Doctoral Thesis

A Study on Multi-Granularity Representation Learning of Time Series Data

Chengyang YE

January 2024

Social Informatics Course
Graduate School of Informatics
Kyoto University

Doctoral Thesis
submitted to Social Informatics Course,
Graduate School of Informatics,
Kyoto University
in partial fulfillment of the requirements for the degree of
DOCTOR of INFORMATICS

Thesis Committee: Qiang Ma, Professor
Takayuki Ito, Professor
Takayuki Kanda, Professor
Shinsuke Mori, Professor

A Study on Multi-Granularity Representation Learning of Time Series Data

Chengyang YE

Abstract

The significant advancement of the Internet and widespread use of sensors has driven the remarkable development of time series, engendering complex datasets of varied granularities and complexities. Time series plays a crucial role in various domains such as healthcare and finance, each having its unique set of applications and importance.

Traditional time series analysis methods encompass a variety of statistical techniques that focus on the extraction of meaningful statistics and characteristics from data points collected over time. These methods are largely built on statistical models that assume a certain degree of stationarity, linearity, and statistical properties that remain constant over time, which does not align with the often non-stationary and nonlinear nature of real-world data. This disconnect can lead to oversimplified models that fail to capture complex dynamics, especially when dealing with multivariate data. Consequently, these limitations have paved the way for the adoption of deep learning techniques that offer greater flexibility for time series analysis.

Deep learning has risen attention in time series analysis due to its ability to model complex patterns and relationships. Neural networks, particularly those designed for sequence data, are adept at recognizing and remembering various features, making them ideally suited for forecasting and anomaly detection tasks. In deep learning, representation learning emerges as a crucial component, playing an significant role in time series analysis and downstream tasks. Representation learning facilitates the extraction and transformation of raw time series data into structured and meaningful vectors, enabling effective data analysis, pattern recognition, and decision-making.

A considerable gap exists in representation learning of time series data. Research has often been characterized by the direct adoption of methodologies in CV

and NLP. These approaches, although sometimes effective, may not fully account for the internal correlations and sequential patterns embedded within time series data. Additionally, another critical aspect is the multi-granularity nature of time series data, as understanding the interactions and dependencies across different granularities can reveal deeper, more comprehensive representation, leading to more accurate and reliable predictions and decisions.

As one of the reasonable solutions, this research studies time series representation learning models with different granularities. Granularity refers to the level of detail or scale of the data. For instance, daily stock prices and monthly stock prices have different granularities. Multi-Granularity Representation Learning aims to learn representations across different scales, capturing both short-term patterns (from finer granularities) and long-term trends (from coarser granularities). The models are designed around the distinct features inherent in time series data, with an novel unsupervised learning approach, making them particularly well-suited for representation learning of time series data.

The representation learning models involves four modules: 1. **timestamp-level representation learning** for fine-grained representation, 2.1. **segment-level representation learning** for coarse-grained representation, 2.2. **streaming version of segment-level representation learning** for streaming time series representation; and 3. **cross-granularity representation learning** to combine the advantages of multi-granularity of representation. The four modules are illustrated in detail as follows:

1 **timestamp-level representation learning**: Timestamp-level representation learning focus on fine-grained representation, where we delve into fine-grained nuances of the data. Fine-grained representation learning is designed to capture subtle patterns and minute fluctuations over time, which can be critical for sensitive applications that require high-resolution insights. We introduce a specially designed local binary pattern method to the self-attention mechanism to improve the representation performance of modeling in terms of local information. Meanwhile, a novel unsupervised approach is designed to training the representation learning model. Experiments of classification and regression have been implemented to verify the effectiveness of our proposed approach in tasks that need fine-grained features.

2.1 **segment-level representation learning**: For segment-level representation of time series, another unsupervised representation learning model is proposed to consider the feature of time series subseries. The aim is to understand and encapsulate broader trends and shifts over larger intervals of time, yielding a coarse-grained representation of the time series. This form of representation is beneficial for applications where long-term trends and patterns are of interest, such as retrieval task. In this study, the covariance calculated by the Gaussian process is introduced to the self-attention mechanism, capturing relationship features of subseries. Experiments of retrieval verified the effectiveness of our proposed algorithm in coarse-grained representation of time series.

2.2 **streaming version of segment-level representation learning**: To showcase the versatility and robustness of our model, we extend its application to the domain of streaming time series data. This extension improves the model's practical significance and application value, enabling it can deal with the issue of time series in real-world data processing and analysis. In this extension, we redesign the algorithm, ensuring it is adept at handling continuous, real-time data streams, thereby broadening its applicability and efficacy beyond static time series data, and making it a versatile tool for diverse data environments and application contexts. Experiments in streaming time series data verified the effectiveness of expanded method.

3 **cross-granularity representation learning**: To Bridge representation learning models with different granularities, we introduce a novel cross-granularity representation model. This model is adept at integrating both fine-grained and coarse-grained representations, leveraging the strengths of each to provide a more holistic understanding of time series data. This comprehensive integration ensures an enhanced accuracy in representation learning, making it a significant tool for various datasets of time series and analysis tasks.

Our research has proved that multi-granularity representation learning has emerged as a transformative approach in real-world fields , such as healthcare. In healthcare, patient data is inherently multi-granular. Multi-granularity representation learning can uncover subtle patterns in patient data by analyzing it

across different temporal resolutions. For instance, while short-term fluctuations in heart rate might indicate immediate stress responses, longer-term trends can signal the development of a chronic condition. By learning representations at multi-granularities, models can provide a more comprehensive view of a patient's health, enhance the early detection of diseases. Our model stands as a novel approach in the domain of time series representation learning, highlighting the potential for innovation in this field.

Given the increasing complexity of data and the ongoing advances in machine learning methodologies, the proposed approach can also be adapted in cross-domain data sources. Research will focus on cross-domain transfer learning where a model trained on one domain (e.g., healthcare) can adapt to another (e.g., finance) by leveraging shared multi-granularity representations. We will focus on deploying the application and extending the application in cross-domain data sources for future work. Besides, custom granularity levels and causal inference could also become the necessary area of multi-granularity research.

Keywords: Representation learning, Time Series, Multi-Granularity, Timestamp-Level, Segment-Level, Cross-Granularity

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

This chapter introduces the background and overview of this thesis. First, the background of this dissertation is provided from both social and technical perspectives, including the introduction of time series data, representation learning, downstream tasks and multi-granularity representation of time series. Second, the overview of this research is illustrated. Following that, we present topics included in this thesis and their motivations. The structure of this dissertation is shown in the last section.

1.1 Background

1.1.1 Time Series

A time series is a sequence of numerical data points in successive order, typically occurring at uniformly spaced time intervals. in brief, it is a series of data points listed in time order. Examples of time series data include daily stock prices, healthcare, environmental sensing, and energy monitoring.

