ENHANCING TIME SERIES ANALYSIS THROUGH EFFICIENT MOTIF DISCOVERY AND ITS APPLICATION TO FORECASTING

by

Ayça Güler

B.S., Industrial Engineering, Dokuz Eylül University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computational Science and Engineering

Boğaziçi University

2025

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my supervisor, Prof. Bertan Badur, for his valuable guidance, support, and encouragement throughout this research. His expertise and insights have been crucial in the successful completion of this thesis.

I am also grateful to my family for their unwavering love and support. Their encouragement and belief in me have been my greatest source of strength during this journey.

To everyone who has contributed to this work, directly or indirectly, thank you for your support and inspiration.

Finally, I wish to acknowledge the use of GPT for assistance with grammar checks and minor stylistic improvements throughout several chapters of this thesis. All intellectual and conceptual contributions are my own.

ABSTRACT

ENHANCING TIME SERIES ANALYSIS THROUGH EFFICIENT MOTIF DISCOVERY AND ITS APPLICATION TO FORECASTING

Time series motif discovery is a powerful technique for identifying recurring patterns in sequential data, offering valuable insights into diverse applications such as finance, healthcare, and climate science. This thesis presents an innovative framework for motif discovery in time series data, addressing key challenges in identifying recurring patterns of varying lengths and instances. The proposed methodology improves sensitivity and accuracy of motif detection by using Binary Integer Programming (BIP) for optimal motif selection and employing a heuristic algorithm for efficient pattern discovery, while integrating a dynamic feedback mechanism to refine parameters. Additionally, the study demonstrates how the discovered motifs can be applied to forecasting tasks by integrating them into a Transformer model pipeline augmented with motif embeddings, assessing whether motif-augmented predictions outperform simpler baselines. Extensive experiments on real-world financial datasets reveal that while the identified motifs can effectively capture recurring structures, incorporating them into advanced forecasting models does not necessarily lead to improved predictive performance. In particular, Transformers with or without motif embeddings underperform relative to basic econometric techniques. These findings underscore the complexity of leveraging motif information within deep learning frameworks for time series forecasting, suggesting that although motif discovery is valuable for understanding temporal patterns, more targeted modeling strategies or alternative machine learning approaches may be necessary to realize its full potential in predictive tasks.

ÖZET

ZAMAN SERİSİ ANALİZİNİ ETKİLİ MOTİF KEŞFİ İLE GELİŞTİRME VE MOTİF KEŞFİNİN TAHMİN UYGULAMALARINA ENTEGRE EDİLMESİ

Zaman serisi motif keşfi, ardışık verilerde tekrar eden desenleri belirlemek için güçlü bir yöntem olup, finans, sağlık hizmetleri ve iklim bilimi gibi çeşitli alanlarda değerli içgörüler sunmaktadır. Bu tez, zaman serisi verilerinde motif keşfi için yenilikçi bir çerçeve sunarak, farklı uzunluklardaki ve örneklerdeki tekrar eden desenlerin tespitinde karşılaşılan temel zorlukları ele almaktadır. Önerilen metodoloji, motif tespitinde duyarlılığı ve doğruluğu artırmak için optimal motif seçimi amacıyla İkili Tamsayılı Programlama (BIP) kullanmakta, verimli desen keşfi için sezgisel bir algoritma uygulamakta ve parametreleri iyileştirmek üzere dinamik bir geri bildirim mekanizması entegre etmektedir. Ayrıca, çalışma, keşfedilen motiflerin, motif gömme özellikleriyle güçlendirilmiş bir Transformer modeline entegre edilerek tahmin görevlerine nasıl uygulanabileceğini gösterir ve motif destekli tahminlerin daha basit temel modelleri aşıp aşmadığını değerlendirir. Gerçek finansal veri kümeleri üzerinde yapılan kapsamlı deneyler, belirlenen motiflerin tekrar eden yapıları etkili bir şekilde yakalayabildiğini ortaya koysa da, bu bilgilerin gelişmiş tahmin modellerine entegre edilmesinin tahmin performansını yeterli olarak artırmadığını göstermiştir. Özellikle, motif gömme içersin veya içermesin, Transformer modelleri temel ekonometrik tekniklere kıyasla daha düşük performans sergilemektedir. Bu bulgular, derin öğrenme çerçeveleri içinde motif bilgisinin kullanılmasının karmaşıklığını vurgulamakta ve motif keşfinin zaman serilerindeki desenleri anlamada değerli olmasına rağmen, potansiyelini tam olarak ortaya koymak için daha hedeflenmiş modelleme stratejileri veya alternatif makine öğrenmesi yaklaşımlarının gerekliliğini işaret etmektedir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . iv
ÖZET . . . v
LIST OF FIGURES . . . viii
LIST OF TABLES . . . ix
LIST OF SYMBOLS . . . x
LIST OF ACRONYMS/ABBREVIATIONS . . . xi
1. INTRODUCTION . . . 1
1.1. Context and Significance of Motif Discovery . . . 1
1.2. Objectives and Scope of the Study . . . 1
2. BACKGROUND . . . 4
2.1. Time Series Data Mining . . . 4
2.2. Motif Discovery . . . 5
2.2.1. Similarity Measures . . . 6
2.2.2. Matrix Profile . . . 8
2.3. Forecasting Time Series . . . 13
2.3.1. Sliding Window Mechanism . . . 16
2.3.2. Motif Embeddings . . . 16
2.4. Transformer Model Architecture . . . 19
2.4.1. Attention Mechanism . . . 20
2.4.2. Hyperparameter Tuning in Transformers . . . 22
2.5. Financial Markets . . . 23
3. METHODOLOGY . . . 26
3.1. Range of Motif Length . . . 29
3.2. Extracting the Motifs . . . 31
3.2.1. Binary Integer Programming . . . 32
3.2.2. Heuristic Approach . . . 34
3.3. Forecasting . . . 35

3.3.1. Data Preparation 36
3.3.1.1. Train-Validation-Test Split 36
3.3.1.2. Scaling of the Original Time Series 37
3.3.2. Evaluation and Comparison Metrics 37
4. EXPERIMENTS AND RESULTS 39
4.1. Experimental Data 39
4.2. Motif Discovery 40
4.3. Motif Discovery Results 43
4.3.1. Influence of Minimum Motif Length m_{start} on Candidate Motifs Percentage 43
4.3.2. Entropy as a Measure of Motif Diversity 44
4.3.3. Comparison of BIP and Heuristic Methods 44
4.3.4. Effect of Instance Count (i) on Parameter Tuning 44
4.4. Forecasting 49
4.5. Forecasting Results 50
4.6. Discussion 52
5. CONCLUSION 54
5.1. Further Research Directions 54
REFERENCES 56
APPENDIX A: CODE REPOSITORY 63
APPENDIX B: COPYRIGHT AND LICENSE INFORMATION 64

LIST OF FIGURES

Figure 2.1. Transformer Architecture. Adapted from [43]. 19

Figure 4.1. Highlighted Instances of BIP Motifs in Time Series 46

Figure 4.2. Highlighted Instances of Heuristic Motifs in Time Series 46

Figure 4.3. Z-Normalized Alligned Instances of BIP Motif 0 47

Figure 4.4. Z-Normalized Alligned Instances of BIP Motif 1 47

Figure 4.5. Z-Normalized Alligned Instances of BIP Motif 2 48

Figure 4.6. Z-Normalized Alligned Instances of Heuristic Motif 0 48

Figure 4.7. Z-Normalized Alligned Instances of Heuristic Motif 1 49

LIST OF TABLES

Table 4.1. Description of Input Data. 40

Table 4.2. Experimental Results for Motif Discovery on ISCTR 40

Table 4.3. Experimental Results for Motif Discovery on GAU/TRY 42

Table 4.4. Experimental Results for Motif Discovery on LCO 42

Table 4.5. Experimental Results for Motif Discovery on USD/TRY 42

Table 4.6. Experimental Results for Motif Discovery on XU100 43

Table 4.7. Parameter Selection for Different Datasets 49

Table 4.8. Comparison of Methods on ISCTR and GAU/TRY Datasets . . . 51

LIST OF SYMBOLS

b_1, b_2	Bias terms in the model
d_k	Dimensionality of key vectors
H	Entropy, measuring uncertainty in motif distributions
$\mathbf{K}$	Key matrix in attention mechanism
$\mathbf{Q}$	Query matrix in attention mechanism
t	Time step
$\mathbf{V}$	Value matrix in attention mechanism
$\mathbf{W}$	Weight matrix for linear transformations
x	Input variable or time series data point
$\mathbf{x}$	Input vector for the model
y	Target variable or predicted value
$\mathbf{y}$	Output vector or target variable
$z(x)$	Z-normalized value of a data point x
μ_X, μ_Y	Means of time series X and Y
σ_X, σ_Y	Standard deviations of time series X and Y

LIST OF ACRONYMS/ABBREVIATIONS

AMH	Adaptive Market Hypothesis
ANN	Artifical Neural Network
BIP	Binary Integer Programming
CNN	Convolutional Neural Network
DTW	Dynamic Time Warping
EM	Expectation Maximization
EMH	Efficient Market Hypothesis
GDP	Gross Domestic Product
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MEME	Multiple EM for Motif Elicitation
RF	Random Forest
RMSE	Root Mean Square Error
SVR	Support Vector Regression
TFT	Temporal Fusion Transformer

1. INTRODUCTION

Motifs are recurring patterns or sequences that hold significant biological, computational, or structural relevance. These patterns play a critical role in understanding complex systems across a wide range of domains, including bioinformatics, time series analysis, text mining, signal processing, and more. In time series data, motif discovery is particularly important for identifying recurring patterns that provide insights into temporal dynamics, such as trends, anomalies, and cycles. These insights are critical for applications in financial modeling, healthcare monitoring, and climate studies. Despite the progress made in motif discovery methods, the field still faces numerous challenges, including computational efficiency, scalability, sensitivity, and accuracy in detecting meaningful motifs of varying lengths and instances in sequential datasets.

1.1. Context and Significance of Motif Discovery

Motif discovery has far-reaching implications in various domains. In time series analysis, motifs help uncover repeating patterns that can inform predictive modeling, anomaly detection, and decision-making in critical systems. For example, identifying motifs in healthcare time series data can lead to early detection of medical conditions, while in financial data, motifs can indicate market trends or risks. This capability extends to other fields like climate science, where motifs reveal recurring weather patterns, and engineering, where motifs can identify structural vibrations or failures. The ability to identify motifs of different lengths and their instances is crucial for understanding the underlying complexities of time series data, making motif discovery a cornerstone of modern analytical methodologies.

1.2. Objectives and Scope of the Study

This thesis seeks to address the limitations of current motif discovery methods by introducing a novel approach focused specifically on time series data. Our research

aims to:

- Improve the sensitivity and accuracy of detecting motifs of varying lengths and multiple instances in complex time series datasets.
- Develop computational techniques that ensure scalability and efficiency for large-scale time series data.
- Provide a flexible framework applicable to diverse time series applications, including healthcare, finance, and climate analysis. The scope of this work includes theoretical advancements, algorithm development, and practical validation on time series datasets representing real-world scenarios.

In this thesis, the primary focus is on enhancing the efficiency and accuracy of time series motif discovery. Specifically, we develop an advanced computational framework that systematically identifies motifs of varying lengths and multiple occurrences within complex time series data. Our approach combines a robust pattern identification method with optimization techniques, including Binary Integer Programming (BIP) and a heuristic algorithm, to ensure meaningful, non-trivial motifs are selected while preventing overlap. Additionally, a dynamic feedback mechanism is integrated to iteratively refine the search parameters, enhancing the motif discovery process. Although our methodology is demonstrated using financial datasets, the proposed framework's flexibility allows it to be easily adapted to other domains requiring deep insights into recurrent temporal patterns.

The remainder of this thesis is organized as follows:

Chapter 2 : Background This chapter provides a comprehensive review of foundational concepts and related work in motif discovery, with a particular emphasis on time series analysis. It highlights existing methods, their limitations, and the motivation for this research.

Chapter 3 : Methodology This chapter details the proposed framework, including the algorithms, techniques, and theoretical developments in this study for time series motif discovery.

Chapter 4 : Experiments and Results This chapter describes the experimental setup and datasets. It also presents and analyzes the results obtained from applying the proposed method to real-world time series datasets.

Chapter 5 : Conclusion The final chapter summarizes the contributions of this research, discusses its implications, and provides suggestions for future research directions.

2. BACKGROUND

This chapter lays the foundation for the rest of the thesis by providing key background concepts and techniques relevant to the research. First, we introduce Time Series Data Mining and explain its importance in discovering patterns and insights within sequential data. Building on this, we delve into Motif Discovery, highlighting its role in detecting recurrent, potentially significant patterns that may hold predictive or explanatory power. Next, we summarize various approaches to Forecasting on Time Series, reviewing both classical techniques and more recent methods in the literature. Following this, we discuss the Transformer Model Architecture, explaining how attention mechanisms and parallel processing can address the unique challenges of sequence modeling. Finally, we delve into the Financial Markets as the chosen application domain, discussing the basics of market behavior and exploring key theories such as Efficient Market Hypothesis (EMH).

2.1. Time Series Data Mining

Time series information is increasingly dominating the global data landscape. For example, [1] demonstrated this trend by analyzing 4,000 graphics from 15 international newspapers between 1974 and 1989, finding that over 75% of them were time series data. Time series data are widely utilized across diverse domains, including finance and economics, healthcare and medicine, energy and utilities, environmental science, transportation, manufacturing, retail and e-commerce, technology and communications, sports, seismology, astronomy, music, and agriculture. A time series, which consists of observations collected at sequential time intervals, plays a pivotal role in understanding temporal dynamics in these fields. The value of time series data is further amplified by advancements in data collection technologies, which allow for granular and high-frequency observations. With exponentially growing datasets and increasing applicability, interest in time series data mining has exploded in recent decades.

However, the unique characteristics of time series data—particularly their potentially massive size, high dimensionality, and random noise—often render traditional machine learning approaches ineffective [2]. This challenge has driven the development of specialized techniques such as indexing, motif discovery, classification, clustering, segmentation, summarization, and anomaly detection, which employ dimensionality reduction and specialized similarity measures to cope with the scale, complexity, and noise of real-world temporal datasets [3].

