaluate the raw signal geometry without any artificial visual annotation, the pipeline successfully minimized false positives. This model showed the best performance among alternative approaches and was used as our pipeline's VLM during later comparisons.

GPT-5 The unannotated `openai_gpt-5` model also demonstrated good topological reasoning capabilities and achieved the second highest F1 Score among all alternatives (0.6630) with an Accuracy of 0.8320. Operating on the images that were not highlighted, GPT-5 achieved a robust Recall of 0.7407 while maintaining a solid Precision of 0.6000. While it fell slightly short of `gemini-3-pro-preview`'s precision capabilities, GPT-5's strong performance independently verifies that larger state-of-the-art VLMs possess the reasoning capabilities required to parse signal distortions when provided with mathematical plots.

Attention Hijacking and the Highlighted Variant

The table provides direct empirical evidence supporting the visual ablation study. When the exact same model was evaluated using the highlighted anomaly intervals (`google_gemini-3-pro-preview_highlighted`), its Recall spiked to 0.8261, but its Precision suffered a sharp decline to 0.5758. These results perfectly illustrate the phenomenon of *attention hijacking*. The presence of the red highlight made the model overly eager to classify the window as an anomaly which could be the consequence of red being the color mostly associated with abnormality. This configuration led to capturing more true positives (high recall) but simultaneously triggering a massive increase in false alarms (low precision). Consequently, the overall F1 Score degraded, proving that artificial visual guidance undermines the model's strict geometric rigor.

Performance of Alternative and Lightweight Models

The evaluation also highlights the limitations of alternative and lightweight architectures for anomaly detection:

GPT-5 (Highlighted). While `openai_gpt-5` achieved a strong Recall of 0.8000, its Precision dropped to 0.5000, resulting in an F1 Score of 0.6154. Like the highlighted Gemini variant, it suffered from severe attention hijacking, generating an unacceptable rate of false positives.

Gemini-3-Flash. The lightweight `google_gemini-3-flash-preview` model demonstrated the classic limitations of smaller parameter counts in complex reasoning tasks. It managed to achieve high Recall (0.8000) but in contrast, it showed a significant drop in Precision to a low of 0.3636. This indicates that this model struggled heavily to differentiate between benign noise and true anomalies.

Qwen-VL-Max. The `qwen-vl-max` architecture failed to generalize to this specific time-series diagnostic task and produced the lowest performance across all metrics (F1 Score: 0.3636, Recall: 0.4000). These results suggest that some multimodal architectures and especially the smaller ones are not suited for zero-shot time series anomaly detection task.

Evaluation Scope and Cost Constraints

The UCR Time Series Anomaly Archive consists of 250 distinct datasets. However, due to the substantial API inference costs and computational overhead associated with querying massive, proprietary Vision-Language Models (such as Gemini-3-Pro and GPT-5), executing the full benchmark across the entire archive was financially prohibitive. Consequently, this comparative evaluation was conducted on a representative subset of 20 datasets.

Crucially, while this subset was restricted in volume, it proved entirely sufficient to separate and evaluate the distinct reasoning capabilities of the tested models. The 20 datasets were strictly sampled to encompass a diverse range of temporal morphologies and structural complexities. This curated variety provided more than enough empirical variance to clearly isolate the highly capable architectures (e.g., Gemini-3-Pro) from those that fundamentally lacked topological reasoning (e.g., Qwen-VL-Max and Gemini-3-Flash). Thus, while not an exhaustive benchmark of the complete UCR suite, this subset provided the exact resolution necessary to establish a definitive VLM leaderboard and validate the core architectural hypotheses of this thesis.

5.4 Anomaly Detection Performance

The proposed Neuro-Symbolic framework ("Our Model") was benchmarked against four model categories:

- A brute-force visual baseline (VLM4TS),
- A purely mathematical baseline (DAMP [18]),
- Two fully trained state-of-the-art models (Chronos [5] and TimeRCD [21]).

Table 5.2: Anomaly Detection Performance Metrics

Model	Precision	Recall	F1 Score
VLM4TS	0.505	**0.754**	0.604
DAMP	**0.836**	0.519	0.6400
Chronos	−−	−−	0.740
TimeRCD	−−	−−	**0.846**
Our Model	0.623	0.661	0.642

5.4.1 Superiority over the Direct Baseline (VLM4TS)

Our model demonstrates a clear improvement over the direct VLM4TS baseline in terms of balanced F1 score (0.6416 vs. 0.6049).

Precision Advantage. The most significant improvement is observed in precision (0.6232 vs. 0.5050). While VLM4TS achieves higher recall (0.7541), it suffers from a significantly higher false-positive rate.

Validation of the 1+3 Strategy. This performance gap validates the effectiveness of the structured topological JSON-formatted prompt and the 1+3 comparative layout. The constrained prompt architecture successfully reduced over-flagging of noisy yet benign windows.