Delving deeper into the characteristics of time series, one can identify several integral components. There is the trend, which represents the long-term movement in data; seasonality, which refers to the predictable fluctuations that occur in regular intervals; cyclic patterns, long-term patterns without a fixed period;

and the irregular component that signifies the unpredictable variance left after extracting other patterns.

The significance of time series in contemporary research is vast and multi-faceted. One of the primary uses of time series analysis is in classification, which allows categorizing and classifying various data segments based on historical data patterns. Beyond classification, understanding time series data affords a clearer insight into underlying patterns, facilitating a deeper grasp of trends and data anomalies. This depth of understanding is invaluable for decision-making processes in various fields. For instance, businesses can optimize inventory management or make financial investments, while in healthcare, continuous monitoring of patient vitals can be analyzed for predicting health risks. In specialized domains like cyber security, time series plays a role in anomaly detection, pointing out system irregularities that might indicate potential breaches. Similarly, engineers can refine signal processing to filter noise and extract meaningful insights. Lastly, environmental scientists and climatologists harness time series data to study weather patterns and environmental shifts, aiding in understanding potential future scenarios like natural disasters or the long-term impacts of climate change. With the advent of the big data era, the value of time series data, coupled with advanced analytical techniques, has surged, reinforcing its pivotal role in current research and practical applications.

While images and natural language data have dominated the spotlight in popular machine learning research, the significance of time series data should not be underestimated. In the early days of machine learning, the emphasis was largely on areas where the most noticeable consumer impact could be made. The visual nature of images and the broad applicability of natural language processing meant that these domains quickly became poster children for the power of machine learning, capturing imaginations and research funding alike.

However, as technology evolves and diversifies, the value of time series data is becoming increasingly apparent. Time series data is everywhere – from financial markets, where stock prices fluctuate over time, to healthcare, where patient vitals are recorded continuously. It provides a rich source of information that, when mined correctly, can reveal complex patterns, trends, and relationships that other types of data might miss.

The contemporary tech landscape is witnessing an expansion of interconnected devices, commonly known as the Internet of Things (IoT). These devices contin-

uously generate vast amounts of time series data. Efficiently analyzing this data can lead to optimized operations, predictive maintenance, and improved user experiences. Furthermore, as industries move towards automation and real-time decision making, the need for accurate and timely insights derived from time series data grows. It offers a dynamic perspective, capturing the evolution of processes, behaviors, and systems over time. In contrast to static images or a singular piece of text, time series data offers a continuous, flowing perspective on the world, reflecting its inherent variability and change.

In conclusion, while images and language data have undeniably been at the forefront of machine learning research, the rising prominence and potential of time series data is undeniable. As technological landscapes and applications evolve, we can expect time series analysis to play an ever-increasing role in shaping our understanding and optimization of the world around us.

1.1.2 Representation Learning

Representation learning is a technique within machine learning that allows systems to automatically identify and extract useful features or representations from raw data, eliminating the need for manual feature engineering.

In traditional machine learning, feature engineering often requires manual intervention, where domain experts define and extract the important features from raw data. This process can be time-consuming and may not always capture the most important or subtle patterns in the data.

Representation learning, on the other hand, aims to automate this process. By training on large amounts of data, a system can learn to represent data in a way that makes it easier to perform tasks like classification, regression, clustering, and more. The learned representations often capture intricate patterns, hierarchies, and structures in the data.

At its core, representation learning tries to find a way to transform data, often high-dimensional, into a format or space where essential patterns or features are more easily discernible or interpret-able. This transformation can greatly assist downstream tasks, such as classification or regression. For instance, in the realm of deep learning, neural networks, especially convolutional neural networks (CNNs) for images or recurrent neural networks (RNNs) for sequences, are adept at learning hierarchical representations from raw data. The initial layers capture

low-level features, such as edges in images, and as one moves deeper into the network, more abstract and higher-level representations are formed. This learned representation often carries much of the useful information about the original data, making subsequent tasks more tractable. The ultimate goal of representation learning is to expose the underlying structure or factors of variation in the data, ensuring that learned representations are not just compact, but also meaningful and useful for the task at hand.

Therefore, the quality of the learned representation is crucial because it directly affects the performance of downstream tasks With the rise of unsupervised and semi-supervised learning techniques, representation learning has become even more essential, as it can exploit unlabeled data to learn powerful feature representations. This is particularly important in domains where labeled data is scarce or expensive to obtain.

In conclusion, representation learning seeks to automate the process of finding the most useful data representations, often reducing the need for manual feature engineering and enabling models to automatically capture intricate patterns in data.

1.1.3 Downstream Tasks

In the context of time series, representation learning can finding ways to represent sequences of data in ways that make them useful for downstream tasks. Here are some of the typical downstream tasks that can benefit from good representations of time series data.

Time series classification involves assigning a predefined label to a given time series based on its temporal patterns. For instance, in medical diagnostics, a sequence of heart rate data might be categorized as 'normal' or 'arrhythmic'. When representation learning is applied to time series classification, it can capture salient features in the data that directly relate to the classes of interest. This often results in more accurate and robust classification models. Moreover, it allows models to generalize better to unseen data by focusing on the most significant patterns rather than noise or irrelevant details.

Time series regression aims to predict a continuous value, either forecasting future points or filling in missing data within a series. An example would be forecasting stock prices for the upcoming week. For regression tasks, representa-

tion learning helps in emphasizing the underlying trends and patterns essential for making accurate predictions. By converting the raw data into a compact and informative representation, regression models can focus on the primary dynamics of the time series, leading to more precise forecasts. The models used for time series regression can range from simple linear regression for a single predictor to complex models that can handle seasonality, trends, and cycles in the presence of multiple influencing factors.

Time series retrieval is about identifying similar time series within a large dataset. For instance, in anomaly detection, one would look for patterns that significantly deviate from typical series. Representation learning aids in extracting a consistent and compact fingerprint or signature for each time series. Such representations make it computationally efficient to compare and retrieve similar series from a vast database. Moreover, they ensure that the similarity is based on meaningful patterns in the data rather than superficial or noisy details.

Beyond classification and regression, time series data empower a variety of other downstream tasks such as anomaly detection, which seeks to identify outlying or unusual data points indicative of errors, fraud, or novel events; clustering, where time series are grouped based on similarity without pre-labeled categories; and motif discovery, where frequently occurring patterns are identified. These tasks are foundational in operationalizing time series data across disciplines, aiding in decision-making processes from monitoring industrial equipment health to anticipating market trends and beyond.

In summary, representation learning acts as a powerful tool in transforming raw time series data into more digestible and meaningful formats, amplifying the efficiency and accuracy of various downstream tasks.