The analysis of time series data provides insights into underlying patterns, trends, and relationships, enabling informed decision-making, forecasting, and anomaly detection. Its importance derives from its ability to capture the temporal dependencies inherent in real-world phenomena, making it indispensable for applications such as predicting future values, identifying anomalies, and understanding periodic behaviors.

2.2. Motif Discovery

Motif discovery is focused on to find out repeating patterns in a sequential dataset. Extracting motifs plays a crucial role in this research. We focused on discovering various potential motifs, each with multiple instancing patterns. However, before delving into these aspects, it is essential to present the foundational definitions. The following definitions are derived from the work of [4], which provides a comprehensive exploration of time series distance profiles and their computation.

Definition 1: Time Series A time series T of length n is an ordered sequence of real numbers measured at equally spaced time intervals:

$$T = (T[1], T[2], \ldots, T[n]). \tag{2.1}$$

Definition 2: Subsequence A subsequence $T_{i,L}$ is a continuous segment of length L extracted from a time series T, starting at position i:

$$T_{i,L} = (T[i], T[i+1], \ldots, T[i+L-1]) \tag{2.2}$$

where $1 \leq i \leq n - L + 1$.

Definition 3: Query A query Q is a time series of length m that is searched within a longer time series T, where $n \gg m$.

Definition 4: The distance profile D of a query Q against a time series T is a sequence of distances computed between Q and all subsequences $T_{i,m}$ of T:

$$D[i] = \text{dist}(T_{i,m}, Q) \tag{2.3}$$

where dist is a distance function, commonly the z-normalized Euclidean distance.

Definition 5: Correlation Profile The relationship between z-normalized Euclidean distance 2.7 and Pearson's correlation 2.8 enables simple transformations from one another [4]. The correlation between two sequences x and y can be expressed as:

$$\text{corr}(x, y) = 1 - \frac{\text{dist}(x, y)^2}{2m} \tag{2.4}$$

where $\text{dist}(x, y)$ represents the z-normalized Euclidean distance between the sequences, and m is the query length.

2.2.1. Similarity Measures

To uncover patterns in a time series, shape similarity is crucial. By comparing the overall structure of subsequences rather than individual points, shape similarity enables the detection of recurring motifs, anomalies, and meaningful trends, forming the foundation for effective time series analysis.

(i) LockStep Measures refer to methods that compare two sequences directly, element by element, while maintaining their original alignment

- Euclidean distance is a fundamental measure of similarity or dissimilarity between two sequences. It calculates the straight-line distance between two points in a multi-dimensional space, making it a widely used metric in time series analysis. Given two time series $X = \{x_1, x_2, \ldots, x_n\}$ and

$Y = \{y_1, y_2, \ldots, y_n\}$ of equal length n, the Euclidean distance $d(X, Y)$ is defined as:

$$d(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \tag{2.5}$$

This formula computes the root of the sum of squared differences between corresponding elements in the sequences. Euclidean distance is straightforward to calculate, making it efficient for comparing time series with the same length and aligned temporal structures. Despite its simplicity, Euclidean distance serves as a baseline metric in time series analysis, providing a foundation for more advanced similarity measures.

- Z-Normalized Euclidean Distance is an extension of the standard Euclidean distance, specifically designed to compare time series while accounting for differences in scale and offset. This normalization ensures that sequences with varying amplitudes and baseline levels can be meaningfully compared. Given two time series $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ of equal length n, the z-normalization process involves standardizing each sequence to have a mean of zero and a standard deviation of one:

$$z(x_i) = \frac{x_i - \mu_X}{\sigma_X}, \quad z(y_i) = \frac{y_i - \mu_Y}{\sigma_Y}, \tag{2.6}$$

  where:

  - μ_X and μ_Y are the means of X and Y, respectively.
  - σ_X and σ_Y are the standard deviations of X and Y, respectively.

  The Z-Normalized Euclidean Distance $d_z(X, Y)$ is then computed as:

$$d_z(X, Y) = \sqrt{\sum_{i=1}^{n} \left(z(x_i) - z(y_i)\right)^2}. \tag{2.7}$$

  This approach eliminates the influence of scale and offset, making it ideal for comparing time series with similar shapes but differing magnitudes or baseline levels.

- Correlation Coefficient is a statistical measure that quantifies the degree and direction of a linear relationship between two variables or time series. It is widely used in time series analysis to assess how strongly two sequences are

related, independent of their scales.

Given two time series $X = \{x_1, x_2, \ldots, x_n\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$, the Pearson correlation coefficient [5] $r(X, Y)$ is defined as:

$$r(X,Y) = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2} \cdot \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}, \tag{2.8}$$

where:

- $\bar{x}$ and $\bar{y}$ are the mean values of X and Y, respectively.
- The numerator $\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$ represents the covariance between X and Y.
- The denominator normalizes the covariance by the standard deviations of X and Y, ensuring that $r(X, Y)$ lies between -1 and 1.

(ii) Elastic Measures are advanced methods used in time series similarity analysis to account for temporal misalignments, variable lengths, and non-linear distortions between sequences. Unlike lockstep measures, elastic measures allow flexible alignment by stretching or compressing time series as needed to find the best match. Dynamic Time Warping (DTW) is a robust algorithm for measuring similarity between time series by aligning them non-linearly along the temporal axis [6]. The Move-Split-Merge (MSM) metric is a time series similarity measure that transforms one series into another using three fundamental operations—Move, Split, and Merge—ensuring robustness to temporal distortions, metric properties, and translation invariance [7].

2.2.2. Matrix Profile

There are several studies on motif discovery. A research group led by Keogh et al. has published an extensive series of papers centered around time series analysis using the Matrix Profile framework.

The Matrix Profile is a data structure that annotates a time series with the nearest neighbor distances and indices of its subsequences. It is a vector that stores the best pairwise z-normalized Euclidean distances between all subsequences of a time

series. The Matrix Profile P of a time series is a vector where each element represents the minimum distance from a subsequence $T_{i,m}$ to all other subsequences:

$$P[i] = \min_{j \neq i} D_i[j]. \tag{2.9}$$

The corresponding index $I[i]$ in the Matrix Profile Index indicates the position of the nearest neighbor:

$$I[i] = \arg\min_{j \neq i} D_i[j]. \tag{2.10}$$

Here, D_i is the distance profile, as explained at the beginning of this section, which represents the z-normalized Euclidean distances between the subsequence $T_{i,m}$ and all subsequences of the time series. The computation of P and I involves:

(i) Calculating the distance profile D_i for each subsequence $T_{i,m}$ using the Fast Fourier Transform (FFT) for efficient convolution.
(ii) Updating $P[i]$ and $I[i]$ by comparing $D_i[j]$ values across subsequences.

The Matrix Profile can be computed for:

- Self-Join: When the time series is compared to itself, trivial matches (subsequences overlapping with themselves) are excluded by setting an exclusion zone around the current index i.
- All-Pairs Join**:** When the time series T_A and T_B are distinct, no exclusion zone is needed, and the profile is denoted P_{AB}.

The Matrix Profile provides a unified solution for several time series tasks:

- Motif Discovery: The lowest values in P indicate recurring patterns.
- Discord Discovery: The highest value in P corresponds to an anomaly.
- Semantic Segmentation: Abrupt changes in the Matrix Profile Index reveal regime boundaries.

Starting with the foundational work in Matrix Profile I [8], which introduced the all-pairs similarity join and the Matrix Profile, they have since expanded the framework with more than 30 papers addressing various challenges and applications. Several studies have focused specifically on motif discovery, including discovering meaningful patterns [8], motifs in multidimensional time series [9], and those with variable lengths [10]. In [8], they introduce STAMP algorithm, aims to find the Matrix Profile of a time series. With this approach, the nearest neighbor (pair) can be identified for every starting index of a fixed-length subsequence. [9] proposed an extension of STAMP as mSTAMP for multivariate motif discovery. The VALMOD algorithm [10] is designed for variable-length motif discovery in time series data, efficiently identifying recurring patterns (motifs) across a user-defined range of lengths. Unlike traditional approaches, which require a fixed motif length, VALMOD discovers motifs and motif sets without excessive computational costs. Key Steps of VALMOD Algorithm are:

(i) Matrix Profile Computation:
- Compute the Matrix Profile for the shortest length
- Iteratively extend to longer lengths.
- Use precomputed results for shorter lengths to accelerate distance calculations for longer lengths.

(ii) Lower Bound Distance Technique:
- Introduces a lower bounding method to approximate distances for longer motifs, pruning subsequences that cannot improve the current best-so-far distance.
- This reduces redundant calculations, making the algorithm scalable.

(iii) Finding Motif Pairs:
- Identify the Top-K most similar subsequences (motif pairs) for each length based on z-normalized Euclidean distance.
- Motif pairs are ranked using a length-normalized distance to ensure fair comparison across lengths.

(iv) Extending Motif Pairs to Motif Sets:

- Expand each motif pair into a motif set by including all subsequences within a radius r, $r = D \cdot \text{dist}(T_{i,\ell}, T_{j,\ell})$. D is a user-defined multiplier, $T_{i,\ell}$ and $T_{j,\ell}$ are the subsequences forming the motif pair, and dist is the z-normalized Euclidean distance between them.
- Apply an exclusion zone to avoid trivial matches and overlapping subsequences.

(v) Iterative Processing:

- Process motif pairs iteratively, starting with the pair that has the smallest distance.
- Mark subsequences in processed motif sets as covered to prevent redundancy in subsequent sets.

Others targeted computational advancements, such as leveraging GPU acceleration [11, 12], and scaling analysis to massive datasets [13]. A notable contribution includes enabling exact motif discovery under Dynamic Time Warping (DTW) [14]

The framework has also been extended to anomaly detection, where methods such as HOT SAX [15] and MERLIN [16] addressed discord detection, while multidimensional anomaly detection was achieved in [17]. Additionally, they tackled challenges such as handling weakly labeled data for predictive modeling [18], discovering motifs with missing data [19], and real-time segmentation using FLOSS [20].

Despite the significant advancements in time series motif discovery, several challenges remain. Focusing on motif discovery with variable lengths, the state-of-the-art model proposed in [10] has several limitations:

(i) Dependency on Radius Factor D : The user-defined parameter D (radius multiplier) heavily influences the size of motif sets. A high D value may include irrelevant subsequences, while a low D value can exclude meaningful variations of motifs. The absence of an adaptive mechanism for D makes the algorithm less

intuitive for non-expert users.

(ii) Potential Loss of Motifs Due to Iterative Covering: The iterative process of processing motif pairs marks subsequences in one motif set as covered to prevent redundancy. While this ensures non-overlapping motif sets, it can inadvertently prune potential motifs in datasets with many highly similar subsequences.

(iii) Dependency on User-Defined K : The K value, which determines the number of Top-K motif pairs, directly impacts the quality of discovered motifs. A large K may result in the inclusion of less similar candidate motif pairs, while a small K might miss less prominent but meaningful motifs.

On the other hand, some of the studies focused on categorical but sequential data. The most widely developed studies are in the bioinformatics domain due to its data structure. [21] proposed Gemoda (Generic Motif Discovery Algorithm) for identifying motifs in sequential data, including DNA, proteins. It operates in three phases: (1) Comparison, where sequences are divided into overlapping windows and compared using customizable similarity metrics; (2) Clustering, which groups similar windows into elementary motifs; and (3) Convolution, which stitches these motifs into maximal motifs. [22] presents a statistical method for identifying common motifs, such as protein-binding sites, in unaligned DNA sequences using the Expectation Maximization (EM) algorithm. The algorithm is a statistical method used to find maximum likelihood estimates of parameters in models with latent (hidden) variables [22]. In this context, the latent variable is the unknown position of the motif within each sequence.

- E-Step (Expectation): Computes the expected value of the latent variables given the current model parameters.
- M-Step (Maximization): Updates the model parameters to maximize the likelihood based on the expected values computed in the E-step.

The algorithm iterates between these steps until the changes in the parameters become negligible, indicating convergence.

[23] introduces the MEME (Multiple EM for Motif Elicitation) algorithm. This algorithm is designed for discovering motifs in sequential data such as DNA and proteins, using an extended version of the Expectation Maximization (EM) algorithm.

Some studies such as [24] focused on representation of time series as categorical sequential dataset. By this application, they aim both reduction of size and benefit from the sequential methods. [25] HIME (Hierarchical based Motif Enumeration) utilized the SAX representation to find variable length motifs in large scale time series. It uses symbolic approximation (SAX) to discretize data, builds an induction graph to manage subsequences, and employs recursive enumeration to iteratively extend short patterns into longer motifs. While HIME demonstrates remarkable scalability and flexibility in variable-length motif discovery, it has limitations, including its approximation-based approach, sensitivity to parameter settings, and challenges with highly noisy or sparse data. Furthermore, it is limited to univariate time series, which restricts its direct application to multivariate datasets

2.3. Forecasting Time Series

This section provides an overview of time series forecasting. It begins with classical approaches (such as ARIMA), which rely on statistical assumptions to capture linear trends and seasonality. It then highlights more advanced methodologies, including machine learning algorithms and attention-based models, which address challenges like non-linear dependencies. Next, the discussion turns to the sliding window mechanism, which reorganizes raw sequential data into supervised learning inputs. Finally, the discussion broadens to the concept of embeddings, exploring various embedding types before focusing on how motif embeddings fit into the sliding window approach by carefully structuring motif embeddings to avoid lookahead bias. Together, these topics offer insight into established practices and emerging trends in the field.

Time series forecasting has become a critical area of research across diverse domains, including finance, healthcare, energy, and retail. The challenge lies in modeling

temporal dependencies and leveraging historical data to predict future values accurately. Several studies have explored this topic, focusing on different aspects such as traditional methods, novel methodologies, practical applications, benchmarking of techniques and hybrid approaches.