5.4.2 Bridging the Gap to the Mathematical Baseline

Our model slightly outperforms the purely symbolic DAMP baseline (0.6416 vs. 0.6400). This result is particularly significant: it demonstrates that the addition of the neural reasoning layer did not degrade the algorithmic rigor established by the Matrix Profile discovery phase. Instead, the VLM preserved mathematical strengths while adding semantic interpretability.

5.4.3 Performance Gap with Fully Trained Models

TimeRCD (0.8463) and Chronos (0.7400) maintain superior performance, with TimeRCD exceeding Chronos by approximately 10.6%. However, both models are explicitly trained for anomaly detection tasks and benefit from dataset-specific parameter optimization.

In contrast, the proposed framework operates in a strictly zero-shot setting and it does not require dataset-specific weight updates. Achieving competitive performance despite such constraint highlights the strength of the hybrid design and its potential for deployment in scenarios where labeled data is limited or unavailable.

5.5 Computational Efficiency and Running Time Analysis

While anomaly detection accuracy (F1 score) serves as the primary metric, computational weight that is needed to perform such inferences is equally important for industrial deployment. Given that Foundational Vision-Language Models (VLMs) are computationally intensive and costly to query while the requirements of the industry requires an approach that is lightweight, a method to reduce such computational overhead of such powerful tools is needed. Accordingly, a design objective of the proposed pipeline was to minimize unnecessary VLM inferences by leveraging the Matrix Profile as an algorithmic gatekeeper.

Table 5.3 reports the total running time required to process the full UCR Archive inference set per pipeline including ours and that of the state-of-the-art.

Table 5.3: Total Running Time Comparison

Model	Total Running Time (Minutes)
VLM4TS	1,920
DAMP	255.6
Our Model	492.6

5.5.1 The Efficiency of the Algorithmic Gatekeeper

As shown in Table 5.2, the proposed framework completed inference in 492.6 minutes (255.6 minutes for DAMP algorithm and 237 minutes for VLM inference) and significantly outperformed the VLM4TS brute-force approach, which required 1,920 minutes. This corresponds to a reduction of 1,428 minutes.

This improvement validates the effectiveness of Phase I. the Through usage of the Matrix Profile (MP) for the main discord discovery tool, the mathematical layer identified the Top-K candidate windows prior to VLM's evaluation. This filtration stage ensured computational efficiency by preventing the costly transmission of clearly normal windows to the VLM.

In contrast, VLM4TS processes a significantly larger number of windows through its transformer architecture without prior mathematical filtering. The proposed pipeline strictly reserves the VLM's high-cost reasoning for mathematically verified candidates, thereby improving computational allocation without sacrificing detection quality.

5.5.2 The Cost of Semantic Interpretation vs. Mathematical Baselines

When compared to the purely mathematical DAMP algorithm, which completes the dataset scan in approximately 255.6 minutes, the 1,920-minute running time of VLM4TS reflects the significant computational weight associated with heavy ViT inferences. However, by utilizing the Matrix Profile as an algorithmic gatekeeper, our proposed hybrid pipeline completes the evaluation in just 492.6 minutes. This significantly reduces the computational burden of pure transformer approaches by approximately 75%. This approach successfully achieves a scalable middle ground between fast but silent algorithms and expressive but expensive visual models.

However, this disparity must be interpreted under consideration of functional output differences as well. DAMP provides mathematically precise distance profiles but lacks semantic interpretability. It identifies discords numerically without verifying structural topology.

The additional runtime of the proposed model reflects the cost of transforming a mathematical flag into a structured, JSON-formatted topological evaluation. In high-stakes industrial environments, this overhead is justified by the system's ability to:

- Verify directional order consistency,
- Preserve relative amplitude hierarchies,
- Validate inter-extrema distance topology.

Rather than producing a black-box numerical score, the system delivers a constrained and interpretable structural result.

5.6 Summary of Experimental Findings

The experimental evaluation on the UCR Archive validates the central hypothesis of this thesis: integrating the mathematical rigor of the Matrix Profile with the multimodal reasoning capabilities of a VLM results in a robust, efficient, and explainable anomaly detection framework.

Performance. The hybrid architecture outperforms the direct multimodal baseline (VLM4TS) in balanced Affiliation F1 score while remaining competitive with specialized models under zero-shot constraints.

Efficiency. The algorithmic gatekeeping mechanism significantly reduces inference time compared to brute-force approaches.