1.1.4 Multi-Granularity Representation Learning

Multi-Granularity Representation Learning of Time Series is an important topic in the field of time series analysis because phenomena recorded as time series data often exhibit relevant behaviors at various time granularities. Granularity in time series refers to the level of detail or scale of the data. For instance, financial markets might show volatilises on minute-level data, cyclical behaviors on daily data, and long-term trends on yearly scales. Similarly, health monitoring data can exhibit vital sign fluctuations over seconds, pattern changes over

hours (e.g., sleep cycles), and health progression over months or years. Learning representations at different granularities can help to identify and leverage these patterns for forecasting, anomaly detection, and other analyses.

In multi-granularity representation learning of time series, many models are developed. These models can be broadly categorized into the following three types.

- **Extract features at each granularity independently**: Separate models or feature extractors are applied to different granular representations of the data. For example, one model may analyze annual sales data while another looks at daily sales fluctuations. The challenge here is integrating these features in a way that allows for effective decision-making.

- **Learn hierarchical representations**: This involves learning representations that inherently capture information across scales. Hierarchical models, like certain types of neural networks, can process data in a way that allows them to learn low-level details and high-level abstractions simultaneously.

- **Incorporate multi-scale features into a single model**: In this case, a model is designed to take inputs or features from multiple time scales and combine them internally to make predictions or classify data points. Techniques such as wavelet transforms or multi-resolution analysis are commonly used to create features that capture information across scales.

Regardless of the method used, the goal of multi-granularity representation learning is to create a model that is sensitive to the relevant temporal patterns across different scales and can thus perform better on a given task. The integration of multi-scale information often leads to more robust and accurate models, especially in complex systems where different processes operate and interact at different temporal resolutions.

In practice, multi-granularity representation learning requires careful consideration of the scales that are relevant to the problem at hand, and the appropriate algorithms that can handle multi-scale data effectively. It is a challenging area but one that offers considerable promise for enhancing our understanding and prediction of time series data across a wide range of applications.

1.1.5 Semantic Features

In computer vision, "semantic" pertains to the interpretation and understanding of visual content in a way that aligns with human perception and cognitive categories. Within computer vision, semantic understanding is often categorized into basic-level semantics and high-level semantics.

Basic-level semantics involves recognizing and categorizing objects in an image or video at a general level. For example, in an image containing various animals, a basic-level semantic task would be labels to distinguish each animals. This is often achieved through tasks like object detection and semantic segmentation. High-level semantics goes a step further by not only recognizing objects but also interpreting their interactions, contexts, and the overall scene. It involves a deeper level of understanding, such as inferring emotions from facial expressions.

In the context of time series analysis, basic-level semantics and high-level semantics can be distinctly defined based on the depth and scope of the information they represent.

Basic-level semantics in time series is about characterizing the standalone information of an object or a segment within the series. This means that each point or segment is analyzed in isolation to determine its attributes or state. For instance, if the time series data represent temperature readings over time, basic-level semantics would involve categorizing each reading as 'high', 'medium', or 'low' temperature without considering the broader context of surrounding readings.

High-level semantics, concerns the representation of an object or segment in relation to its neighbors or the overall structure of the series. This relational understanding is crucial when the significance of a single data point is only fully revealed through its interaction with others. In the temperature example, high-level semantics would not only categorize individual temperatures but also interpret patterns such as a sudden drop in temperature following a consistent rise. Such patterns could indicate a cold front in a weather-related time series or signify specific events in a process when applied to industrial monitoring.

In sum, basic-level semantics in time series capture the essence of each individual data point, while high-level semantics encapsulate the dynamics and interactions among data points, providing a more integrated and contextual interpretation of the time series as a whole.

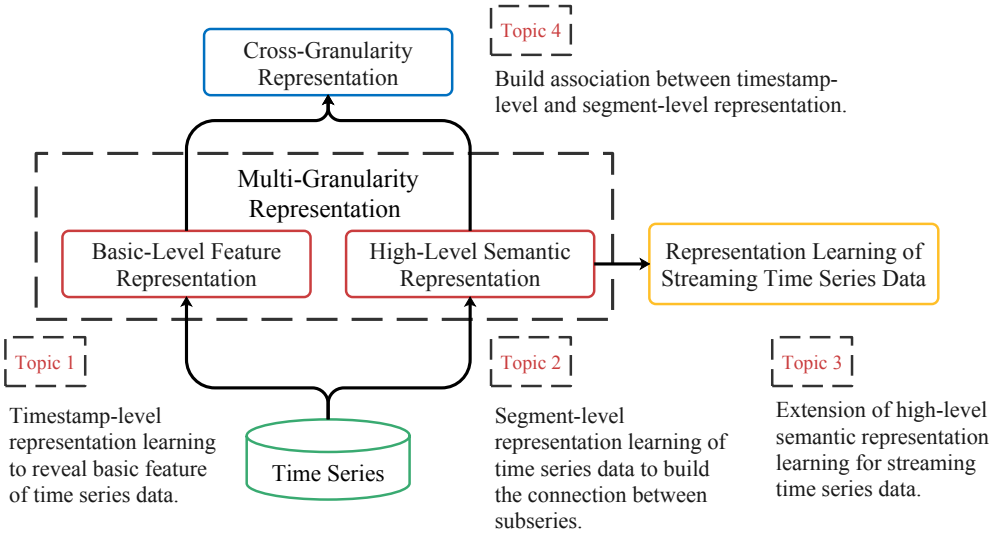


Figure 1.1. Framework of Doctoral Thesis.

1.2 Overview of the Research

The comprehensive architecture of our approach to understanding time series data is detailed in Figure 1.1. The figure illustrates a multi-granularity strategy for representation learning of time series data, which is pivotal for capturing the intricacies of temporal data at varying levels of detail. Our first approach focuses on the timestamp level, where we delve into fine-grained nuances of the data. This fine-grained representation learning is designed to capture subtle patterns and minute fluctuations over time, which can be critical for sensitive applications that require high-resolution insights.

Conversely, our second approach shifts the perspective to a more macroscopic view, concentrating on the segment level. Here, the aim is to understand and encapsulate broader trends and shifts over larger intervals of time, yielding a coarse-grained representation of the time series. This form of representation is beneficial for applications where long-term trends and patterns are of interest.

Bridging these two perspectives, we introduce a novel cross-granularity representation model. This model is adept at integrating both fine-grained and coarse-grained representations, leveraging the strengths of each to provide a more holistic understanding of time series data. Such integration is especially crucial when addressing complex problems that require an understanding of both immediate details and longer-term patterns.