Traditional methods, including Autoregressive (AR), Moving Average (MA), and Autoregressive Integrated Moving Average (ARIMA) models, are among the most widely used techniques for forecasting time series data [26]. These methods rely on statistical principles and assume a linear relationship between the past values (lags) of the series and future observations, making them particularly effective for datasets exhibiting clear trends or seasonality. The simplicity and interpretability of traditional methods are key advantages, as they allow researchers and practitioners to gain insight into the underlying patterns and relationships within the data.

Although traditional methods can effectively model linear patterns and seasonality, they often struggle to capture the complex non-linear dynamics present in many real-world datasets; consequently, more advanced methodologies such as machine learning algorithms and attention-based models have emerged to address these limitations.

[27] introduces the Temporal Fusion Transformer (TFT), a novel deep learning architecture designed for multi-horizon time series forecasting. Multi-horizon forecasting involves predicting variables over multiple future time steps, which is crucial in applications like retail, healthcare, and economics. The model uses attention mechanisms and recurrent layers to capture short and long term dependencies in the data.

In their work, [28] introduced a novel attention mechanism tailored for multivariate time series forecasting, called Temporal Pattern Attention (TPA). Unlike traditional attention mechanisms that focus on specific time steps, TPA selects relevant variables and leverages their temporal patterns across time. By integrating Convolutional Neural Networks (CNNs) for temporal pattern detection and attention for variable selection, the model effectively captures complex dependencies in multivariate

datasets. This approach is particularly impactful in scenarios involving periodic and non-linear patterns in time series data.

[29] explores the use of machine learning algorithms, particularly Random Forests (RF) and Support Vector Machines (SVM), to predict stock market prices based on historical statistical data. It focuses on developing an effective model that can accurately predict the direction and magnitude of stock prices, providing valuable insights for investors and financial analysts.

The study by [30] explores the application of advanced deep learning architectures for short-term stock price prediction. It focuses on addressing the limitations of traditional forecasting methods by leveraging Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Convolutional Neural Networks (CNNs) using a sliding window approach. The study aims to identify the most suitable model for capturing the non-linear dynamics and sudden variations characteristic of financial time series data.

[31] proposes a two-stage hybrid machine learning approach for predicting stock market indices. The proposed methodology integrates Support Vector Regression (SVR), Artificial Neural Networks (ANN), and Random Forest (RF) in a two-stage fusion framework. The study evaluates the performance of these hybrid models on the CNX Nifty and S&P BSE Sensex indices from the Indian stock market using 10 years of historical data.

The diversity of approaches in time series forecasting reflects the complexity and interdisciplinary nature of the field. Novel methodologies like TFT and TPA push the boundaries of model architecture design, while benchmarking studies and practical applications provide valuable insights into their real-world utility. Traditional methods, such as ARIMA and Exponential Smoothing [32] , remain indispensable for their explanatory power and as a baseline for comparison with more advanced approaches. Hybrid approaches, such as the two-stage fusion framework, demonstrate how combin-

ing existing methods can address specific challenges and improve forecasting accuracy. These studies collectively contribute to the advancement of the field, enabling more informed decision-making in various domains.

2.3.1. Sliding Window Mechanism

A fundamental challenge in time series forecasting lies in transforming raw sequential data into a form suitable for supervised learning. The sliding window (or rolling window) approach addresses this by systematically partitioning the time series into fixed-length overlapping segments, called "windows", each serving as an input instance for the predictive model.

Suppose we have a univariate time series $\{x_t\}_{t=1}^{T}$. In the sliding window method, we choose a window size W that specifies how many past observations are used to predict future values. Specifically, for each time step t (such that $t - W \geq 1$), we gather the observations $\{x_{t-W}, \ldots, x_{t-1}\}$ as the model's input features, and then use the subsequent data point x_t (or multiple subsequent data points, for multi-step forecasting) as the target output.

- Window Definition: Fix a window length W.
- Input-Output Pair Construction: For each time t, the input (feature vector) is $\big(x_{t-W}, x_{t-W+1}, \ldots, x_{t-1}\big)$ and the output (label) is typically x_t for one-step-ahead prediction, or $(x_t, x_{t+1}, \ldots)$ for multi-step forecasting. We used one-step-ahead prediction in this study.
- Rolling Forward: Shift the window by one or more time steps and repeat the process, producing a dataset of many $(\text{input}, \text{output})$ pairs.

2.3.2. Motif Embeddings

The concept of embeddings plays a pivotal role in representing complex and unstructured data, such as text, images, and audio, in a format amenable to machine

learning models. Embeddings convert data into dense numerical vectors in a multidimensional space, where similar data points are clustered together based on their semantic or contextual relationships. This transformation not only facilitates the efficient handling of high-dimensional data but also enhances the interpretability and performance of machine learning algorithms [33]

The primary aim of embeddings is to represent unstructured data as dense vectors while preserving semantic relationships between data points. This representation is crucial for various tasks, such as similarity search, classification, and clustering, where understanding the intrinsic relationships between data points is fundamental. Moreover, embeddings serve as a bridge between raw data and advanced analytical processes, enabling the integration of different data modalities in a unified space.

Embeddings can be categorized based on the type of data they represent. Common types include [33]:

- Word Embeddings: Represent individual words as vectors, capturing their semantic and syntactic relationships. Techniques such as Word2Vec [34] and GloVe [35] are widely used for this purpose.
- Sentence Embeddings: Extend word embeddings to represent entire sentences as vectors, preserving contextual meaning. Methods such as Universal Sentence Encoder (USE) [36]and SkipThought [37] are notable examples.
- Document Embeddings: Represent entire documents as vectors, enabling tasks like document similarity and classification. Techniques like Doc2Vec [38] are widely used.
- Image Embeddings: Map image features to vectors using convolutional neural networks (CNNs) or pre-trained models like VGG [39] and ResNet [40]. These embeddings are crucial for tasks like image retrieval and classification.
- User and Product Embeddings: Capture user behaviors, preferences, and product features, commonly used in recommendation systems and personalized marketing.
- Multimodal Embeddings: Integrate multiple data modalities (e.g., text and im-

ages) into a unified vector space. Models like CLIP [41] exemplify this approach.

Each type of embedding is tailored to preserve the most relevant characteristics of the data it represents, making embeddings versatile tools across diverse applications.

In this study, the concept of embeddings is extended to time series data in the form of motif embeddings. These embeddings represent recurring patterns (motifs) discovered in the time series, capturing their temporal and structural characteristics. By integrating motif embeddings into the forecasting pipeline, this work aims to leverage their ability to derive essential data patterns, thereby enhancing the accuracy and robustness of predictions.

Motifs were identified within the time series data, and their occurrences were encoded into the input matrix. Each motif was represented as a dense, continuous vector, generated via a motif-specific embedding layer. In this study, we adopted a column-based integration approach, where each motif was treated as an independent feature dimension within the input data.

At each time step, the input vector consisted of:

(i) The original univariate time series value
(ii) Additional columns, each corresponding to a motif, containing either the motif embedding vector (for time steps where the motif occurred) or zeros (for time steps without the motif).

A sliding window of a fixed lookback length is moved across the time series. For each window, the input features consist of the original univariate data (in the first column) augmented by additional columns for each motif. To prevent future data leakage, motif embeddings are incorporated into each window according to strict temporal rules. If a motif occurrence lies completely within the window (i.e., the motif starts at or after the window's beginning and ends at or before the window's end), its

embedding is marked over the corresponding time steps. If a motif started before the window but ends inside it, only the overlapping portion (from the window's start to the motif's end) is filled with the embedding. If a motif begins within the window but would extend beyond the window's end, no embedding is added for that occurrence. This desicion is critical because it prevents using information from future time steps. By this construction, we eliminate the risk of lookahead bias.

The Pytorch 'Embedding' layer [42] was utilized to map categorical indices to dense vector representations, facilitating the integration of motif-based features into the Transformer model. This layer provides an efficient mechanism for learning embeddings during model training.

2.4. Transformer Model Architecture

The Transformer model, introduced by [43], represents a significant departure from traditional sequence architectures like recurrent neural networks (RNNs) and convolutional neural networks (CNNs). It eliminates the reliance on recurrence and convolutions by leveraging a self-attention mechanism to model dependencies across sequences. This approach significantly improves computational efficiency and parallelization capabilities.

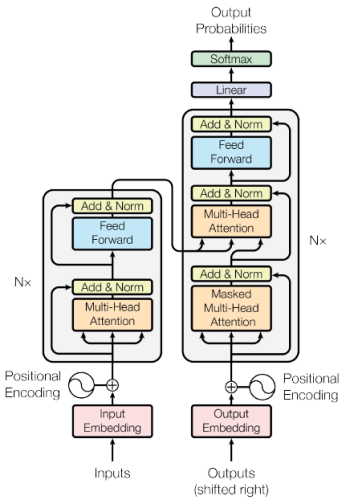

Figure 2.1. Transformer Architecture. Adapted from [43].

The architecture in Figure 2.1 includes an encoder-decoder structure, where both components are composed of stacked layers of self-attention and feed-forward networks. Self-attention, implemented using scaled dot-product attention, calculates dependencies between tokens regardless of their positional distance, enabling the model to effectively learn long-range relationships. The multi-head attention mechanism allows the model to attend to multiple representation subspaces simultaneously, improving its ability to capture nuanced patterns. To incorporate sequence order, the Transformer employs positional encodings, which are added to input embeddings. These encodings are designed as fixed sinusoidal functions or learned embeddings, ensuring the model can distinguish between different positions within a sequence.

Key advantages of the Transformer include its scalability, reduced sequential computation and ability to capture global dependencies efficiently. These attributes make it a foundational model for modern deep learning applications in sequence modeling.

In the following, the main components with mathematical formulations are described:

2.4.1. Attention Mechanism

The Transformer model's attention mechanism computes the weighted sum of values (V) using queries (Q) and keys (K). For each query vector, attention scores are calculated as the scaled dot product between the query and all keys, followed by a softmax normalization. The equations are as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{2.11}$$

where:

- $Q \in \mathbb{R}^{T_q \times d_k}$: Query matrix, where T_q is the query sequence length, and d_k is the dimensionality of each query and key vectors.
- $K \in \mathbb{R}^{T_k \times d_k}$: Key matrix, where T_k is the key sequence length.
- $V \in \mathbb{R}^{T_k \times d_v}$: Value matrix, where d_v is the dimensionality of the value vector.

- $\frac{1}{\sqrt{d_k}}$ is the scaling factor to prevent large dot product values.
- softmax$(\cdot)$: Normalizes attention scores to sum to 1.

For self-attention:

(i) Q, K and V, derived from the same input sequence $X \in \mathbb{R}^{T \times d_{\text{model}}}$.

(ii) Thus, $T_q = T_k = T$, the sequence length of the input.

The output is a matrix of shape $\mathbb{R}^{T_q \times d_v}$.

Multi-Head Attention: To capture different representation subspaces, the Transformer uses Multi-Head Attention, defined as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{2.12}$$

where each attention head is calculated as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2.13}$$

and:

- $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$: Learnable weight matrices for the i-th attention head.
- $W^O \in \mathbb{R}^{h \cdot d_v \times d_{model}}$: Output projection matrix.
- d_{model}: Dimensionality of the input embeddings.
- h: Number of attention heads.

Positional Encoding: To incorporate sequence order information, the Transformer uses Positional Encoding, computed as:

$$PE_{\text{pos},2i} = \sin\left(\frac{\text{pos}}{10000^{2i/d_{model}}}\right), \quad PE_{\text{pos},2i+1} = \cos\left(\frac{\text{pos}}{10000^{2i/d_{model}}}\right), \tag{2.14}$$

where:

- pos: Position index in the sequence.
- i: Dimension index.

- d_{model}: Dimensionality of the input embeddings.

The positional encoding matrix has a shape of $\mathbb{R}^{T \times d_{model}}$, where T is the sequence length.

Transformer Encoder Layer: Each Encoder layer consists of:

(i) Multi-Head Attention:

$$\text{Attention Output} = \text{MultiHead}(Q, K, V) \quad \in \mathbb{R}^{T \times d_{model}}. \tag{2.15}$$

(ii) Add & Normalize:

$$\text{Norm}_1 = \text{LayerNorm}(X + \text{Attention Output}) \quad \in \mathbb{R}^{T \times d_{model}}. \tag{2.16}$$

(iii) Feedforward Neural Network (FFN):

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2. \tag{2.17}$$

(iv) Add & Normalize:

$$\text{Norm}_2 = \text{LayerNorm}(\text{Norm}_1 + \text{FFN}(\text{Norm}_1)) \quad \in \mathbb{R}^{T \times d_{model}}. \tag{2.18}$$

2.4.2. Hyperparameter Tuning in Transformers

For Transformer architectures, which rely on complex mechanisms such as self-attention and feedforward networks, the choice of hyperparameters plays a pivotal role in determining the model's effectiveness. Transformers involve several hyperparameters that influence their performance:

- Embedding Dimension (d_{model}): Controls the size of the vector space for input embeddings.
- Number of Attention Heads (n_{head}): Determines the granularity of attention mechanisms.
- Number of Layers (n_{layers}): Impacts the depth and capacity of the model.
- Feedforward Network Dimension (d_{ff}): Governs the size of the hidden layers in

the feedforward network.

- Dropout Rate: Regularizes the model by preventing overfitting.
- Learning Rate: Determines the pace of model weight updates during training.
- Batch Size: Specifies the number of samples processed per training iteration.

Hyperparameter tuning for Transformers poses several challenges:

- Computational Cost: Exploring large search spaces requires significant computational resources.
- Overfitting to Validation Set: Excessive tuning may lead to overfitting to the validation data.

Efficient hyperparameter tuning is vital for optimizing Transformer models, ensuring a balance between model capacity, computational efficiency, and predictive accuracy. Future work may explore automated and dynamic hyperparameter adjustment during training.

2.5. Financial Markets

Capital markets facilitate the exchange of financial assets with its actors. Some of the assets (financial instruments or securities) traded in the financial markets are stocks, bonds, derivatives and commodities [44]. A common stock (typically just called a stock) represents a share of ownership in a corporation. It is a security that is a claim on the earnings and assets of the corporation [45].