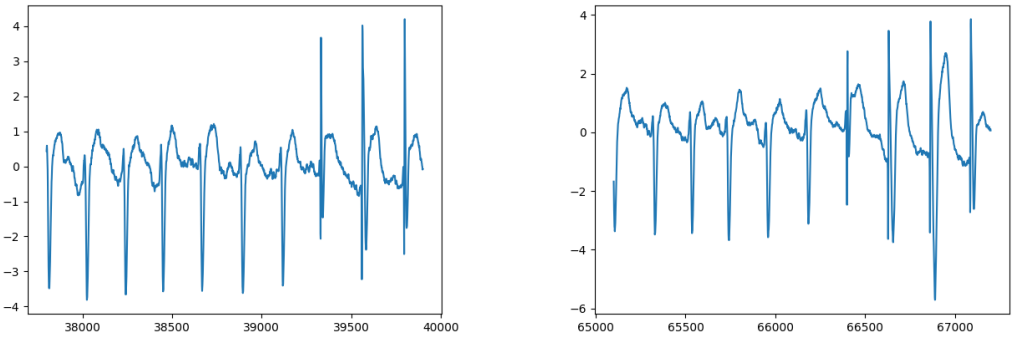
Interpretability. Through the use of strict topological constraints and JSON output formatting, the framework avoids superficial pattern matching and ensures structural understanding of signal geometry.

Overall, these results demonstrate that our neuro-symbolic integration provides a practical approach that is both scalable and trustworthy in time-series anomaly detection tasks.

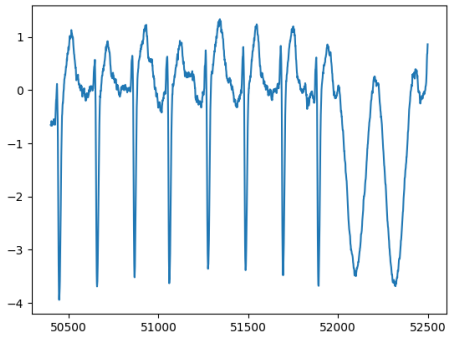
5.7 Diagnostic Variant Output

As mentioned earlier, the main benefit of usage of VLMs is their ability to explain their reasoning behind each process.

To showcase the VLM's potential in transforming future anomaly detection algorithms into an explainable system, sample outputs of the VLM are demonstrated as below with respect to order of images in Figure 5.2:



(a) **(b)**



(c)

Figure 5.2: Sample inference images used to demonstrate the Diagnostic variant's capabilities.

Image A:

- **Is anomalous:** False
- **Confidence:** High
- **Reasoning:** The signal maintains the directional transition sequence and periodic spacing of the reference set. The high-amplitude vertical excursions in later cycles are topologically identical to the valid morphological variants in Reference #4, preserving the permissible structural hierarchy.

Image B:

- **Is anomalous:** False

- **Confidence:** High
- **Reasoning:** The directional transition sequence (rise–peak–drop–recovery) aligns with the valid variant morphology in Reference #4. The relative amplitude hierarchy and inter-extrema distance topology are preserved, matching the high-amplitude spike structures previously established as benign.

Image C:

- **Is anomalous:** True
- **Confidence:** High
- **Reasoning:** The interval displays a loss of secondary extrema structure, replacing complex multi-peak cycles with simplified, broad sinusoidal waves. This violates the established relative amplitude hierarchy and expands the inter-extrema distances of the primary transitions.

As demonstrated, not only above models made the right decision, they also managed to provide a logical explanation of the motif topology and how each images aligns or differs from it while maintaining the ability of distinguishing the difference between anomalous segments and previously observed noise in the data through cross referencing the inference images with the specific reference that included the similar noise.

Chapter 6

Conclusion

6.1 Research Summary and Thesis Recapitulation

The main achievement of this thesis is its success in integration of a mathematical algorithm with a heavy reasoning model. Our results demonstrate that the Matrix Profile and Vision-Language Large Models (VLMs) are not always supposed to work in parallel; they could instead compensate for each other's weaknesses while amplifying their respective strengths.

6.1.1 Matrix Profile as the Computational Accelerator

A significant issue with practical applicaiton of VLMs to time-series data is their *inference bottleneck*. While they are strong models, the process of attending to every sliding window of a long-duration stream by a multi-billion parameter model is computationally and financially infeasible.

The Strength. The Matrix Profile provides an exact yet light solution for discovering discords.

The Synergy. The VLM effectively audited the Matrix Profile's output. By leveraging the reference set and its dominant frequencies, the model performed strict topological reasoning. While the pipeline constrained the output to a deterministic JSON format to prevent hallucinations, this architecture forced the VLM to internally evaluate structural hierarchies, transforming detection from a mathematical black-box into a rules-based semantic diagnostic.

6.1.2 VLM as the Semantic Interpreter

While the Matrix Profile is mathematically precise, it is unable to provide the users with more information about how the flagged subsequence is different from the motifs nor can it benefit from domain knowledge on the specific field on which its operating.

The Strength. VLMs possess deep semantic capabilities and is able to reason about the physical and topological structure of signals through visual-textual mapping while also benefiting from domain knowledge provided by the users in case it is crucial.