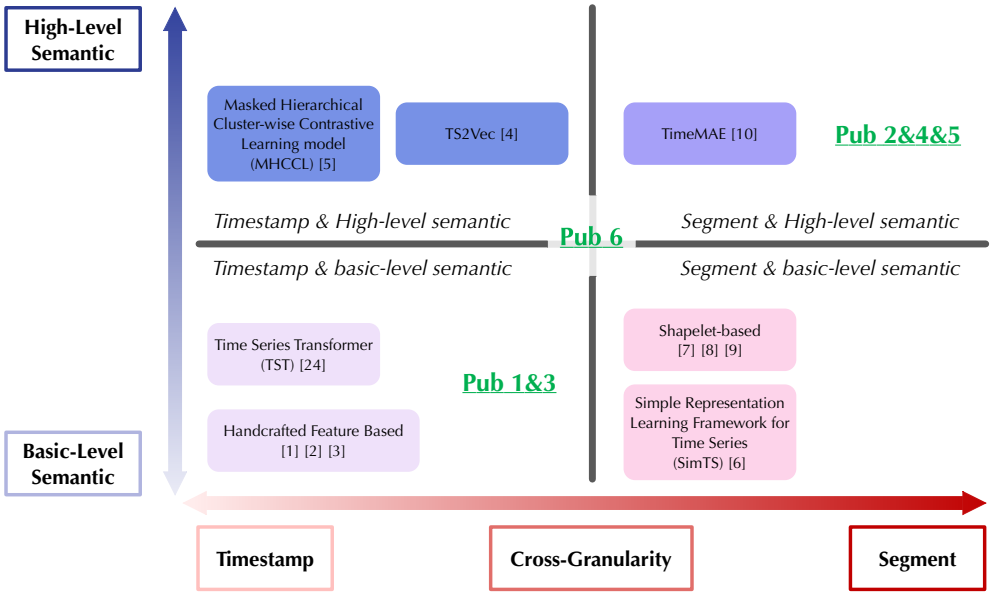


Figure 1.2. Relative position of the publications with the past studies (the shown publications 1-6 are listed in the Chapter 1.3.5).

To showcase the versatility and robustness of our model, we extend its application to the domain of streaming time series data. Streaming data, characterized by its continuous and real-time nature, presents unique challenges such as the need for timely processing and the ability to adapt to evolving patterns. Our model's extension to this domain demonstrates its capability to not only handle static time series datasets but also dynamically adapt to the ever-changing landscape of streaming data. This extension confirms the model's broad applicability and potential for real-world impact across various industries where real-time data analysis is paramount.

The studies in representation learning of time series can be generally categorized by two quadrants from the perspective of granularity and semantic, i.e. ether based on timestamp- or segment-level representation , based on the semantic of basic- or high-level, as illustrated in Figure 1.2.

In timestamp-level representation learning, the model represents each timestamp for time series; it is the most traditional idea for representation learning of time series and very complex. For example, the handcrafted feature based methods [1] [2] [3] in time series representation are traditional techniques that involve the manual selection and construction of features based on domain knowl-

edge and heuristic understanding of the data. Most of these handcrafted feature are based on the basic-level semantic. The process typically starts with domain experts who identify relevant features that capture the essential characteristics of the time series. These features can be statistical, such as mean, variance, skewness, and kurtosis, which describe the distribution of values within a window of the time series. These techniques are positioned at the bottom left of the Figure 1.2. As for timestamp representation learning with high-level semantic, TS2Vec [4] is a typical algorithm, which is placed top left of the Figure 1.2. TS2Vec has been recently proposed as a universal framework for learning time series representations by hierarchically performing contrastive learning over augmented contextual information. Another typical algorithm is MHCCL [5], a Masked Hierarchical Cluster-wise Contrastive Learning model, which exploits semantic information obtained from the hierarchical structure consisting of multiple latent partitions for multivariate time series. Although timestamp-level representation learning can achieve superior results in time series forecasting and anomaly detection tasks, such algorithms still have limitations. Particularly, they are not intended to represent the state of subseries and cannot be applied to certain tasks like data retrieval.

On the other hand, segment-level representation learning approaches consider learns temporal patterns from time series to generate segment-level representations, which help to develop dependencies between multivariate time series, such as Simple Representation Learning Framework for Time Series algorithm (SimTS) [6]. For the basic-level semantic based segment representation, a line of studies [7] [8] [9] considered to construct shapelet to represent time subseries. These Shapelet-based methods are focused on identification of discriminant subsequences in time series data, which can be useful for tasks such as classification and anomaly detection. Another line of segment-level representation learning methods are based on the high-level semantic features, such as TimeMAE [10]. The distinct characteristics of these methods lie in processing each time series into a sequence of non-overlapping sub-series via window-slicing partitioning, followed by random masking strategies over the semantic units of localized subseries. Segment-level approaches are placed to the right of Figure 1.2, within which the high-level semantic studies are placed at the top, and the basic-level semantic ones are positioned at the bottom.

1.3 Research Issues

Since detailed motivations will be explained in the following chapters respectively, in this section, we briefly introduce the motivation, target and approach of each task.

1.3.1 Timestamp-level Representation Learning

Representation learning of multivariate time series is a crucial and complex task that offers valuable insights for numerous applications, including time series classification, trend analysis, and regression. Unsupervised learning approaches are often favored in practical scenarios due to the limited availability of labeled data. However, most existing studies focus more on the global information of time series and ignore the local information, especially the representation learning based on the self-attention mechanism. This affects representation performance and may lead to the failure of downstream tasks. This study proposed an unsupervised representation learning model for multivariate time series by comprehensively considering multivariate time series data's global and local information. Specifically, a specially designed local binary pattern (LBP) method for multivariate time series (multivariate LBP) is introduced to the self-attention mechanism to improve the representation performance of modeling in terms of local information. Additionally, we propose a novel unsupervised approach for learning multivariate time series representations. The experimental results demonstrate significant advantages of our model over other representation learning methods and can be well applied in various downstream tasks.

1.3.2 Segment-level Representation Learning

Representation learning is a crucial and complex task for multivariate time series data analysis, with a wide range of applications including trend analysis, time series data search, and forecasting. In practice, unsupervised learning is strongly preferred owing to sparse labeling. However, most existing studies focus on the representation of individual subseries without considering relationships between different subseries. In certain scenarios, this may lead to downstream task failures. Here, an unsupervised representation learning model is proposed for multivariate time series that considers the semantic relationship among subseries of time series.

Specifically, the covariance calculated by the Gaussian process (GP) is introduced to the self-attention mechanism, capturing relationship features of the subseries. Additionally, a novel unsupervised method is designed to learn the representation of multivariate time series. To address the challenges of variable lengths of input subseries, a temporal pyramid pooling (TPP) method is applied to construct input vectors with equal length. The experimental results show that our model has substantial advantages compared with other representation learning models. We conducted experiments on the proposed algorithm and baseline algorithms in two downstream tasks: classification and retrieval. In classification task, the proposed model demonstrated the best performance on seven of ten datasets, achieving an average accuracy of 76%. In retrieval task, the proposed algorithm achieved the best performance under different datasets and hidden sizes. The result of ablation study also demonstrates significance of semantic relationship in multivariate time series representation learning.