The World Bank provides comprehensive data on stock market capitalization, specifically the market capitalization of listed domestic companies expressed as a percentage of GDP or in current USD. In 2022, the global capitalization of the stock market reached $93.69 trillion, equivalent to 106. 2% of global GDP, highlighting the significant role of stock markets in the overall economy[1] .

[1]\url{https://data.worldbank.org/indicator/CM.MKT.LCAP.GD.ZS}, \url{https://data.worldbank.org/indicator/CM.MKT.LCAP.CD}

There are several factors affecting stock markets. These factors can be categorized as exogenous and endogenous. Exogenous factors are external influences outside the direct operations of the stock market itself. The COVID-19 pandemic (2020), the Russia-Ukraine conflict (2022), and the 2008 financial crisis are examples of these factors. Some of these factors are external shocks [46], geopolitical events [46], international economic conditions, macroeconomic indicators such as interest rates, credit ratings of countries, inflation rate, unemployment rate, political and regulatory factors and commodity prices.

On the other hand, some of the endogenous variables are company fundamental factors such as price earnings ratio (P/E), market capitalization and size effect, liquidity and trading activity, investor sentiment and behavioral factors.

Investment under these high demanding markets is highly competitive. There are several factors that affect investment strategies. Evaluating the performance of trading strategies is a critical area of study. Numerous metrics and analyses have been developed to assess the success of these strategies [47].

Regardless of the chosen strategy, the ability to accurately predict stock movements significantly enhances performance. The Random Walk Theory is a financial theory that suggests that stock prices (and other financial asset prices) move in a random and unpredictable manner, following a "random walk". The idea is that the future price of a stock is independent of its past prices, meaning that price changes cannot be predicted based on historical trends, and there are no discernible patterns or trends that can consistently be exploited for profit. According to the theory, stock price changes are random and cannot be predicted with certainty. The future price of a stock is as likely to go up as it is to go down, and the magnitude of the change is equally unpredictable. The Random Walk Theory is often associated with the Efficient Market Hypothesis (EMH), particularly the weak form of EMH.

The Efficient Market Hypothesis (EMH) is a financial theory that states that financial markets are "informationally efficient", meaning that asset prices in the market reflect all available information at any given time [48]. According to EMH, it is impossible to consistently achieve higher-than-market returns (i.e., "beat the market") by using technical analysis or historical price data. The core of the EMH is that market participants are rational, and prices adjust quickly to incorporate new information, making it difficult for any investor to consistently outperform the market.

[49], in his influential book , extended these ideas, emphasizing that stock prices move in a random manner, independent of past trends, and that the Random Walk Theory is a natural consequence of an efficient market. He argued that both technical analysis and fundamental analysis are ineffective in consistently outperforming the market.

On the other hand, these theories debated with different points of view, especially in the areas of market inefficiencies and behavioral finance. In this influential paper, [50] challenges the notion that stock prices follow a random walk and argues that stock price volatility is much higher than can be justified by changes in dividends, indicating that markets are not efficient in the way EMH suggests. His findings suggest that prices are often driven by factors like investor sentiment and speculation, rather than just fundamental information. [51] argues that asset pricing models based on rational expectations (like EMH) are incomplete and should incorporate behavioral factors such as emotions, biases, and imperfect information. [51] emphasizes that these psychological factors can lead to predictable price movements, contradicting the idea of a random walk. [52] argues that markets are adaptive rather than fully efficient, as assumed by EMH. He proposes the Adaptive Markets Hypothesis (AMH), which suggests that financial markets evolve over time and market efficiency fluctuates depending on the experience and learning of market participants. According to AMH, inefficiencies may arise in certain situations, which contradicts the assumption that prices always follow a random walk.

3. METHODOLOGY

In this chapter, we introduce our proposed motif discovery mechanism for identifying recurring patterns in time series data. Each major step of the process is clarified with pseudocode, allowing for a clear understanding of the underlying algorithms and procedures. We then describe how the identified motifs can be incorporated into a forecasting application, detailing the necessary configurations.

The principal problem addressed in this study is to identify all possible motifs in a given time series while adhering to overlapping constraints. The goal is to discover meaningful patterns that maximize the coverage and significance of motifs without excessive overlap, ensuring that each motif contributes uniquely to the analysis. This requires a combination of algorithms to efficiently identify, refine, and select motifs while optimizing key parameters dynamically.

The main algorithm 1 integrates the processes of time series motif identification, optimization, and selection into a unified framework. It combines multiple subalgorithms and a feedback mechanism to iteratively refine the results and ensure meaningful motifs are selected. This algorithm also allows flexibility in the motif selection process by providing two parallel approaches: Binary Integer Programming (BIP) and a heuristic method. The final output is a set of optimized motifs, ready for further analysis or application.The details of this process are summarized in the algorithm 1 below.

Algorithm 1 Main Algorithm: Motif Identification and Optimization Process

Require: Input data, initial parameters

Ensure: Final set of selected motifs

Step 1: Run Algorithm 2 (Time Series Pattern Identification with Exclusion Zones)
    Obtain *ith_start_index_matrix* and *ith_row_distance_matrix*
Step 2: Apply Feedback Mechanism
    Refine *m_start* and *m_stop* based on the entropy and candidate motif space
Step 3: Re-run Algorithm 2 with Feedback
    Re-execute **Algorithm 2** with updated parameters
Step 4: Select Motifs Using Optimization Techniques
    Option 1: Binary Integer Programming (BIP)
        Solve the motif selection problem via BIP using the outputs of **Algorithm 2**
    Option 2: Heuristic Approach
        Use the heuristic **Algorithm** 3 as an alternate method for motif selection
Step 5: Finalize Motif Selection
    Use the selected method (BIP or heuristic) to determine the final set of motifs
return Final selected motifs

In our first algorithm 2 , we identify potential motifs and their instances across a range of window lengths. The MASS algorithm [4] is employed to compute the distance profile and correlation profile. The algorithm used to create the potential motifs can be found in below:

Algorithm 2 Time Series Pattern Identification with Exclusion Zones

Require: *data*: Input time series

m_start, *m_stop*: Range of query window lengths

s: Step size for query lengths

c: Correlation threshold

i: Number of instances

Ensure: *ith_start_index_matrix*, *ith_row_distance_matrix*

Initialize *ith_start_index_matrix* and *ith_row_distance_matrix* as empty
for *start_index* in *data* **do**
  for m from *m_start* to *m_stop*, step size s **do**
    Compute *distance_profile* and *corr_profile*
    Apply exclusion zones and select instances with correlation $> c$
    if at least i instances are found **then**
      ith_row_distance_matrix ← Instances, their correlations and mean of correlations
    end if
  end for
  ith_start_index_matrix ←Find the query length m with the best mean correlation
end for
return *ith_start_index_matrix*, *ith_row_distance_matrix*

The algorithm 2 produces two primary outputs: ith start index matrix and ith row distance matrix. These data structures are dictionaries that store and organize results from the time series pattern identification process. Their roles and relationships are explained below.

The ith start index matrix dictionary provides a summary of the best motif identified for each starting index in the time series.

- Keys: Each key represents the starting index of a motif family within the time series.

- Values: The value is the optimal length of the motif (denoted as m) for the corresponding starting index.

This matrix acts as a high-level summary, mapping each motif's starting index to its most suitable length based on the algorithm's evaluation.

The ith row distance matrix dictionary provides detailed information about motifs for each starting index and different motif lengths.

- Keys: Each key corresponds to a starting index (matching the keys in ith start index matrix).
- Values: Each value is a nested dictionary, where:
  - The key is a motif length m.
  - The value is a list containing:
    (i) Instance indices: The starting indices of all motif instances identified for the given length m.
    (ii) Correlation values: The correlation scores for the identified motif instances.
    (iii) Mean correlation: The average correlation score for all instances of the motif at length m.

The ith start index matrix and ith row distance matrix are interconnected. For each starting index in the time series, ith start index matrix specifies the best motif length (m) for that starting index. Using this length, the corresponding motif instances and details can be accessed in ith row distance matrix.

3.1. Range of Motif Length

Determining the appropriate range for motif length is a critical step in the process of motif extraction. The selection of a shorter motif length can lead to trivial results, as shorter motifs are more likely to occur frequently within the dataset due to the

high probability of observation. This can result in the identification of redundant or uninformative patterns that do not contribute meaningful insights to the analysis.

Conversely, choosing an appropriately balanced motif length ensures that the extracted motifs capture significant and non-trivial patterns. These patterns are essential for understanding the underlying dynamics of the data and for producing robust results in downstream analyses. Therefore, careful consideration of the motif length range is indispensable for achieving reliable and interpretable outcomes in motif extraction studies

Entropy, a fundamental concept in information theory, measures the uncertainty or diversity in a probability distribution and is widely used in probabilistic modeling [53]. A higher entropy indicates a more diverse set of motif lengths, while a lower entropy highlights concentration and potential triviality.

The entropy H of a distribution is calculated as:

$$H = -\sum_{i=1}^{j} p_i \cdot \log(p_i), \tag{3.1}$$

where:

- p_i is the probability of the ith motif length, computed as the frequency of i divided by the total number of occurrences.
- j is the total number of unique motif lengths.

The proportion of candidate motifs identified within a given motif length range plays a crucial role in ensuring meaningful pattern discovery. If motifs are detected too frequently across the dataset, there is a risk of capturing trivial and redundant patterns that do not contribute valuable insights. On the other hand, if motifs appear too sparsely, essential underlying structures may be overlooked, reducing the effectiveness of the analysis.

To evaluate motif prevalence, we define two key measures:

- Motif Space: The number of indices where a motif has been identified as the best match. This represents the actual locations in the time series where significant motifs appear. Since the ith start index matrix stores the best motif identified for each starting index that contains a motif, its length is equal to the motif space.
- Total Space: The total number of indices in the time series that are eligible for motif detection within the given motif length range. Since motifs must fit within the time series without exceeding its boundaries, the number of valid starting indices decreases as motif length increases. The total space can be formulated as:

$$\text{Total Space} = n - (\text{m}_{\text{start}}) + 1 \tag{3.2}$$

where n is the length of the time series, and m_{start} is the minimum motif length.

The Candidate Motifs %, derived from these measures, serves as an indicator of how frequently motifs occur within the considered space. A balanced proportion of candidate motifs helps in identifying meaningful and representative patterns while avoiding excessive repetition. This balance ensures that the detected motifs capture meaningful patterns within the time series, rather than being influenced by overly permissive or restrictive motif length constraints.

We will use both entropy and Candidate Motifs % to determine the search space for the motif length range.

3.2. Extracting the Motifs

After determining the optimal window length for all available starting indices, the next question is: how many distinct motifs can exist within a time series? Addressing this question requires a robust mechanism to select a set of motifs that accurately reflects the underlying structure of the data. In principle, this selection process can be formulated as a binary integer programming problem, which guarantees an optimal solution by exhaustively exploring all feasible combinations subject to given constraints.

However, integer programming can quickly become computationally expensive as the size of the dataset grows. Consequently, while an exact solution may be theoretically appealing, practical considerations often motivate the adoption of more scalable alternatives. This section presents two contrasting strategies: first, an optimization approach leveraging binary integer programming, and second, a heuristic approach designed to balance accuracy with efficiency.

3.2.1. Binary Integer Programming

This optimization problem involves selecting rows from a given set (motif space), where each row represents a motif and contains the starting indices of its instances. Each row is associated with a single "total correlation" value, which is the sum of the correlation values stored in the ith row distance matrix. We wish to select a subset of these rows such that:

- No two selected rows share any overlapping instances.
- The sum of the correlation values of the selected rows is maximized.

Input Data: Let $\mathcal{I} = \{1, 2, \ldots, n\}$ denote the set of all rows (motifs). For each row $i \in \mathcal{I}$:

- L_i: The fixed length of each instance in row i.
- Motif Instances $\mathcal{S}_i$: Set of start instances of the row i $\{s_{i,1}^{\text{start}}, s_{i,2}^{\text{start}}, \ldots\}$. Hence,

$$\mathcal{S}_i \;=\; \Big\{ [\, s_{i,m}^{\text{start}},\, s_{i,m}^{\text{start}} + L_i \,) \;:\; m \in M_i \Big\} \tag{3.3}$$

  where M_i indexes all instances in row i.
- Correlation C_i: A scalar value representing the total correlation contributed by selecting row i. This could be the sum or mean of correlations of the instances within $\mathcal{S}_i$.

Decision Variables: We define one binary variable for each row $i \in \mathcal{I}$:

$$y_i \;=\; \begin{cases} 1, & \text{if row (motif) } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

If $y_i = 1$, then *all instances* in $\mathcal{S}_i$ are considered active in the solution.

Objective Function: Our goal is to maximize the total correlation of selected rows

$$\max \quad Z \;=\; \sum_{i \in \mathcal{I}} C_i \, y_i. \tag{3.5}$$

Selecting row i (i.e., $y_i = 1$) adds C_i to the objective value; not selecting it ($y_i = 0$) adds nothing.

Constraints:

(i) Non-Overlapping Intervals: Any two chosen rows must not share any overlapping instances. Concretely, for any pair (i, k) with $i < k$, if there exist instances $[a, b) \in \mathcal{S}_i$ and $[c, d) \in \mathcal{S}_k$ such that $[a, b) \cap [c, d) \neq \emptyset$, we require

$$y_i + y_k \;\leq\; 1. \tag{3.6}$$

Thus, if any one of row i's instances intersects any one of row k's instances, the solver cannot select both i and k together.

(ii) Binary Constraints: Each decision variable y_i is constrained to be binary:

$$y_i \in \{0, 1\}, \quad \forall \, i \in \mathcal{I}. \tag{3.7}$$

We can summarize the problem as a Binary Integer Programming (BIP) problem:

Decision Variables:

$$y_i \in \{0, 1\} \quad \forall \, i \in \mathcal{I},$$

Objective:

$$\max \quad \sum_{i \in \mathcal{I}} C_i \, y_i,$$

Subject to:

(Non-overlapping) :

$$\forall\, i < k,\ \text{if } \exists\, [a,b) \in \mathcal{S}_i,\ \exists\, [c,d) \in \mathcal{S}_k \text{ with } [a,b) \cap [c,d) \neq \emptyset,$$

$$\text{then } y_i + y_k \leq 1,$$

(Binary variables) :

$$y_i \in \{0,1\} \quad \forall\, i \in \mathcal{I}.$$

This problem is solvable via standard Mixed Integer Linear Programming (MILP) solvers such as *IBM ILOG CPLEX*, *Gurobi*, or open-source tools and libraries like *PULP*.