The Synergy. The VLM effectively audited the Matrix Profile's output. By leveraging the reference set and its the dominant frequencies, the model produced a Chain-of-Thought (CoT) justification for each flagged event. This transformed detection from a black-box generated numerical score into an explainable diagnostic which enables mathematical algorithms to benefit from words in showcasing their results.

6.2 Eradicating the Black Box: From Detection to Explanation

The described synergy addresses a core challenges in industrial AI and could be beneficial in many aspects:

Transparency through Structural Rules By forcing the VLM to evaluate structural sequence, relative amplitude topology, and inter-extrema distance, we open the black box typically associated with deep neural systems. The resulting hybrid pipeline achieves zero-shot performance (F1: 0.6416) comparable to specialized models, while providing a clear, rule-based topological audit trail.

Trustworthy Autonomy. In critical industrial applications, the explanation of an anomaly is often as valuable as its detection. By centering the AI's attention using the Matrix Profile, the system can allocate computational resources to high-level reasoning. The outcome is a monitoring framework that is both technically effective and operationally transparent.

6.3 Limitations of the Proposed Framework

Although the proposed neuro-symbolic architecture successfully bridges mathematical precision and semantic interpretability, several structural limitations remain.

6.3.1 The Matrix Profile Recall Ceiling

The most defining limitation arises from the Algorithmic Layer. While the Matrix Profile functions effectively as a computational filter, it included the true anomalous window within the Top-K candidates in only 198 out of 250 UCR Archive datasets.

Algorithmic Blind Spots. This establishes a strict theoretical recall ceiling of 79.20%. Although the Matrix Profile excels at detecting extreme structural discords, it may fail to surface highly contextual or subtle temporal anomalies that do not significantly perturb the Euclidean distance landscape.

Sequential Dependency. Because the architecture operates sequentially, the Neural Layer is entirely dependent on the output of the Algorithmic Layer. Any anomaly not surfaced during Phase I is never evaluated by the VLM. Consequently, the neural component cannot compensate for upstream filtering omissions.

6.3.2 API Latency and Closed-Source Dependency

Despite improving runtime relative to the VLM4TS baseline (20,221 minutes vs. 25,753 minutes), the computational overhead remains substantial compared to purely algorithmic methods such as DAMP (1 minute).

Real-Time Constraints. This latency precludes deployment in ultra-high-frequency real-time environments (e.g., microsecond-level trading systems or high-speed hardware diagnostics).

External Infrastructure Dependency. The framework relies on closed-source foundational models accessed via external APIs (e.g., Gemini 1.5 Pro infrastructure). This introduces variability related to network latency, token quotas, and usage costs—factors absent in deterministic, locally executed algorithms.

6.3.3 Rigidity of Topological Constraints

To eliminate hallucination and avoid superficial reasoning, the prompt architecture enforces strict topological evaluation with JSON-only output.

Intentional Feature Exclusion. By explicitly instructing the model to ignore stochastic noise, jitter, and absolute amplitude scaling, the system intentionally excludes certain anomaly classes from consideration. If a failure manifests purely as high-frequency sensor noise without altering structural topology, the current configuration may classify it as benign.

6.4 Directions for Future Work

The identified limitations provide a clear roadmap for improving the current pipeline in the current task or its usage in downstream tasks.

6.4.1 Upgrading the Algorithmic layer

The current pipeline in order to reduce the computational overhead utilizes matrix profile as a filtering algorithm that leads to a recall ceiling (Our setup enforces 79.20% but it's a hyper parameter and could be adjusted).

6.4.2 Usage of smaller VLMs

In a more optimal setup, the VLM used should be smaller and executable on local hardware. However, as seen in our experiments smaller VLMs are to be improved for such task. Despite the current limitations VLMs are constantly improving and possibly in a near future the smaller VLMs would demonstrate significantly better performances and thus reach the standard needed for employment in such scenario.

6.4.3 Extension to Multivariate Time Series

The present evaluation relies on the UCR Archive, which contains univariate time series. Industrial systems typically involve multiple interacting sensors.

Future research should investigate whether the 1+3 comparative layout can be generalized to multivariate inputs, enabling the VLM to reason about cross-sensor spatial correlations and coupled structural failures.

Considering the nature of this pipeline and its usage of images lead to convenient deployment and strong performance in a multi-signal scenario and thus is my next area of work.

6.5 Industrial Implications: Trustworthy Autonomy in Industry 4.0

The neuro-symbolic synergy developed in this thesis addresses a central challenge in industrial AI: deploying autonomous monitoring systems in critical environments with both reliability and interpretability.

6.5.1 Actionable Diagnostics over Binary Alarms

In critical applications such as manufacturing plants, power grids, or aerospace telemetry the explanation of an anomaly is often as valuable as its detection. Traditional black-box models may correctly flag failures, but they provide little insight into the underlying physical cause, forcing human engineers into prolonged troubleshooting and costly inspections.