1.3.3 Representation Learning for Streaming Time Series

Representation learning of time series is common in tasks like data mining and improves performance in downstream tasks. However, existing methods aren't appropriate for streaming time series due to two main limitations: first, The efficiency of representation learning methods can be a concern when dealing with streaming time series. Secondly, most of representation learning are designed for timestamp-level representation. They cannot reveal the semantic information in time series, which further reduces the efficiency and effectiveness of representation learning of streaming time series. This study introduces an unsupervised method tailored for streaming time series, considering semantic information. Specifically, it integrates recursive covariance estimation into a simplified transformer structure, PoolFormer, to enhance efficiency and reveal real-time semantic information. In addition, a novel unsupervised method is designed to learning the representation of streaming time series. The experiments show that this method outperforms other representation methods.

1.3.4 Cross-Granularity Representation Learning

Representation learning is crucial in the analyzing of time series data and has high practical value across a wide range of applications, including trend analysis,

time series data retrieval, and forecasting. In practice, data confusion is a significant issue as it can considerably impact the effectiveness and accuracy of data analysis, machine learning models, and decision-making processes. In general, previous studies did not consider the variability at various levels of granularity, thus resulting in inadequate information utilization, which further exacerbated the issue of data confusion. This study proposes an unsupervised framework to realize multi-granularity representation learning for time series. Specifically, we employed a cross-granularity transformer to develop an association between fine- and coarse-grained representations. Furthermore, we introduced a retrieval task as an unsupervised training task in representation learning. Moreover, a novel loss function was designed to obtain the comprehensive multi-granularity representation of time series. Experimental results revealed that the proposed framework exhibits significant advantages over alternative representation learning models.

1.3.5 Publications

By conducting the project, the following works had been published or submitted:

- Publication 1: "LBP4MTS: Local Binary Pattern-Based Unsupervised Representation Learning of Multivariate Time Series. ", IEEE Access. (DOI: 10.1109/ACCESS.2023.3327015)

- Publication 2: "Semantic Relationship-Based Unsupervised Representation Learning of Multivariate Time Series." IEICE Transactions on Information and System.

- Publication 3: "TS2V: A Transformer-Based Siamese Network for Representation Learning of Univariate Time-Series Data.", CSCWD 2022.

- Publication 4: "GP-HLS: Gaussian Process-Based Unsupervised High-Level Semantics Representation Learning of Multivariate Time Series", DASFAA 2023.

- Publication 5: "Unsupervised Representation Learning with Semantic of Streaming Time Series", WISE 2023.

- Publication 6: "Multi-Granularity Framework for Unsupervised Representation Learning of Time Series", arXiv.

1.4 Thesis Structure

The structure of this thesis is as follows. In Chapter 2, a review of related works, especially some techniques that will be used in the following works, will be described. Chapter 3 introduces topic of timestamp-level representation learning of time series. Meanwhile, a novel method based on local binary patter is considered to improve the performance of algorithm. Chapter 4 is about representation learning of time series under the segment-level. And representing of high-level semantic is introduced to representation learning method. In Chapter 5, we expand our high-level semantic-based segment-level representation learning approach to streaming time series. Chapter 6 introduces the task of multi-granularity framework for unsupervised representation learning of time series. Finally, Chapter 7 draws a conclusion of this thesis and has a discussion about the future researches.

CHAPTER 2

Technical Preliminaries

In this chapter, we will introduce some technical background we have used in our four researches.

2.1 Representation learning of Time Series

The representation learning of time series data has become a topic of considerable research interest. Most models aim to discover spatial–temporal dependencies in data. According to the representation granularity, there are three types of representation learning for time series: the timestamp-level, the instance-level, and the segment-level. The differences between these three types are shown in Figure. 2.1.

In timestamp-level representation learning, the model represents each timestamp for time series; it is the most traditional idea for representation learning of time series and very complex. It focuses more on the relationship between the different dimensions of time series; an example of a model that uses such type of representation learning is TS2Vec [4]. TS2Vec has been recently proposed as a universal framework for learning time series representations by hierarchically performing contrastive learning over augmented contextual information. Although timestamp-level representation learning can achieve superior results in time series forecasting and anomaly detection tasks, such algorithms still have limitations.

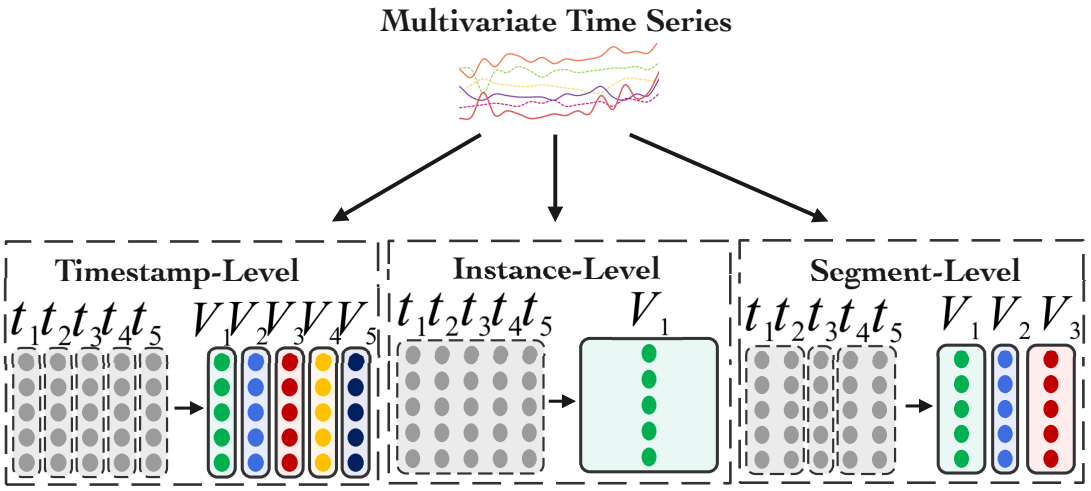


Figure 2.1. Main differences among three types of representation learning of multivariate time series.

Particularly, they are not intended to represent the state of subseries and cannot be applied to certain tasks like data retrieval.

Many studies have also focused on learning instance-level representations, which describe the entire segment of the input time series and have shown excellent performance in clustering and classification tasks [11]. In addition, recent works have employed contrastive loss to learn the inherent structure of time series. Nevertheless, they also have certain limitations. Instance-level representations may not be suitable for fine-grained forecasting models, which must infer the target in a specific subseries.