3.2.2. Heuristic Approach

The proposed heuristic selection mechanism iteratively identifies and selects the optimal rows from a dataset while ensuring non-overlapping index ranges. The rows are determined based on the starting indices and their corresponding instances, as derived from Algorithm 2. The process is driven by a scoring system that evaluates each candidate using the mean correlation values of its indices. To prevent overlapping selections, the mechanism employs a masking strategy. At each step, the highest scoring non-overlapping candidate is selected, and its associated index ranges are masked to exclude them from future consideration. This greedy approach ensures the prioritization of high quality candidates while maintaining adherence to non-overlapping constraints.

The algorithm initializes a mask to track indices that have already been selected. It then iterates through all available rows, checking each candidate for overlap with the masked indices. If a candidate's index range does not overlap and its mean correlation value exceeds the current best score, it is marked as the best selection for that iteration. Once selected, the ranges of the chosen row are added to the mask, and the process repeats until no valid candidates remain. This iterative approach effectively handles

overlapping constraints and guarantees the selection of the most optimal rows in each iteration.

Algorithm 3 Heuristic Selection Mechanism for Non-Overlapping Motifs

Require: *ith_start_index_matrix*: Dictionary of masking lengths

ith_row_distance_matrix: Dictionary of motifs (rows) with indices and mean correlation scores

Ensure: *selected_rows*: List of selected keys

Initialize *masked_indices* as an empty set
Initialize *selected_rows* as an empty list
while True **do**
  Set *best_selection* ← None and *best_mean_corr* ← $-\infty$
  for *key, subkey, data* in *ith_row_distance_matrix* **do**
    Extract *start_indices* from *data[0]* and mean correlation from *data[2]*
    Check if *start_indices* overlap with *masked_indices*
    if *no overlap and data[2]* > *best_mean_corr* **then**
      Set *best_selection* ← (*key, subkey*) and *best_mean_corr* ← *data[2]*
    end if
  end for
  if *best_selection is None* **then**
    break
  end if
  Add ranges from *start_indices* of *best_selection* to *masked_indices*
  Append *key* from *best_selection* to *selected_rows*
end while
return *selected_rows*

3.3. Forecasting

After deriving motifs, this part outlines the methodology for integrating these motifs into a forecasting model and evaluating their impact on prediction accuracy. The motifs, which capture recurring patterns in the time series, are embedded as

features to enrich the forecasting process. A Transformer model is employed for the forecasting task. The methodology proceeds as follows:

3.3.1. Data Preparation

3.3.1.1. Train-Validation-Test Split. To evaluate the performance of the forecasting model, the dataset was divided into three subsets: training, validation, and testing. This splitting strategy ensures that the model is trained and validated on distinct data, mitigating the risk of overfitting and providing an unbiased evaluation of its predictive capabilities.

The training set is the largest portion of the data and is used to optimize the model's parameters. By exposing the model to the patterns and dependencies in the training data, the learning process minimizes the error with respect to a pre-defined loss function.

The validation set is a smaller portion of the data, separate from the training set, used to monitor the model's performance during training. This set allows for hyperparameter tuning and serves as a measure of the model's ability to generalize to unseen data. Early stopping techniques were applied based on validation loss to prevent overfitting.

The test set is completely isolated from both the training and validation processes. It is used only after the training and tuning phases are complete to provide an unbiased assessment of the model's performance on new, unseen data.

In this study, the dataset was split into training, validation, and testing subsets based on temporal ordering to respect the sequential nature of the data. The proportions used were typically 70% for training, 15% for validation, and 15% for testing. This division ensured that the model's predictions on the test set could be reliably evaluated without information leakage from future time steps.

3.3.1.2. Scaling of the Original Time Series. Scaling is a crucial preprocessing step in time series analysis, particularly when employing machine learning models. In this study, the original univariate time series data was scaled using the Min-Max Scaling method. This approach rescales the data to a fixed range, typically between 0 and 1, which is mathematically expressed as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{3.8}$$

where x represents the original data and x' is the scaled data.

Scaling the data is essential for the following reasons:

- Model Stability: Many machine learning models, especially those involving gradient-based optimization (e.g. neural networks), perform better and converge faster when the data is within a small, consistent range.
- Feature Homogeneity: Scaling ensures that features with larger numerical ranges do not dominate the optimization process, allowing the model to treat all input features equally.
- Compatibility with Activation Functions: In neural networks, activation functions such as sigmoid and hyperbolic tangent saturate for extreme input values, which can hinder learning. Scaling mitigates this effect.

The Min-Max Scaling was applied to the entire data set before splitting it into training, validation and testing subsets. This ensured consistency across all subsets and prevented information leakage from future time steps. The scaling parameters (min and max) were computed solely from the training data and subsequently applied to scale the validation and test sets.

3.3.2. Evaluation and Comparison Metrics

In this study, the performance of forecasting models are evaluated and compared using specific metrics. These metrics were chosen to provide a comprehensive understanding of the model's predictive accuracy, robustness, and generalizability.

The following metrics are used for evaluation and comparison:

(i) Mean Squared Error (MSE)

MSE quantifies the average squared difference between the predicted and actual values. It is sensitive to large errors, making it a reliable indicator of significant deviations:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2. \tag{3.9}$$

Why chosen: MSE is widely used in regression tasks as it emphasizes large prediction errors, which are critical in time series forecasting.

(ii) Mean Absolute Error (MAE)

MAE calculates the average absolute difference between predictions and ground truth. Unlike MSE, it treats all errors equally:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|. \tag{3.10}$$

Why chosen: MAE is interpretable in the original units of the data, making it suitable for practical comparisons.

The combination of these metrics ensures a robust and interpretable evaluation framework. Each metric highlights a distinct aspect of model performance and enables a comprehensive comparison.

4. EXPERIMENTS AND RESULTS

In this chapter, we detail the experimental setup and procedures designed to validate the effectiveness of our proposed motif discovery framework. We begin by describing the datasets used, highlighting their characteristics and relevance to real-world scenarios. We then outline the parameter configurations employed to systematically detect motifs across varying conditions, including different motif lengths and instances, while ensuring robustness and generalizability. An in-depth discussion of motif discovery results follows, analyzing critical aspects such as motif diversity, and the comparative effectiveness of the Binary Integer Programming (BIP) and heuristic selection methods. Building on these findings, we proceed with comprehensive forecasting experiments, integrating the identified motifs into a Transformer-based forecasting pipeline. This enables us to evaluate the practical implications and potential advantages of incorporating motif information into predictive modeling tasks. The outcomes presented here provide both empirical validation of our methodology and clear insights into the practical utility of motif discovery within our targeted application domain.

4.1. Experimental Data

As we discussed in the chapter 2, financial markets present both challenges and significant opportunities. For our experiments, we therefore draw on financial market data. The five univariate time series datasets summarized in Table 4.1 will be utilized for both motif discovery and forecasting experiments. The granularity of all datasets is daily. Although the start and end dates span the same time interval, the data lengths vary due to differences in data availability. The data used in this study was obtained from [54].

Table 4.1. Description of Input Data.

Ticker	**Explanation**	**Type**	**Length**	**Start Date**	**End Date**
ISCTR	Türkiye İş Bankası	Stock	4003	30.01.2009	09.01.2025
GAU/TRY	Gold Gram Turkish Lira	Commodity	3530	11.04.2013	09.01.2025
LCO	Brent Oil Futures	Commodity	4118	30.01.2009	09.01.2025
USD/TRY	US Dollar Turkish Lira	Currency	4160	30.01.2009	09.01.2025
XU100	BIST 100 Index	Index	4004	30.01.2009	09.01.2025

4.2. Motif Discovery

In our motif discovery approach, there are some user parameters. These parameters need to be tuned with some set of experiments.

- Step size (s): 1
- Number of instances (i): {4, 6, 8}
- Window length range start (m_{start}): {5, 10, 20, 30}
- Window length range stop (m_{stop}): 50
- Correlation threshold (c) : {0.95, 0.97}

Table 4.2. Experimental Results for Motif Discovery on ISCTR

Parameters					**Parameter Tuning**				**BIP**			**Heuristic**		
m-start	**m-stop**	**s**	**i**	**c**	**Motif Space**	**Total Space**	**Cand. Sp. %**	**Entropy**	**Motifs Count**	**Inst. Count**	**Total Corr.**	**Motifs Count**	**Inst. Count**	**Total Corr.**
30	50	1	4	0.97	129	3974	3%	2.96	3	15	14.70	1	5	4.92
20	50	1	4	0.97	192	3984	5%	3.24	3	18	17.62	2	10	9.83
10	50	1	4	0.97	849	3994	21%	2.45	22	124	121.59	16	91	89.49
5	50	1	4	0.97	3839	3999	96%	0.60	NA	NA	NA	53	871	858.17
20	50	1	6	0.97	64	3984	2%	3.03	1	10	9.78	1	7	6.87
20	50	1	8	0.97	19	3984	0%	2.55	1	12	11.73	1	9	8.81
20	50	1	4	0.95	1131	3984	28%	3.09	10	61	74.18	5	38	36.88
20	50	1	6	0.95	900	3984	23%	3.12	6	52	50.23	3	29	28.20
20	50	1	8	0.95	724	3984	18%	3.13	5	50	48.17	2	24	23.34

Table 4.2 summarizes the results of motif extraction based on different parameter configurations. The table is divided into four main sections:

(i) Parameter Tuning This section evaluates different parameter combinations to determine the most suitable motif extraction settings. The key metrics have been previously defined in the Chapter 3. These values help assess motif coverage, prevalence, and diversity under different configurations, providing insight into how parameter choices influence the results.

(ii) BIP Method This section evaluates motif selection based on an optimization approach:
   - Motifs Count: The number of distinct motifs selected.
   - Instance Count: The total number of motif occurrences (the number of instances for all motifs).
   - Total Correlation: The sum of correlation values for the selected motifs.

(iii) Heuristic Method A secondary motif selection approach, with the following metrics:
   - Motifs Count: The number of distinct motifs selected.
   - Instance Count: The total number of motif occurrences.
   - Total Correlation: The sum of correlation values for the selected motifs.

The same structure for GAU/TRY, LCO, USD/TRY and XU100 can be found in Tables 4.3, 4.4, 4.5 and 4.6 , respectively.

Table 4.3. Experimental Results for Motif Discovery on GAU/TRY

Parameters					Parameter Tuning				BIP			Heuristic		
m-start	**m-stop**	**s**	**i**	**c**	**Motif Space**	**Total Space**	**Cand. Sp. %**	**Entropy**	**Motifs Count**	**Inst. Count**	**Total Corr.**	**Motifs Count**	**Inst. Count**	**Total Corr.**
30	50	1	4	0.97	183	3501	5%	2.55	3	19	18.64	3	18	17.67
20	50	1	4	0.97	358	3511	10%	2.86	5	28	27.42	3	24	23.55
10	50	1	4	0.97	940	3521	27%	2.65	16	95	93.12	11	61	60.07
5	50	1	4	0.97	3351	3526	95%	0.70	NA	NA	NA	45	745	733.69
20	50	1	6	0.97	210	3511	6%	2.64	3	22	21.55	2	19	18.65
20	50	1	8	0.97	126	3511	4%	2.18	2	21	20.57	1	12	11.80
20	50	1	4	0.95	1201	3511	34%	3.22	6	54	52.19	2	35	34.07
20	50	1	6	0.95	1002	3511	29%	3.22	2	44	42.53	2	37	35.98
20	50	1	8	0.95	847	3511	24%	3.17	2	38	36.73	1	30	29.20

Table 4.4. Experimental Results for Motif Discovery on LCO

Parameters					Parameter Tuning				BIP			Heuristic		
m-start	**m-stop**	**s**	**i**	**c**	**Motif Space**	**Total Space**	**Cand. Sp. %**	**Entropy**	**Motifs Count**	**Inst. Count**	**Total Corr.**	**Motifs Count**	**Inst. Count**	**Total Corr.**
30	50	1	4	0.97	14	4089	0%	2.04	2	10	9.79	2	10	9.79
20	50	1	4	0.97	114	4099	3%	2.22	5	26	25.47	3	15	14.72
10	50	1	4	0.97	793	4109	19%	1.98	21	120	117.78	15	94	92.38
5	50	1	4	0.97	3949	4114	96%	0.60	NA	NA	NA	56	978	963.32
20	50	1	6	0.97	26	4099	1%	1.82	2	16	15.65	2	14	13.72
20	50	1	8	0.97	7	4099	0%	1.35	1	11	10.75	1	9	8.82
20	50	1	4	0.95	1122	4099	27%	3.16	10	64	61.77	5	35	34.05
20	50	1	6	0.95	750	4099	18%	3.00	5	48	46.25	3	27	26.25
20	50	1	8	0.95	534	4099	13%	2.74	4	47	45.23	2	24	23.33

Table 4.5. Experimental Results for Motif Discovery on USD/TRY

Parameters					First Process				BIP			Heuristic		
m-start	**m-stop**	**s**	**i**	**c**	**Motif Space**	**Total Space**	**Cand. Sp. %**	**Entropy**	**Motifs Count**	**Inst. Count**	**Total Corr.**	**Motifs Count**	**Inst. Count**	**Total Corr.**
30	50	1	4	0.97	678	4131	16%	2.92	3	19	18.60	2	10	9.85
20	50	1	4	0.97	870	4141	21%	3.30	6	36	35.29	5	27	26.54
10	50	1	4	0.97	1496	4151	36%	3.14	20	128	125.66	15	84	82.64
5	50	1	4	0.97	3991	4156	96%	0.78	NA	NA	NA	51	952	938.05
20	50	1	6	0.97	705	4141	17%	3.31	2	27	26.46	2	14	13.76
20	50	1	8	0.97	573	4141	14%	3.26	2	23	22.56	1	9	8.86
20	50	1	4	0.95	1692	4141	41%	3.22	8	65	62.80	6	67	65.07
20	50	1	6	0.95	1536	4141	37%	3.24	3	59	57.16	2	27	26.36
20	50	1	8	0.95	1403	4141	34%	3.27	2	56	54.27	2	32	31.15

Table 4.6. Experimental Results for Motif Discovery on XU100

Parameters					First Process				BIP			Heuristic		
m-start	**m-stop**	**s**	**i**	**c**	**Motif Space**	**Total Space**	**Cand. Sp. %**	**Entropy**	**Motifs Count**	**Inst. Count**	**Total Corr.**	**Motifs Count**	**Inst. Count**	**Total Corr.**
30	50	1	4	0.97	445	3975	11%	2.90	4	21	20.58	2	14	13.79
20	50	1	4	0.97	655	3985	16%	3.29	5	28	27.48	2	14	13.79
10	50	1	4	0.97	1354	3995	34%	2.94	25	138	135.41	17	109	107.16
5	50	1	4	0.97	3790	4000	95%	0.74	NA	NA	NA	52	1073	1056.93
20	50	1	6	0.97	515	3985	13%	3.24	4	28	27.38	2	16	15.73
20	50	1	8	0.97	396	3985	10%	3.21	2	24	23.47	2	9	8.88
20	50	1	8	0.95	1577	3985	40%	3.20	13	72	69.56	6	46	44.71
20	50	1	6	0.95	1353	3985	38%	3.20	3	58	56.04	3	53	51.47
20	50	1	8	0.95	1172	3985	29%	3.22	3	54	52.26	3	47	45.72

4.3. Motif Discovery Results

This section analyzes the impact of different parameter settings on motif extraction and selection. The results highlight key trends in motif prevalence, entropy, and selection strategies.