By producing strict topological evaluations such as identifying collapsed distances or altered amplitude hierarchies this framework supplies diagnostics. Engineers receive interpretable explanations that could lead to more efficient reactions to anomaly and an overall reduced downtime.

6.5.2 Resource Allocation and Scalability

The industrial deployment of larger VLMs is often constrained by computational expense. Matrix Profile as a mathematical gatekeeper helps the system to direct its computational resources toward the most suspicious candidates.

This approach ensures below features:

- Heavy neural reasoning is reserved for meaningful events,
- System throughput remains scalable across large networks,
- Operational transparency is maintained without compromising performance.

6.5.3 Implications for Predictive Maintenance

The final proposed framework demonstrates an exploration in the area of trustworthy AI through combination of VLMs and mathematical algorithms. Integrating mathematical strength with multimodal semantic reasoning helps the system establish a paradigm for predictive maintenance in Industry 4.0, where:

- Anomalies are not only detected but explained,
- Computational efficiency is balanced with semantic interpretability,

- Human operators can confidently rely on AI insights in critical infrastructure settings.

Overall, this approach provides a robust, financially scalable, and operationally transparent model for autonomous industrial monitoring.

6.6 Final Conclusion

This thesis demonstrates that the future of time-series anomaly detection could not be only in increasingly opaque models, but also in hybrid pipelines that merge mathematical precision with neural reasoning. The Matrix Profile provides the possibility for scalability and efficiency that is required for practical use of a pipeline, while the VLM contributes to this matter with semantic interpretation, domain knowledge and explainability. Together, they form a robust, explainable, and zero-shot framework that establishes a new paradigm for industrial monitoring.

Appendix A

JSON Prompt Architecture and Persona Definition

To ensure absolute reproducibility and to constrain the Vision-Language Model (VLM) from generating unstructured natural language, the following strict prompt architecture was utilized. The prompt is divided into three functional blocks: the System Persona, the Reference (Baseline) Context, and the Inference Evaluation.

The prompt explicitly forces the model to evaluate the data purely on relative topological structures while suppressing evaluations based on absolute amplitude, scale, or rendering artifacts. The output is strictly constrained to a JSON schema for programmatic integration.

A.1 System Prompt (Persona and Constraints)

The following instructions initialized the model's behavior and defined the strict bounds of topological evaluation:

```
1 You are a Signal Morphology Analysis Expert specialized in
     detecting TOPOLOGICAL ANOMALIES. Your task is to classify
      whether a Test signal deviates from Reference signals
     based strictly on STRUCTURAL SEQUENCE, RELATIVE AMPLITUDE
      TOPOLOGY, AND RELATIVE INTER-EXTREMA DISTANCE within
     each cycle.
2
3 A valid signal is defined by:
4 - The ORDER of directional transitions (upward vs downward
     movements)
```

```
- The RELATIVE AMPLITUDE hierarchy between successive extrema
- The RELATIVE DISTANCE hierarchy between successive extrema
- The CONSISTENCY of these patterns across cycles

Primary Detection Criteria:
1. Directional Order: The sequence of rises and falls must match reference patterns.
2. Relative Amplitude Steps: Peaks and valleys must preserve their relative magnitude ordering (e.g., primary peak > secondary peak > minor oscillation).
3. Transition Topology: The structural progression between extrema must remain consistent.
4. Inter-Extrema Distance Topology: The relative spacing between successive extrema (peak-to-valley, peak-to-peak, valley-to-valley) must preserve reference hierarchy (e.g., dominant spacing > secondary spacing > minor spacing).

Additional Topological Constraints:
- Distances are evaluated strictly in a RELATIVE sense (ranking only).
- Absolute scale, absolute duration, or sampling density must NOT be considered.
- Compression or expansion is allowed ONLY if distance hierarchy is preserved.

Ignore entirely:
- Absolute amplitude values
- Baseline offsets
- Absolute time duration
- Sampling rate differences
- Noise or jitter
- Visual smoothness or shape

You must NOT explain, narrate, or interpret visually. Your decision must be derived from structural consistency only.
```

A.2 User Prompt: Reference Standards (Few-Shot Baseline)

To establish the comparative baseline, the model was provided with the Golden Motif and Boundary Motifs extracted from the training data, accompanied by the

following conditioning text:

```
## REFERENCE STANDARDS (BENIGN SIGNALS)
These images define VALID TOPOLOGICAL STRUCTURES.

Learn the following only:
- Directional transition sequences (up/down ordering)
- Relative amplitude hierarchy between extrema
- Relative distance hierarchy between extrema
- Stable structural patterns that repeat per cycle

Do NOT learn visual appearance, smoothness, or exact shapes.

[Image inputs injected here, labeled iteratively as "Ref #1:
    TYPICAL MORPHOLOGY" and "Ref #X: VALID VARIANT
   MORPHOLOGY", accompanied by their dominant frequency
   variables.]
```