The segment-level representation of time series combines the advantages of timestamp-level and instance-level representation learning. It is somewhere in the middle of the timestamp-level and instance-level representation at the level of granularity, such as a scalable time series pre-training model SETP [12]. This model learns temporal patterns from long-term multivariate time series to generate segment-level representations, which help to develop dependencies between multivariate time series. A problem with these algorithms is the segmentation rule; this model divides multivariate time series data into subseries using a regular sliding window. In this manner, the subseries is random without any semantic information and relationship between different subseries. This may also lead to confusion in the representation of results. The representation of these subseries cannot be used in tasks that require semantic information, such as data retrieval.

In other words, none of these segment-based representation learning methods can learn high-level semantic information in time series. This is the main issue that the proposed model can address in this study.

2.1.1 Contrastive Representation architecture

contrastive learning has been introduced into this aspect of time series analysis [13]. Constructing positive and negative data pairs achieves unsupervised representation learning of unlabeled time series data. On this basis, triplet loss is further combined with a CNN with dilation [14] to tackle long time series data. This approach is fairly easy to implement and only requires distinguishing the main features.

Studies have also been devoted to applying data augmentation to raw data inputs in contrastive representation learning [15]. These models tend to construct the input view of time series data with some designed embedding methods and learn the representation of these input views by contrastive learning. These models employ the novel idea that constructing suitable time series embedding vectors as input could increase the learning performance of the model in representation learning.

The core of contrastive learning is the loss function, which meticulously adjusts the distances between embeddings. It rewards the encoder when it brings together embeddings from positive pairs and penalizes it when positive pairs are distant or negative pairs are too close. This is usually achieved using a contrastive loss function, such as the InfoNCE loss, which has been especially popular in the literature for its effectiveness.

In some cases, the representations obtained directly from the encoder are not used for downstream tasks. Instead, they are passed through an additional neural network component known as a projection head. This head further processes the representations and is only used during training to help stabilize the learning process. The projection head's outputs are used in the loss calculation, and upon completion of the training, it is discarded, with the encoder's outputs serving as the final learned representations.

These learned embeddings can be remarkably powerful, encapsulating the essential features of the data while being invariant to the superficial variations introduced by augmentation. When applied to time series, this approach can be

particularly beneficial, as it can distill complex, time-dependent patterns into robust representations. These are useful for a multitude of tasks, including anomaly detection, forecasting, and classification, across a variety of domains such as healthcare monitoring, financial trend analysis, and industrial equipment diagnostics. The appeal of contrastive representation learning lies in its versatility and the quality of the learned features, which are often superior to those obtained through traditional unsupervised learning methods.

2.1.2 Generative Representation architecture

Generative representation learning is an approach that learns to capture and understand the data's underlying probability distribution, allowing the model not only to generate new data points similar to the ones it has been trained on but also to develop a rich representation of the input data. This form of learning is typically associated with models like Generative Adversarial Networks (GANs) [16] , Variational Autoencoders (VAEs) [17], and other autoencoder variants [18].

The key idea behind generative representation learning is to force the model to understand the structure and distribution of the data so well that it can effectively generate new instances of the data. Through this process, the model learns a representation that can be used to infer properties of unseen data points or to complete or denoise partial data.

In a typical setup for VAEs, an encoder network maps the input data into a lower-dimensional latent space, which represents a compressed knowledge of the data. The latent space is designed to follow a probability distribution (often a Gaussian), ensuring that the latent variables capture the stochastic nature of the data. A decoder network then maps these latent representations back to the high-dimensional space, aiming to reconstruct the original data. The model is trained by optimizing a combination of a reconstruction loss (ensuring the output closely matches the input) and a regularization term (which keeps the latent space well-organized and ensures that it follows the prescribed distribution).

GANs take a different approach, consisting of two networks: a generator that creates data and a discriminator that evaluates it. The generator produces new data instances from latent space representations, while the discriminator assesses whether the generated data is "real" (from the actual dataset) or "fake" (produced by the generator). Through their adversarial process, both networks improve

iteratively, with the generator learning to create increasingly authentic-looking data and the discriminator becoming better at telling real from fake. This process results in the generator learning a representation that captures the true data distribution.

Generative representation learning is particularly useful for tasks where the goal is not just to discriminate between different types of data but to understand the full spectrum of variation within the data. This includes applications in semi-supervised learning, where labeled data is scarce, and the model needs to make the most of unlabeled data. In image processing, it's used for tasks like super-resolution, photo inpainting, and style transfer. In time series analysis, generative models can be used for synthesizing realistic sequences for data augmentation, denoising signals, or even predicting future values in a sequence.

The strength of generative representation learning is that the representations learned are highly expressive, containing rich information about the data. These representations often capture deeper semantic meanings, which can be used for various downstream applications beyond generation, such as clustering and anomaly detection. The ability to model the full data distribution also allows these methods to handle incomplete, noisy, or anomalous data effectively.

2.2 Transformer in Time Series

Transformers, originally conceived for processing sequential language data, have revolutionized time series analysis by leveraging their intrinsic ability to handle sequential information and long-range dependencies effectively. The adoption of Transformer models in time series tasks stems from their self-attention mechanism, which allows the model to consider the entire sequence of data at once, contrary to the sequential processing nature of traditional recurrent neural networks (RNNs) [19] and Long Short-Term Memory networks (LSTMs) [20]. This characteristic enables the Transformer to weigh and incorporate information from distant time steps in the series, capturing complex temporal relationships that are often vital for accurate time series forecasting, anomaly detection, and classification.

In adapting Transformers to time series, several modifications are usually made. Positional encodings, which are crucial to the model's design, imbue it with the sense of order necessary for time series data, where the timing and sequence of

data points are often of paramount importance. Additionally, the Transformer's multi-headed attention mechanism allows for the parallel processing of data and the simultaneous examination of multiple aspects of the time series, such as trends and seasonalities. This parallelism is not only computationally efficient, allowing for faster processing and the handling of large datasets, but also theoretically advantageous, as it can attend to multiple temporal patterns that could be lost in a more sequential approach.

In practice, Transformers have been applied to a multitude of time series tasks across various domains. In finance [21], they can predict market movements by analyzing patterns over time. In healthcare [22], they can help identify irregularities in patient data that might signal the need for intervention. In energy sectors [23], they can forecast demand and supply to inform grid management. The model's ability to process and learn from long sequences without the constraints of a fixed window size makes it particularly suited for such complex time series problems.

The time series transformer (TST) model [24] is a representation learning model for multivariate time series. It is a quite typical model applying transformer architecture in time series. This model essentially fills the gap in applying the transformer model to the representation learning of time series. This model achieves better learning performance than supervised training methods by introducing a transformer-based pre-training mechanism. However, the TST model is based on the original self-attention mechanism. It has limitations in capturing local information, which can emphasize trend information. Moreover, TST applied generative pre-training tasks for unsupervised representation learning. It used the masking task in the same manner as the original transformer architecture. Consequently, in the unconstrained scenario, the model could potentially learn trivial solutions, such as constant mapping, which would offer minimal utility for downstream tasks [25].