4.3.1. Influence of Minimum Motif Length m_{start} on Candidate Motifs Percentage

As observed from the results, the selection of m_{start} has a significant effect on the percentage of candidate motifs. As m_{start} decreases, the proportion of candidate motifs increases. For instance, when $m_{\text{start}} = 5$, 96% of the available indices were identified as candidate motifs, suggesting a high level of redundancy. Conversely, for $m_{\text{start}} = 30$, only 3% of the indices were detected as motifs, indicating that longer motifs are rarer but may contain more meaningful patterns.

These results suggest that shorter motifs are more frequent and may capture trivial or redundant patterns, while longer motifs, though less prevalent, are likely to represent more structured and informative patterns within the time series.

4.3.2. Entropy as a Measure of Motif Diversity

Entropy serves as a crucial metric for assessing motif diversity. Lower entropy values (e.g., 0.60 for $m_{\text{start}} = 5$) indicate a highly repetitive motif structure, whereas higher entropy values (e.g., 3.24 for $m_{\text{start}} = 20$) reflect a more balanced distribution of motif lengths. The results demonstrate that entropy can act as a guiding measure to prevent excessive redundancy in motif selection and ensure the identification of meaningful motif variations.

4.3.3. Comparison of BIP and Heuristic Methods

As observed from the results, for all parameter combinations, the BIP method outperforms the heuristic method in terms of the number of extracted motifs. However, using BIP can be computationally exhaustive. Notably, when $m_{\text{start}} = 5$, the method fails to generate results within the allocated time, highlighting its increased computational cost in certain scenarios.

A comparative analysis of the BIP and heuristic methods highlights key differences in motif selection. The BIP method identifies more motifs and achieves higher total correlation values (e.g. 121.59 for $m_{\text{start}} = 10$), indicating that it selects a larger set of relevant motifs. The Heuristic method, on the other hand, selects fewer motifs but ensures that the selected motifs have high correlation values, suggesting a more filtered selection process.

4.3.4. Effect of Instance Count (i) on Parameter Tuning

The instance count parameter (i) significantly impacts motif selection by enforcing a stricter recurrence requirement, leading to a reduction in motif space. When $m_{\text{start}} = 20$ and $c = 0.97$, motif space decreases from 192 to 19 as i increases from 4 to 8. This demonstrates that a higher i value removes many motifs that do not meet the stricter occurrence constraint, potentially filtering out meaningful patterns. Since

increasing i in combination with a high c further restricts motif selection, it is essential to carefully balance these parameters to avoid excessive exclusion of relevant motifs.

The key findings from the experimental results can be summarized as follows:

- Short motifs (m_{start} small) result in excessive motif occurrences, whereas longer motifs are more selective and meaningful.
- Entropy serves as an effective metric for identifying motif diversity, with higher entropy indicating more varied motif structures.
- A high instance count constraint (i) can excessively limit motif selection, potentially missing meaningful patterns.
- The BIP method selects a greater number of motifs, while the heuristic method enforces stricter selection criteria, leading to fewer but potentially stronger motifs.
- NA values in the BIP section indicate cases where the heuristic method could not find a solution within the given time limit.

These findings emphasize the importance of carefully tuning m_{start}, entropy, and motif selection strategies for extracting meaningful patterns from time series data. The results also highlight the trade-offs between BIP and heuristic selection methods, offering guidance on their optimal application depending on the analysis objectives.

Using this information, we present visualizations of the motifs and their instances identified by the BIP and heuristic models. The selected parameters for this demonstration are: $m_{\text{start}} = 20$, $i = 4$ and $c = 0.97$.

Figures 4.1 and 4.2 show the motifs and their instances on the ISCTR time series, respectively, using the full time index. The figures 4.3, 4.4, 4.5, 4.6, 4.7 illustrates the instances of a detected motif within the ISCTR time series, aligned based on a relative time index. Each line represents an instance of the motif, and the x-axis corresponds to the relative time index, ensuring that all instances are visualized in a standardized manner. The y-axis shows the Z-normalized values of the motif instances, allowing for

direct comparison by removing differences in mean and variance.