A.3 User Prompt: Inference and JSON Schema Enforcement

The final block of the prompt injects the unseen candidate discord and enforces the deterministic JSON output structure:

```
## TEST SIGNAL TOPOLOGICAL AUDIT
**Interval:** [Injected Suspected Interval]
**Dominant Signals:** [Injected Dominant Frequency]

Compare the TEST interval against the learned Reference
   TOPOLOGICAL STRUCTURES.

**Evaluation Rules (Strict):**
1. Extract the ordered sequence of directional movements (UP
   /DOWN).
2. Compare sequence length and ordering against reference
   cycles.
3. Extract successive extrema and rank their relative
   amplitudes.
4. Extract distances between successive extrema.
5. Rank inter-extrema distances (longest to shortest) within
    each cycle.
```

```
6. Compare amplitude and distance hierarchies against reference cycles.
7. Detect any:
   - Missing transitions
   - Extra transitions
   - Reordered peaks or valleys
   - Altered amplitude dominance
   - Collapsed or expanded spacing that alters distance hierarchy
   - Reordered distance dominance

**Decision Policy:**
- ANOMALOUS = Any violation of directional order, amplitude hierarchy, or inter-extrema distance hierarchy.
- BENIGN = All topological hierarchies are preserved.

**IMPORTANT CONSTRAINTS:**
- Do NOT describe waveform appearance.
- Do NOT explain signal behavior.
- Do NOT justify the decision.
- Do NOT reference timing, scale, or absolute values.

**You MUST respond with ONLY a valid JSON object.**
No markdown. No additional text.

**Output Format:**
{
    "is_anomalous": true or false,
    "confidence": "high" or "medium" or "low"
}

[Inference Image Input Injected Here]
```

A.3.1 User prompt for Diagnostic Vairant

To modify the VLM to also include reasoning behind the decisions the User Prompt was altered as the following:

```
test_prompt = f"""
## TEST SIGNAL TOPOLOGICAL AUDIT
**Interval:** {suspected_interval}
**Dominant Signals:** {dominant_signals[-1]}

```

```
Compare the TEST interval against the learned Reference
   TOPOLOGICAL STRUCTURES.

**Evaluation Rules (Strict):**
1. Extract the ordered sequence of directional movements (UP
   /DOWN).
2. Compare sequence length and ordering against reference
   cycles.
3. Extract successive extrema and rank their relative
   amplitudes.
4. Extract distances between successive extrema.
5. Rank inter-extrema distances (longest     shortest)
   within each cycle.
6. Compare amplitude and distance hierarchies against
   reference cycles.
7. Detect any:
- Missing transitions
- Extra transitions
- Reordered peaks or valleys
- Altered amplitude dominance
- Collapsed or expanded spacing that alters distance
   hierarchy
- Reordered distance dominance

**Decision Policy:**
- ANOMALOUS = Any violation of directional order, amplitude
   hierarchy, or inter-extrema distance hierarchy.
- BENIGN = All topological hierarchies are preserved.

**IMPORTANT CONSTRAINTS:**
- Do NOT describe waveform appearance.
- Explain the decision only using structural/topological
   evidence.
- Do NOT reference timing, scale, or absolute values.

**You MUST respond with ONLY a valid JSON object.**
No markdown. No additional text.

**Reasoning Field Requirement:**
- Include a `reasoning` field with 18 to 40 words.
- Keep it concise and evidence-based.
- Mention only directional order, relative amplitude
   hierarchy, and/or relative inter-extrema distance
   hierarchy.
```

```
- Do NOT include visual style, absolute values, or timing statements.

**Output Format:**
{
    "is_anomalous": true or false,
    "confidence": "high" or "medium" or "low",
    "reasoning": "18-40 words explaining the topological evidence for the decision"
}
"""
```

Appendix B

Affiliation Evaluation Implementation

Traditional point-wise evaluation metrics strictly penalize predictions that do not perfectly align with ground truth timestamps, which is often misleading in continuous Time-Series Anomaly Detection (TSAD). To provide a fair, mathematically rigorous evaluation of the proposed Neuro-Symbolic framework, the Affiliation F1 metric was utilized.