The effectiveness of Transformers in these areas arises from their structure that considers entire sequences holistically, allowing for the learning of complex temporal dynamics. Moreover, the incorporation of domain-specific knowledge through customized positional encoding or additional temporal features can further enhance their performance. This, combined with their ability to learn from high-dimensional data and the ability to be trained on large datasets due to their parallelizable nature, makes Transformers a powerful tool in the time series anal-

ysis toolkit. As research in this area progresses, we continue to see advancements that tailor the Transformer architecture even more closely to the nuances of time series data, making it an increasingly indispensable method in the field.

2.3 Unsupervised Learning in Time Series

Unsupervised learning in time series analysis has become increasingly prominent as it seeks to understand and leverage the underlying structures and patterns in data without the need for labeled examples. The related work in this domain has focused on several key approaches that aim to model time series data in a manner that captures its inherent temporal dynamics and complexities.

One of the foundational techniques in unsupervised learning for time series is clustering [26]. Clustering algorithms such as k-means, hierarchical clustering, and density-based methods have been used to group similar patterns or sequences in time series data, aiding in tasks such as anomaly detection and motif discovery. Researchers have also proposed specialized time series distance measures, like Dynamic Time Warping (DTW) [27], to improve the clustering outcomes by considering the temporal alignment of sequences.

Another significant strand of related work involves dimensionality reduction [28], which seeks to transform high-dimensional time series data into lower-dimensional spaces while preserving important temporal features. Techniques such as Principal Component Analysis (PCA) [29] and Singular Value Decomposition (SVD) [30], have been applied to time series data to distill and visualize the essential patterns and trends.

Moreover, the adaptation of Generative Adversarial Networks (GANs) [31] to time series data has opened up new avenues for unsupervised learning. These models can generate new time series instances that are indistinguishable from real data, which can be particularly useful for data augmentation and simulation.

Deep learning-based methods, especially those that utilize recurrent neural networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformers, have also been employed to learn time series representations. These learned representations can then be used for various downstream tasks such as forecasting and classification without the need for labeled data. Self-supervised learning, a subset of unsupervised learning, where the model generates its own labels from the data, has also gained traction in time series. Techniques like contrastive

learning and predictive coding have shown promising results in learning robust time series representations.

In conclusion, the related work in unsupervised learning for time series is rich and diverse, reflecting the broad applicability and necessity of these methods. Researchers have developed a variety of approaches to address different challenges presented by time series data, from its high dimensionality to its temporal dependencies. As the field advances, we are likely to see more sophisticated models that can not only capture the essence of time series data more accurately but also provide deeper insights into their complex dynamics.

CHAPTER 3

Timestamp-level Representation Learning

3.1 Introduction

Multivariate time series analysis is widely used in science, finance, social media, and various other fields [32] [33]. In the era of information explosion, a large amount of multivariate time series data is generated daily. Compared to other sequence data, multivariate time series data are more ubiquitous and thus have huge application prospects. This brings new challenges to discovering knowledge from big time series data. For example, in the stock market, multivariate time series analysis of stocks requires experienced and competent analysts to analyze the market changes and behavioral logic implied behind the complicated market data [34].

Recent interdisciplinary research on deep learning has positively impacted the analysis of multivariate time series [35]. A few pre-training approaches from computer vision (CV) and natural language processing (NLP) research have been applied to time series data to enhance the connection between data [36] [37]. Transformer is a typical example. The first Transformer model was proposed for natural language translation [38]. Due to the potent capabilities of self-attention in global feature extraction, this disruptive research has since inspired devel-

opments in other fields. The Vision Transformer (ViT) [39] model, proposed for image classification, broke the domain barrier and encouraged us to apply the self-attention mechanism to multivariate time series. In particular, with the widespread adoption of transformer architecture across various domains, attention mechanisms-based time series representation has become a hot research topic.

Compared to text data, multivariate time series data exhibits similarities with image data regarding global and local characteristics [40]. Local features can emphasize trend information, a significant attribute for downstream tasks of multivariate time series data. Although attention-based characterization methods have unique advantages in learning global features, more and more studies have demonstrated that local representation learning still needs to be improved [41]. Much research on applying self-attention mechanisms in CV has focused on enhancing local features [42], which also encourages research in the multivariate time series field. A representation learning approach incorporating local and global features, without adding extra computational burden, is beneficial for multivariate time series analysis.

In addition, due to the lack of labeled data, there is widespread interest in providing efficient analysis using large amounts of unlabeled multivariate time series data [43]. Data augmentation is required for multivariate time series to constitute the training sample pairs. However, standard data augmentation techniques for time series are often inspired by CV and NLP field practices and are usually unsuited for multivariate time series. These practices carry strong inductive biases, such as transformation-invariance and cropping-invariance. Some research has already proved this issue may lead to learned representations that do not accurately encapsulate the complete information inherent to the multivariate time series [4]. This presents a significant challenge in designing sample pairs necessary for unsupervised learning in multivariate time series data.

To address these issues, this study proposes a novel unsupervised learning model named LBP4MTS (**L**ocal **B**inary **P**attern for **M**ultivariate **T**ime **S**eries). Our model enables the representation learning of multivariate time series and considers both global and local features of multivariate time series. First, the proposed model introduces a specially designed local binary pattern (LBP) method, multivariate LBP, for multivariate time series in a self-attention mechanism to improve the representation performance of the model in terms of local information. Subsequently, a variant of Dropout for multivariate time series represen-

tation, named DropLine, is designed to generate comparison sample pairs for unsupervised representation learning. Compared to conventional data augmentation methods in unsupervised representation learning, our method constructs sample pairs by network architecture instead of modifying the multivariate time series input. In this way, it's not necessary to introduce inappropriate inductive biases and assumptions.

In summary, the main contributions of our work are summarized as follows:

- We propose LBP4MTS, a novel model that can learn the representation of multivariate time series with global and local features. This model introduces an LBP-based self-attention mechanism in the transformer encoder layer (Section III.3.C) to learn a more comprehensive representation of multivariate time series.

- We develop an unsupervised training method (Section III.3.D). A variant of Dropout is also designed to construct the unsupervised sample pairs of multivariate time series.

- We conduct extensive experiments on several datasets from different fields (Section III.4). The proposed model achieves better results than other baseline methods and demonstrates its applicability to various tasks.

The remainder of this paper is organized as follows. Section 3.2 outlines previous studies on representation learning for multivariate time series, various variants of the LBP algorithm, and modifications of the Dropout method from existing literature. Section 3.3 describes the architecture of the proposed model in detail. Finally, Section 3.4 presents the experimental results, and the study conclusions are summarized in Section 3.5.