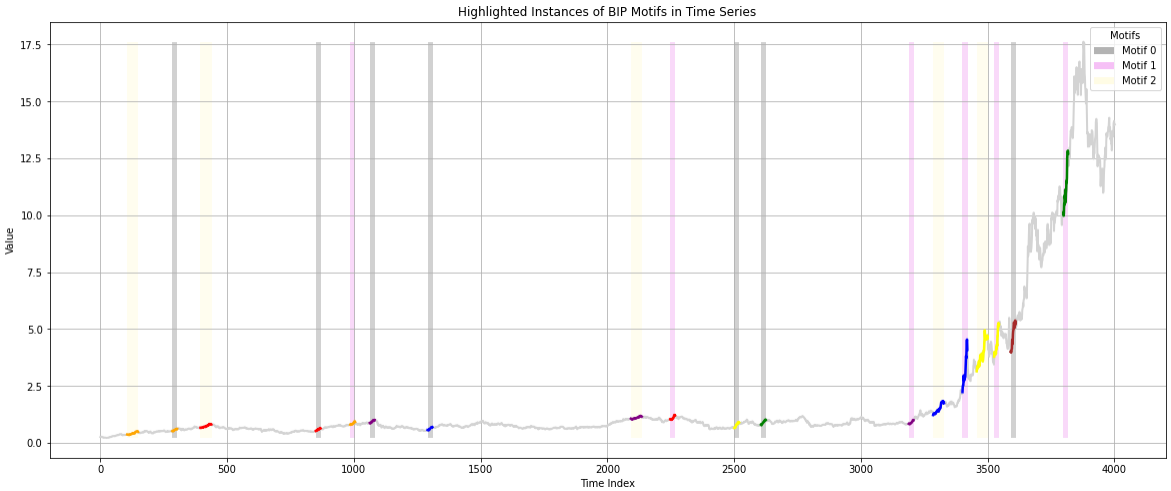

Figure 4.1. Highlighted Instances of BIP Motifs in Time Series

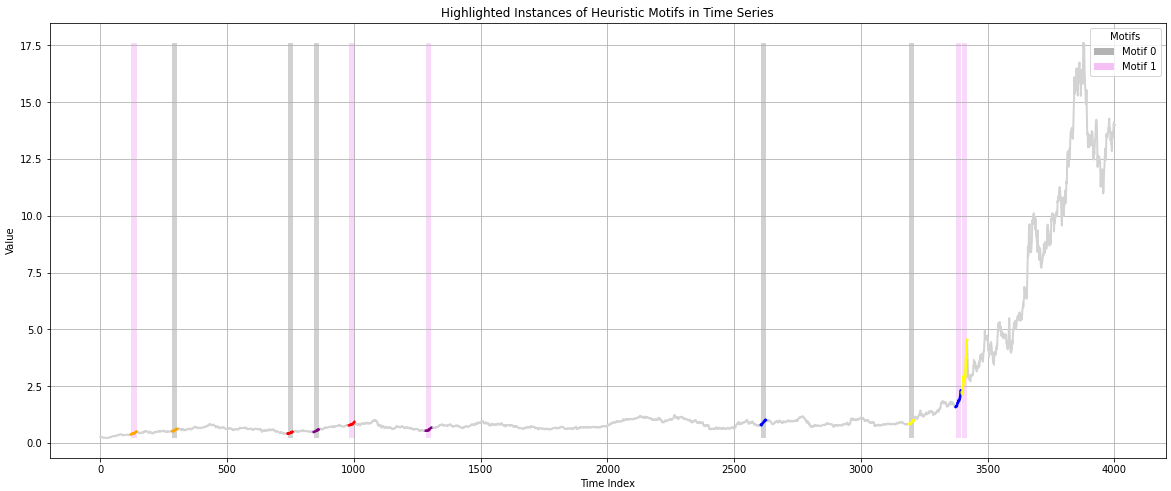

Figure 4.2. Highlighted Instances of Heuristic Motifs in Time Series

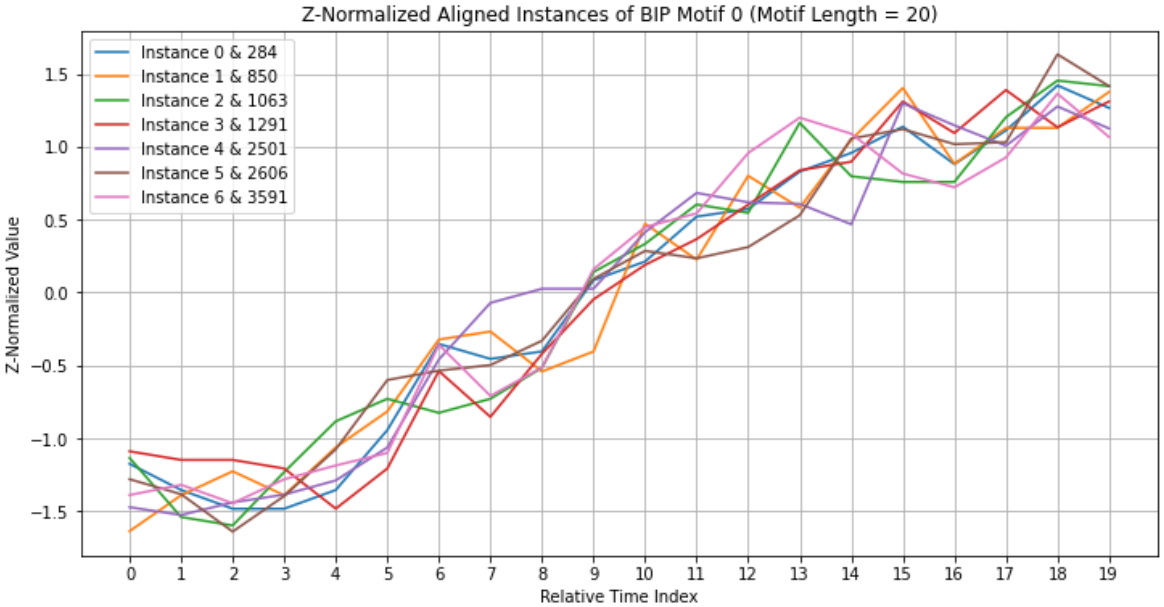

Figure 4.3. Z-Normalized Alligned Instances of BIP Motif 0

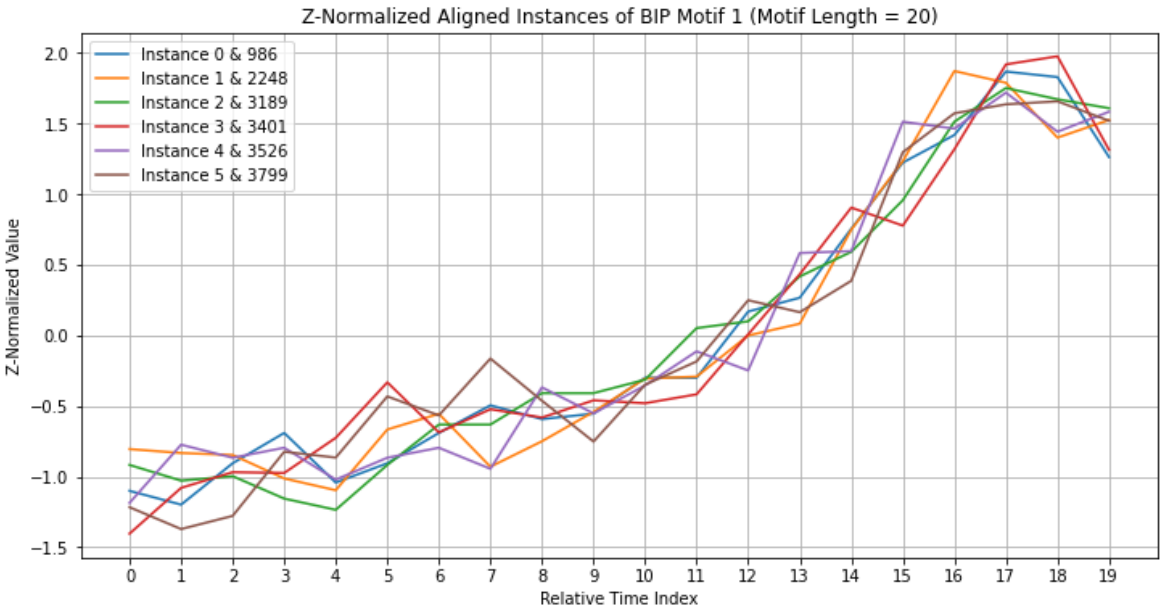

Figure 4.4. Z-Normalized Alligned Instances of BIP Motif 1

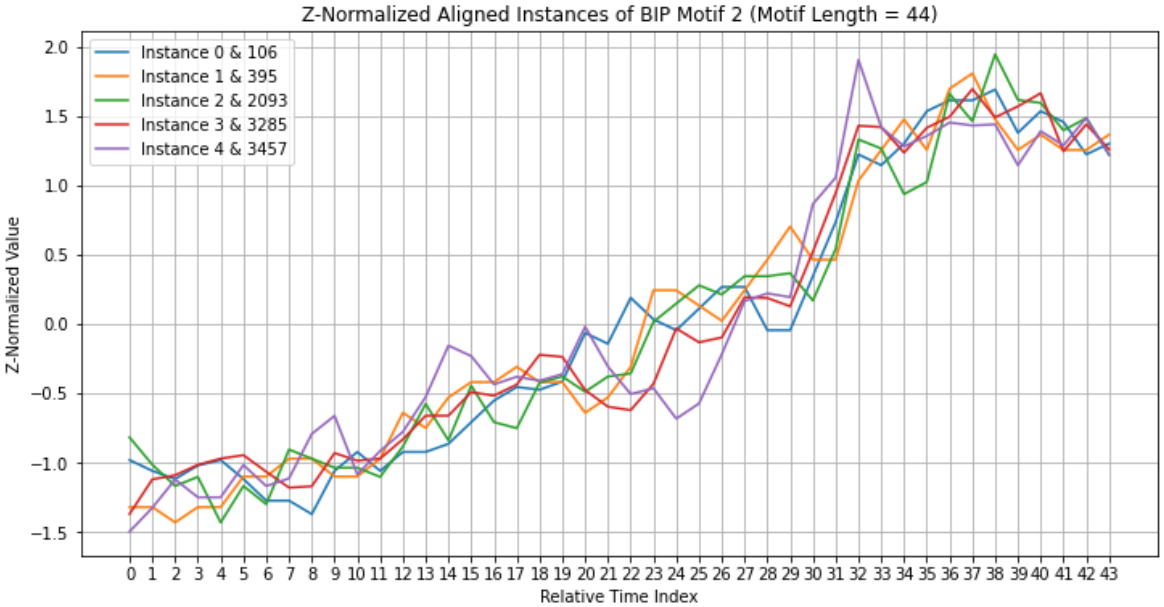

Figure 4.5. Z-Normalized Alligned Instances of BIP Motif 2

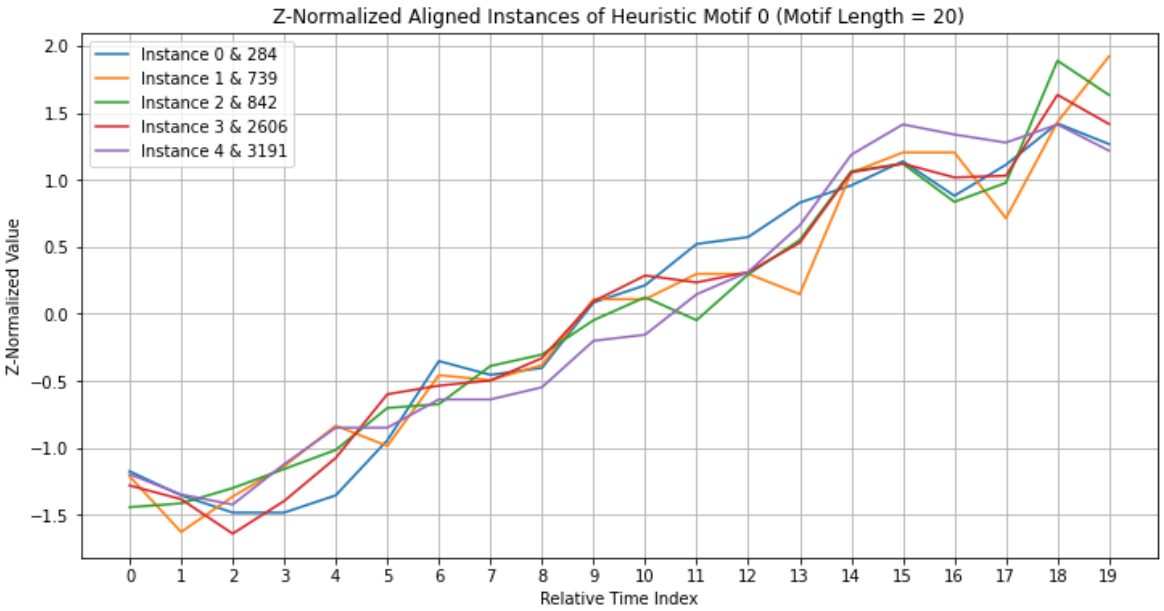

Figure 4.6. Z-Normalized Alligned Instances of Heuristic Motif 0

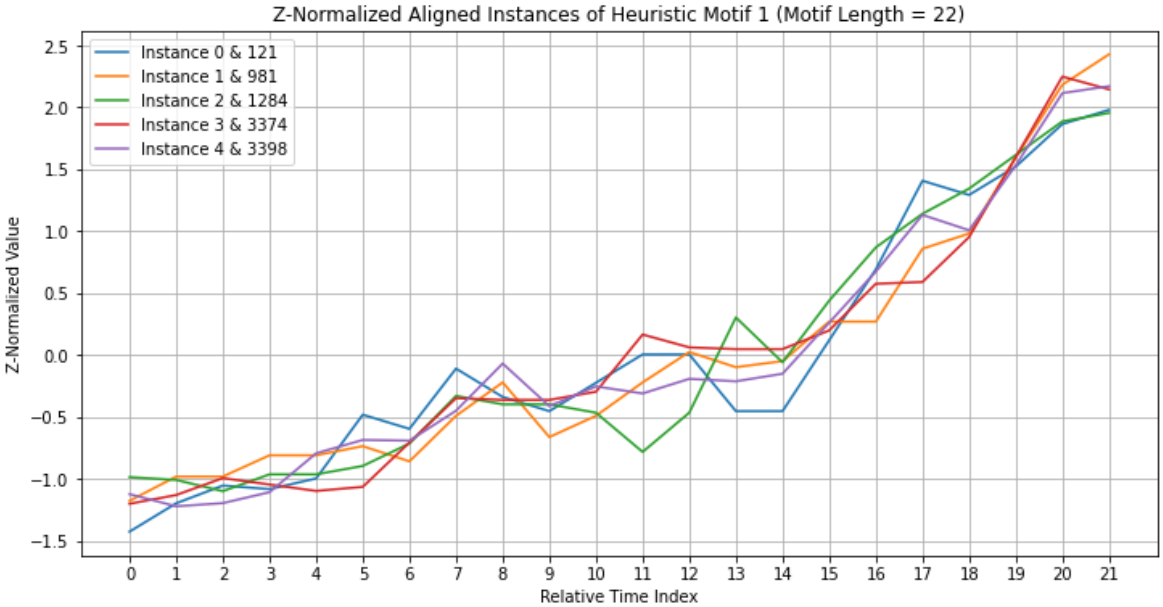

Figure 4.7. Z-Normalized Alligned Instances of Heuristic Motif 1

4.4. Forecasting

We used two datasets, ISCTR and GAU/TRY, for our forecasting experiments. Table 4.7 shows the selected motif parameter set for motif embeddings. Additionally, we conducted a one-step-ahead forecasting experiment.

Table 4.7. Parameter Selection for Different Datasets

Datasets	**m-start**	**m-stop**	**s**	**i**	**c**
ISCTR	20	50	1	4	0.97
GAU/TRY	30	50	1	4	0.97

To study the impact of input lookback window (detailed in 2.3.1) sizes, we conducted experiments with window size $W \in \{60, 100\}$. For a fair comparison, four models are evaluated.

(i) Naïve method, which simply repeats the last observed value in the look-back window.

(ii) Exponential Smoothing

(iii) A Vanilla Transformer Model

(iv) Transformer with Motif Embeddings.

For Transformer- based models, hyperparameter tuning was conducted over 30 trials using the Optuna framework, leveraging Bayesian optimization for efficient exploration of the search space. Separate tuning was performed for both of these models. The following hyperparameters were tuned:

- d_{model}: {32, 64, 128}
- n_{head}: {2, 4, 8}
- n_{layers}: {1, 2, 3}
- d_{ff}: {128, 256, 512}
- Dropout Rate: $[0.1, 0.3]$
- Learning Rate: $[10^{-4}, 10^{-2}]$
- Batch Size: {16, 32, 64}

In this study, number of epochs and early stopping mechanism were intentionally kept consistent across experiments with and without motif embedding to ensure a fair comparison between the two configurations.

4.5. Forecasting Results

Table 4.8 summarizes the performance metrics for the models of four forecasting models—Vanilla Transformer, Motif, Naive, and Exponential Smoothing—applied to two datasets, ISCTR and GAU/TRY. The performance of each model is assessed based on Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Table 4.8. Comparison of Methods on ISCTR and GAU/TRY Datasets

Datasets	**Methods**	**Vanilla Transformer**		**Motif Embedded Transformer**		**Naive**		**Exp. Smoothing**	
	Metrics	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
ISCTR	60	1.900	0.902	3.150	1.323	0.070	0.195	0.070	0.195
	100	3.052	1.191	2.152	1.064	0.070	0.195	0.070	0.196
GAU/TRY	60	88225.0	226.5	119246.2	268.8	463.1	14.9	460.7	14.9
	100	140911.4	291.1	66143.9	204.5	468.4	15.1	465.3	15.1

(i) Performance on ISCTR Dataset: The Naive and Exponential Smoothing models consistently achieve the lowest error values, with MSE ≈ 0.070 and MAE ≈ 0.195, regardless of the lookback window. This suggests that simple statistical models effectively capture the underlying patterns in the ISCTR dataset.

Conversely, the Vanilla and Motif Embedded Transformer models exhibit significantly higher errors. The Vanilla Transformer model's performance deteriorates when the lookback window increases from 60 to 100 (MSE increases from 1.900 to 3.052), indicating possible instability. The Motif model shows a slight improvement with an extended lookback window (MSE decreases from 3.150 to 2.152), but it remains far less effective than the simpler models.

(ii) Performance on GAU/TRY Dataset: Similar trends are observed in the GAU/TRY dataset. The Naive and Exponential Smoothing models maintain stable performance with MSE values around 460–468 and MAE values around 15 across different lookback windows. Their consistency indicates robustness and adaptability to the dataset's characteristics.

In contrast, the Vanilla and Motif Embedded Transformer models exhibit extremely high errors. The Vanilla Transformer's MSE increases significantly from 88,225 (lookback 60) to 140,911 (lookback 100), indicating that increasing the lookback window negatively impacts its ability to generalize. The Motif Embedded Transformer model, while showing some improvement with a longer lookback window (MSE reduces from 119,246 to 66,143), still performs poorly compared to the Naive and Exponential Smoothing models.

Overall, the results indicate that incorporating motif embeddings does not yield significant improvements over the baseline model without motif embeddings.

Additionally, this study contributes to the ongoing debate about the utility of technical analysis within the framework of the Efficient Market Hypothesis (EMH). By embedding motifs derived from historical stock market data into a forecasting model, we assessed whether such patterns could enhance predictive accuracy. The results indicate that motif embeddings do not significantly improve forecasting performance, aligning with the weak-form EMH. This hypothesis posits that historical price patterns cannot consistently predict future price movements. These findings suggest that financial markets may already incorporate the informational content of motifs, reducing their value as standalone predictors.

4.6. Discussion

Motif discovery uncovered several recurring temporal patterns in the financial time series, demonstrating the presence of locally repeated shapes even in volatile markets. Two selection methods—Binary Integer Programming (BIP) and a heuristic approach—were employed to filter and finalize these patterns. While BIP guarantees an optimal motif set with a heavier computation cost, the heuristic method offered a more scalable alternative. Ultimately, both approaches effectively highlighted non-trivial structural repetitions, facilitating a clearer understanding of the data and laying the groundwork for subsequent forecasting experiments. Despite successful motif discovery, integrating these motifs into a Transformer-based forecasting model did not yield the significant improvements initially anticipated. In simpler comparative experiments, classical statistical methods outperformed or remained competitive with the motif-enhanced Transformer approach. This outcome suggests that while motifs can provide insights into data structures, leveraging them directly within complex neural architectures, such as Transformers, may require more refined integration strategies or additional model adaptations.

Several potential factors could account for the observed forecast results:

- Transformers can struggle with certain forms of time series data, especially when

data exhibit strong non-stationary behavior or limited sample sizes. Although powerful in many sequence-modeling contexts, they may require extensive tuning or architectural modifications to handle real-world time series effectively. In this paper [55] , researchers reveal that a simple linear model, which directly learns from historical trends without the complex attention mechanism, consistently outperforms Transformers across various benchmark. This finding challenges the assumption that Transformers' success in NLP and computer vision readily carries over to time series forecasting, suggesting that they may not be the best fit for capturing long-range temporal dependencies.

- Financial time series often experience rapid regime shifts and market shocks which may diminish the consistency of learned motifs. Even motifs that appear frequently may not retain stable predictive power when the data distribution changes abruptly.
- The presence of a motif in a time series does not guarantee that each occurrence carries a uniform predictive signal. There may be no consistent linear or non-linear relationship between motifs and the target variable. This variability introduces noise or bias into the model, potentially offsetting the benefits of motif embeddings and limiting their contribution to improved forecasting accuracy.

5. CONCLUSION

This thesis introduces a novel framework for motif discovery in time series and demonstrates its practical applications in forecasting. It begins by establishing the motivation and significance of time series motif analysis and laying a solid theoretical foundation, covering key concepts like time series data mining, similarity measures, and transformer-based forecasting methods. It then focuses on how to identify motifs of varying lengths, comparing two distinct approaches—Binary Integer Programming and a heuristic—to balance accuracy and computational efficiency. These methods are then applied to real-world financial datasets for extracting motifs and integrating motif information into a forecasting model. After evaluating four different forecasting models, the results show that Transformers—with or without motif embeddings—underperformed simpler traditional baselines, underscoring the challenges of leveraging motifs within deep learning frameworks for financial time series forecasting.