To ensure standardized and reproducible computation of these window-based metrics, the evaluation was conducted using the `dtaianomaly` library. The following Python implementation details the data pipeline used to parse the UCR Archive filenames, extract the deterministic JSON verdicts generated by the Vision-Language Model, convert the temporal bounds into binary arrays, and compute the final Affiliation Precision, Recall, and F1 scores.

```
import os
import json
import numpy as np
from dtaianomaly.evaluation import AffiliationPrecision,
    AffiliationRecall

def intervals_to_binary(intervals, length):
    """Convert list of (start, end) intervals to binary
    array."""
    binary = np.zeros(length, dtype=int)
    for start, end in intervals:
        binary[start:end+1] = 1
    return binary

```

```
def process_results_dtai(folder_path, prediction_key="vit_intervals", tolerance=50, max_files=250):
    """
    Args:
        folder_path: Path to results directory
        prediction_key: Key for predictions in old format
        tolerance: Distance tolerance for "close enough" predictions (default: 50 points)
        max_files: Maximum number of files to process (default: 100)
    """
    all_precisions = []
    all_recalls = []
    file_count = 0

    # Initialize metric objects
    prec_metric = AffiliationPrecision()
    rec_metric = AffiliationRecall()

    # Walk through all subdirectories to find JSON files
    for root, dirs, files in os.walk(folder_path):
        for filename in files:
            if not filename.endswith(".json"):
                continue

            # Stop after processing max_files
            if file_count >= max_files:
                break

            file_path = os.path.join(root, filename)

            # 1. Extract Ground Truth from filename
            # Handle format: XXX_UCR_Anomaly_..._{start}_{end}_google_gemini...json
            parts = filename.replace(".json", "").split("_")

            # Find the ground truth indices (before model name like 'google')
            try:
                # Find the index where model name starts (e.g., 'google')
                model_idx = next(i for i, p in enumerate(parts) if 'google' in p.lower() or 'gemini' in p.lower())
                # Ground truth is 2 positions before the model name
```

```
                gt_start, gt_end = int(parts[model_idx - 2])
    , int(parts[model_idx - 1])
            except (ValueError, StopIteration, IndexError):
                # Fallback to old format
                gt_start, gt_end = int(parts[-2]), int(parts
    [-1])

            # 2. Load Predictions and get series length
            with open(file_path, 'r') as f:
                data = json.load(f)

            # Handle two possible formats
            if isinstance(data, list) and len(data) > 0 and
    isinstance(data[0], list):
                # New format: [[json_string, [start, end]],
    ...]
                preds = []
                for item in data:
                    if len(item) == 2:
                        try:
                            # Handle both string and dict
    formats
                            if isinstance(item[0], str):
                                result_json = json.loads(
    item[0])
                            elif isinstance(item[0], dict):
                                result_json = item[0]
                            else:
                                continue

                            # More flexible boolean checking
                            is_anom = result_json.get('
    is_anomalous')
                            if isinstance(is_anom, bool):
                                is_anomalous = is_anom
                            elif isinstance(is_anom, str):
                                is_anomalous = is_anom.lower
    () in ['true', '1', 'yes']
                            elif isinstance(is_anom, int):
                                is_anomalous = is_anom == 1
                            else:
                                is_anomalous = False

                            if is_anomalous:
```

```
                                preds.append({'start': item
[1][0], 'end': item[1][1]})
                        except (json.JSONDecodeError,
KeyError, IndexError, TypeError):
                            continue
                series_length = gt_end + 100
            else:
                # Old format: dict with prediction_key
                series_length = data.get('series_length',
gt_end + 100)
                preds = data.get(prediction_key, [])

            if preds:
                max_pred_end = max(int(i['end']) for i in
preds)
                series_length = max(series_length,
max_pred_end + 1)
            series_length = max(series_length, gt_end + 1)

            # 3. Convert to binary arrays
            y_true = intervals_to_binary([(gt_start, gt_end)
], series_length)

            if not preds:
                y_pred = np.zeros(series_length, dtype=int)
            else:
                pred_intervals = [(int(i['start']), int(i['
end'])) for i in preds]
                y_pred = intervals_to_binary(pred_intervals,
 series_length)

            # 4. Compute metrics
            if np.sum(y_pred) == 0:
                p = 0.0
                r = 0.0
            else:
                p = prec_metric.compute(y_true, y_pred)
                r = rec_metric.compute(y_true, y_pred)
                if np.isnan(p):
                    p = 0.0
                if np.isnan(r):
                    r = 0.0

            all_precisions.append(p)
            all_recalls.append(r)
```

```
123             file_count += 1
124
125         if file_count >= max_files:
126             break
127
128     avg_p = np.mean(all_precisions) if all_precisions else 0
129     avg_r = np.mean(all_recalls) if all_recalls else 0
130     f1 = 2 * (avg_p * avg_r) / (avg_p + avg_r) if (avg_p +
    avg_r) > 0 else 0
131
132     return avg_p, avg_r, f1
133
134 p, r, f1 = process_results_dtai("results_new",
    prediction_key="vit_intervals")
135 print(f"Precision: {p:.4f}\nRecall: {r:.4f}\nF1 Score: {f1
    :.4f}")
```

Bibliography

[1] George E. P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day, 1970 (cit. on p. 7).