The results demonstrate that motif discovery can effectively reveal recurring patterns within financial time series, offering valuable insight into how prices evolve. However, these discovered motifs did not yield significant gains in forecasting accuracy, indicating that once markets absorb observable structures, their predictive leverage tends to diminish. From an EMH standpoint, the limited impact of motif-based signals aligns with the notion that new information and obvious patterns are quickly priced in, reducing the potential for any predictive advantage. Despite these constraints on predictive utility, motif discovery remains a valuable analytical tool, offering deeper insights into structural behaviors, assisting with exploratory data analysis, identifying recurring segments, and illustrating how market dynamics may shift over time.

5.1. Further Research Directions

Future research can address how to better integrate motifs within complex models and tailor architectures to the unique dynamics of time series data. Below are several

possible directions:

- Refined Motif Enhanced Models : Future work could explore alternative or more adaptive ways of embedding motifs such as weighted embeddings or hierarchical architectures that selectively incorporate motif information only when it proves statistically beneficial.
- Preprocessing and Feature Engineering : Improved preprocessing (e.g., noise filtering, regime detection) or feature engineering (e.g., combining motifs with domain-driven indicators) may reveal hidden relationships between motif occurrences and the target variables.
- Incorporating Time Invariant and External Factors : Including time-invariant features (e.g., sector classifications or fundamental company metrics) within external variables (e.g., macroeconomic indicators, geopolitical events, or market sentiment) could enrich the forecasting framework. Such additions may provide the model with broader contextual information, helping it interpret the role of motifs within complex market dynamics more effectively.
- Applications Beyond Forecasting : Given that motif discovery holds promise for interpretability, clustering, and anomaly detection, future work could broaden its scope to non-forecasting applications. Real-time analytics in healthcare, fault detection in engineering, or behavior analysis in complex systems might particularly benefit from timely identification of recurring patterns.

In conclusion, while the proposed motif discovery framework can robustly identify repeated structures in time series, leveraging these motifs directly within a Transformer-based pipeline did not yield the anticipated performance gains for forecasting. Nonetheless, the advancements in motif detection and the accompanying insights into time series structure remain valuable contributions. By refining integration techniques, tailoring models to the unique dynamics of time series data, and extending applications to broader analytical tasks, future research can continue to unlock the potential of motif-driven insights in real-world settings.

REFERENCES

1. Tufte, E. R., *The Visual Display of Quantitative Information*, Graphics Press, Cheshire, Connecticut, 1983.

2. Chakrabarti, K., E. Keogh, S. Mehrotra and M. Pazzani, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases", *ACM Trans. Database Syst.*, Vol. 27, No. 2, p. 188–228, 2002.

3. Ralanamahatana, C. A., J. Lin, D. Gunopulos, E. Keogh, M. Vlachos and G. Das, *Mining Time Series Data*, pp. 1069–1103, Springer US, Boston, MA, 2005.

4. Zhong, S. and A. Mueen, "MASS: distance profile of a query over a time series", *Data Min. Knowl. Discov.*, Vol. 38, No. 3, pp. 1466—-1492, 2024.

5. Rodgers, J. L. and W. A. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient", *The American Statistician*, Vol. 42, No. 1, pp. 59–66, 1988.

6. Keogh, E. J. and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications", *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 285–289, Association for Computing Machinery, New York, NY, USA, 2000.

7. Stefan, A., V. Athitsos and G. Das, "The Move-Split-Merge Metric for Time Series", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 6, pp. 1425–1438, 2013.

8. Yeh, C.-C. M., Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen and E. Keogh, "Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View that Includes Motifs, Discords, and Shapelets", *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1317–1322, Barcelona, Spain, 2016.

9. Yeh, C.-C. M., N. Kavantzas and E. Keogh, "Matrix Profile VI: Meaningful Multidimensional Motif Discovery", *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 565–574, New Orleans, LA, 2017.

10. Linardi, M., Y. Zhu, T. Palpanas and E. Keogh, "Matrix Profile X: VALMOD - Scalable Discovery of Variable-Length Motifs in Data Series", *Proceedings of the 2018 International Conference on Management of Data*, pp. 1053—-1066, Association for Computing Machinery, New York, NY, USA, 2018.

11. Zhu, Y., Z. Zimmerman, N. Senobari, C.-C. M. Yeh, G. Funning, A. Mueen and E. Keogh, "Matrix Profile II: Exploiting a Novel Algorithm and GPUs to Break the One Hundred Million Barrier for Time Series Motifs and Joins", *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 739–748, Barcelona, Spain, 2016.

12. Zimmerman, Z., K. Kamgar, N. S. Senobari, B. Crites, G. Funning, P. Brisk and E. Keogh, "Matrix Profile XIV: Scaling Time Series Motif Discovery with GPUs to Break a Quintillion Pairwise Comparisons a Day and Beyond", *Proceedings of the ACM Symposium on Cloud Computing*, pp. 74—-86, Association for Computing Machinery, New York, NY, USA, 2019.

13. Lu, Y., R. Wu, A. Mueen, M. A. Zuluaga and E. Keogh, "Matrix Profile XXIV: Scaling Time Series Anomaly Detection to Trillions of Datapoints and Ultra-fast Arriving Data Streams", *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1173—-1182, Association for Computing Machinery, New York, NY, USA, 2022.

14. Alaee, S., K. Kamgar and E. Keogh, "Matrix Profile XXII: Exact Discovery of Time Series Motifs Under DTW", *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 900–905, Sorrento, Italy, 2020.

15. Kamgar, K., S. Gharghabi and E. Keogh, "Matrix Profile XV: Exploiting Time

Series Consensus Motifs to Find Structure in Time Series Sets", *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 1156–1161, Beijing, China, 2019.

16. Nakamura, T., M. Imamura, R. Mercer and E. Keogh, "MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives", *2020 IEEE International Conference on Data Mining (ICDM)*, pp. 1190–1195, Sorrento, Italy, 2020.

17. Tafazoli, S. and E. Keogh, *Matrix Profile XXVIII: Discovering Multi-Dimensional Time Series Anomalies with K of N Anomaly Detection†*, pp. 685–693, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2023.

18. Yeh, C.-C. M., N. Kavantzas and E. Keogh, "Matrix profile IV: Using Weakly Labeled Time Series to Predict Outcomes", *Proc. VLDB Endow.*, Vol. 10, No. 12, pp. 1802–1812, 2017.

19. Zhu, Y., A. Mueen and E. Keogh, "Matrix Profile IX: Admissible Time Series Motif Discovery With Missing Data", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 33, No. 6, pp. 2616–2626, 2021.

20. Gharghabi, S., Y. Ding, C.-C. M. Yeh, K. Kamgar, L. Ulanova and E. Keogh, "Matrix Profile VIII: Domain Agnostic Online Semantic Segmentation at Superhuman Performance Levels", *2017 IEEE International Conference on Data Mining (ICDM)*, pp. 117–126, New Orleans, LA, 2017.

21. Jensen, K. L., M. P. Styczynski, I. Rigoutsos and G. N. Stephanopoulos, "A Generic Motif Discovery Algorithm for Sequential Data", *Bioinformatics*, Vol. 22, No. 1, pp. 21–28, 2005.

22. Lawrence, C. E. and A. A. Reilly, "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopoly-

mer Sequences", *Proteins: Structure, Function, and Genetics*, Vol. 7, No. 1, pp. 41–51, 1990.

23. Bailey, T. L. and C. Elkan, "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization", *Machine Learning*, Vol. 21, No. 1–2, p. 51–80, 1995.

24. Lin, J., E. Keogh, S. Lonardi and B. Chiu, "A Symbolic Representation of Time Series, with Implications for Streaming Algorithms", *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, p. 2–11, Association for Computing Machinery, New York, NY, USA, 2003.

25. Gao, Y. and J. Lin, "HIME: Discovering Variable-length Motifs in Large-scale Time Series", *Knowledge and Information Systems*, Vol. 61, No. 1, p. 513–542, 2019.

26. Box, G. E. P., G. M. Jenkins, G. C. Reinsel and G. M. Ljung, *Time Series Analysis: Forecasting and Control*, John Wiley & Sons, Hoboken, NJ, 5th edn., 2015.

27. Lim, B., S. Arık, N. Loeff and T. Pfister, "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting", *International Journal of Forecasting*, Vol. 37, No. 4, pp. 1748–1764, 2021.

28. Shih, S.-Y., F.-K. Sun and H.-y. Lee, "Temporal Pattern Attention for Multivariate Time Series Forecasting", *Machine Learning*, Vol. 108, No. 8-9, pp. 1421–1441, 2019.

29. Akhtar, M. M., A. S. Zamani, S. Khan, A. S. A. Shatat, S. Dilshad and F. Samdani, "Stock market prediction based on statistical data using machine learning algorithms", *Journal of King Saud University - Science*, Vol. 34, No. 4, p. 101940, 2022.

30. Selvin, S., R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock Price Prediction Using LSTM, RNN and CNN-Sliding Window Model",

2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1643–1647, Udupi, India, 2017.

31. Patel, J., S. Shah, P. Thakkar and K. Kotecha, "Predicting Stock Market Index Using Fusion of Machine Learning Techniques", *Expert Systems with Applications*, Vol. 42, No. 4, pp. 2162–2172, 2015.

32. De Gooijer, J. G. and R. J. Hyndman, "25 Years of Time Series Forecasting", *International Journal of Forecasting*, Vol. 22, No. 3, pp. 443–473, 2006.

33. Kukreja, S., T. Kumar, V. Bharate, A. Purohit, A. Dasgupta and D. Guha, "Vector Databases and Vector Embeddings-Review", *2023 International Workshop on Artificial Intelligence and Image Processing (IWAIIP)*, pp. 231–236, Yogyakarta, Indonesia, 2023.

34. Mikolov, T., K. Chen, G. Corrado and J. Dean, "Efficient Estimation of Word Representations in Vector Space", ArXiv preprint arXiv:1301.3781[cs.CL], 2013.

35. Pennington, J., R. Socher and C. Manning, "GloVe: Global Vectors for Word Representation", *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Association for Computational Linguistics, Doha, Qatar, 2014.

36. Cer, D., Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. St. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, B. Strope and R. Kurzweil, "Universal Sentence Encoder for English", *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 169–174, Association for Computational Linguistics, Brussels, Belgium, 2018.

37. Kiros, R., Y. Zhu, R. Salakhutdinov, R. S. Zemel, A. Torralba, R. Urtasun and S. Fidler, "Skip-Thought Vectors", *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 2*, pp. 3294—-3302, MIT

Press, Cambridge, MA, USA, 2015.

38. Le, Q. and T. Mikolov, "Distributed Representations of Sentences and Documents", *Proceedings of the 31st International Conference on Machine Learning*, Vol. 32, pp. 1188–1196, PMLR, Bejing, China, 2014.

39. Simonyan, K. and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", ArXiv preprint arXiv:1409.1556[cs.CV], 2015.

40. He, K., X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, IEEE, Las Vegas, NV, USA, 2016.

41. Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger and I. Sutskever, "Learning Transferable Visual Models From Natural Language Supervision", ArXiv preprint arXiv:2103.00020[cs.CV], 2021.

42. PyTorch, "Embedding Layer", `https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html`, accessed on December 11, 2024.

43. Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need", ArXiv preprint arXiv:1706.03762[cs.CL], 2023.

44. Bailey, R. E., *The Economics of Financial Markets*, Cambridge University Press, Cambridge, UK, 2005.

45. Mishkin, F. S., *The Economics of Money, Banking, and Financial Markets*, Pearson, New York, NY, 2021.

46. Thompson, O., "Factors Influencing Stock Market Prices: A Comprehensive Analysis", *Academy of Accounting and Financial Studies Journal*, Vol. 28, No. 4, pp.

1–2, 2024.

47. Pedersen, L. H., *Efficiently Inefficient: How Smart Money Invests and Market Prices Are Determined*, Princeton University Press, Princeton, NJ, 2015.

48. Malkiel, B. G. and E. F. Fama, "Efficient Capital Markets: A Review of Theory and Empirical Work", *The Journal of Finance*, Vol. 25, No. 2, pp. 383–417, 1970.

49. Malkiel, B. G., *A Random Walk Down Wall Street*, W. W. Norton & Company, New York, NY, 12th edn., 2019.

50. Shiller, R. J., "Do Stock Prices Move Too Much to Be Justified by Subsequent Changes in Dividends?", *American Economic Review*, Vol. 71, No. 3, pp. 421–436, 1981.

51. Shefrin, H., *A Behavioral Approach to Asset Pricing*, Academic Press advanced finance series, Elsevier Academic Press, Burlington, MA, 2005.

52. Lo, A. W., "The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective", *Journal of Portfolio Management*, Vol. 30, No. 5, pp. 15–29, 2004.

53. Bishop, C. M., *Pattern Recognition and Machine Learning*, Springer, New York, NY, 1st edn., 2006.

54. "Investing", Online Database, `https://www.investing.com/indices/ise-banks`, accessed on December 11, 2024.

55. Zeng, A., M. Chen, L. Zhang and Q. Xu, "Are Transformers Effective for Time Series Forecasting?", ArXiv preprint arXiv:2205.13504[cs.AI], 2022.

APPENDIX A: CODE REPOSITORY

The source code developed for this thesis, including scripts for data processing, model implementation, and result generation, is available on GitHub. You can access it using the following link: https://github.com/aycabayraktar/Thesis/tree/main

The repository includes:

- Google Colab notebook containing the complete implementation of the methodology, including:
  - Preprocessing scripts for cleaning and normalizing time series data.
  - Code for motif extraction and embedding into Transformer models.
  - Steps for result visualization and analysis.

APPENDIX B: COPYRIGHT AND LICENSE INFORMATION

2.1 was taken from the paper [43]. The paper is distributed by arXiv.org under a non-exclusive license granted to arXiv for distribution. The figure has been used with proper attribution and in compliance with the licensing conditions.