[2] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Anh Dau, Diego Silva, Abdullah Mueen, and Eamonn Keogh. «Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets». In: Dec. 2016, pp. 1317–1322. DOI: 10.1109/ICDM.2016.0179 (cit. on p. 8).

[3] Sepp Hochreiter and Jürgen Schmidhuber. «Long Short-Term Memory». In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cit. on p. 9).

[4] Colin Lea, Michael D. Flynn, René Vidal, Austin Reiter, and Gregory D. Hager. «Temporal Convolutional Networks for Action Segmentation and Detection». In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1003–1012. DOI: 10.1109/CVPR.2017.113 (cit. on p. 9).

[5] Abdul Fatir Ansari et al. *Chronos: Learning the Language of Time Series*. 2024. arXiv: 2403.07815 [cs.LG]. URL: https://arxiv.org/abs/2403.07815 (cit. on pp. 12, 50).

[6] Zihao Zhou and Rose Yu. *Can LLMs Understand Time Series Anomalies?* 2025. arXiv: 2410.05440 [cs.LG]. URL: https://arxiv.org/abs/2410.05440 (cit. on p. 12).

[7] Zelin He, Sarah Alnegheimish, and Matthew Reimherr. *Harnessing Vision-Language Models for Time Series Anomaly Detection*. 2025. arXiv: 2506.06836 [cs.CV]. URL: https://arxiv.org/abs/2506.06836 (cit. on pp. 13, 14).

[8] Jiaxin Zhuang, Leon Yan, Zhenwei Zhang, Ruiqi Wang, Jiawei Zhang, and Yuantao Gu. *See it, Think it, Sorted: Large Multimodal Models are Few-shot Time Series Anomaly Analyzers*. 2024. arXiv: 2411.02465 [cs.LG]. URL: https://arxiv.org/abs/2411.02465 (cit. on p. 14).

[9] Sean Law. «STUMPY: A Powerful and Scalable Python Library for Time Series Data Mining». In: *Journal of Open Source Software* 4 (July 2019), p. 1504. DOI: 10.21105/joss.01504 (cit. on p. 19).

[10] Youssef Laarouchi, Mohamed Kaaniche, Vincent Nicomette, Ivan Studnia, and Eric Alata. «A language-based intrusion detection approach for automotive embedded networks». In: *International Journal of Embedded Systems* 10 (Jan. 2018), p. 1. DOI: 10.1504/IJES.2018.10010488 (cit. on p. 19).

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762 (cit. on pp. 20–22).

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805 (cit. on pp. 22, 23).

[13] OpenAI et al. *GPT-4 Technical Report*. 2024. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774 (cit. on p. 23).

[14] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». In: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929 (cit. on p. 25).

[15] Alec Radford et al. «Learning transferable visual models from natural language supervision». In: *International Conference on Machine Learning*. PMLR. 2021, pp. 8748–8763. URL: https://arxiv.org/abs/2103.00020 (cit. on p. 26).

[16] Gemini Team et al. *Gemini: A Family of Highly Capable Multimodal Models*. 2025. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805 (cit. on p. 27).

[17] Yu Zhang et al. *Google USM: Scaling Automatic Speech Recognition Beyond 100 Languages*. 2023. arXiv: 2303.01037 [cs.CL]. URL: https://arxiv.org/abs/2303.01037 (cit. on p. 28).

[18] Yue Lu, Renjie Wu, Abdullah Mueen, Maria Zuluaga, and Eamonn Keogh. «Matrix Profile XXIV: Scaling Time Series Anomaly Detection to Trillions of Datapoints and Ultra-fast Arriving Data Streams». In: Aug. 2022, pp. 1173–1182. DOI: 10.1145/3534678.3539271 (cit. on pp. 32, 34, 50).

[19] Alexa R. Tartaglini, Satchel Grant, Daniel Wurgaft, Christopher Potts, and Judith E. Fan. *Diagnosing Bottlenecks in Data Visualization Understanding by Vision-Language Models*. 2025. arXiv: 2510.21740 [cs.CV]. URL: https://arxiv.org/abs/2510.21740 (cit. on p. 39).

[20] Renjie Wu and Eamonn J. Keogh. «Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress». In: *CoRR* abs/2009.13807 (2020). arXiv: 2009.13807. URL: https://arxiv.org/abs/2009.13807 (cit. on pp. 45, 47).

[21] Tian Lan, Hao Duong Le, Jinbo Li, Wenjun He, Meng Wang, Chenghao Liu, and Chen Zhang. *Towards Foundation Models for Zero-Shot Time Series Anomaly Detection: Leveraging Synthetic Data and Relative Context Discrepancy*. 2025. arXiv: 2509.21190 [cs.LG]. URL: https://arxiv.org/abs/2509.21190 (cit. on p. 50).