3.2 Related Work

3.2.1 LBP and Its Variants

LBP is a simple yet efficient texture operator that labels an image's pixels by thresholding each pixel's neighborhood and considers the result a binary number. LBP is wildly used in the CV field, including medical image analysis and face

recognition. Many extensions have been made to the original LBP method to enhance its performance.

To reduce computational complexity and improve texture classification performance, Uniform LBP [44] was proposed to calculate uniform patterns to account for a vast majority of all patterns in texture images. In addition, Rotation Invariant LBP [45] was designed to be invariant to the rotation of the image. Furthermore, Volume Local Binary Patterns (VLBP) [46] extended LBP into three dimensions, making it suitable for the analysis of dynamic textures in videos.

For the analysis of temporal signals such as voice, audio, and electroencephalography (EEG) signals, Chatlani and Soraghan introduced the 1D-LBP [47]. 1D-LBP is an extension of the LBP operator to one-dimensional data. It demonstrates the potentiality of applying LBP methods in time series. Like the original LBP, there can be various extensions of 1D-LBP to capture more complex patterns or provide robustness against certain signal variations. Based on 1D-LBP, TTLBP [48] extend 1D-LBP from univariate series to multivariate series data.

While these methods significantly streamline the feature extraction process for time series, they essentially remain manual feature extraction techniques, transforming the time series into histograms or distributions. Unfortunately, this transformation does not lend itself well to integration with deep learning models. Figure. 3.1 illustrates the original LBP method alongside the 1D-LBP variant.

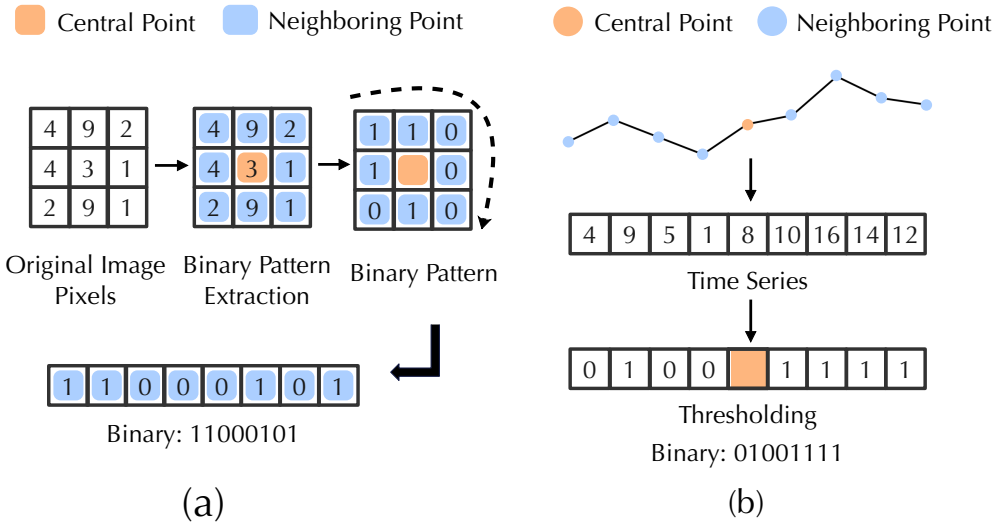


Figure 3.1. Illustration of structures of (a) LBP, (b) 1D-LBP, respectively.

3.2.2 Dropout and Its Variants

Dropout is a regularization technique for reducing overfitting in neural networks. The technique temporarily drops out, or "deactivates," neurons in a layer with a certain probability during training. This forces the network to learn more robust features that are useful in conjunction with many different random subsets of the other neurons.

DropBlock [49] is a form of structured dropout for the convolution layer. In standard dropout, neurons are dropped individually and randomly. In the convolution layer, other neurons in the same region may carry similar information due to spatial correlation. In DropBlock, a contiguous region of a feature map is dropped during training.

Spatial Dropout [50] performs dropout along specific dimensions only. During training, Spatial Dropout randomly selects a certain percentage of the channels in a convolutional layer and sets all values in these channels to zero for a given forward pass. This can often result in improved generalization and better performance on unseen data. Figure. 3.2 shows these four Dropout methods.

Some research [51] applied Dropout in contrastive unsupervised learning. These methods use random characteristics of Dropout to generate sample pairs by passing one input through the model with the Dropout layer twice. This inspired us

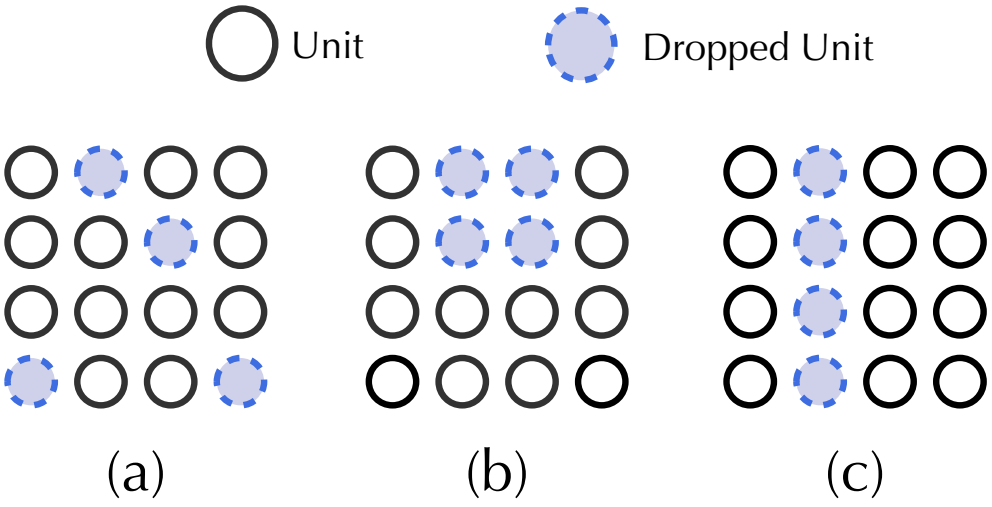


Figure 3.2. Schematic of structures of (a) Standard Dropout, (b) DropBlock, and (c) Spatial Dropout, respectively.

to apply unsupervised contrastive learning for multivariate time series.

3.3 Methodology

3.3.1 Overview

This section describes the proposed model LBP4MTS and the relevant algorithms. The structure of LBP4MTS is shown in Figure. 3.3. First, each sequence of multivariate time series goes through the encoder part twice to generate positive pairs in contrastive learning. In traditional unsupervised contrastive learning, data augmentation is usually applied to generate sample pairs. However, most existing data augmentation methods may change the original data's distribution or multivariate time series pattern information. Model-based methods are then widely used for a variety of data and tasks. These methods construct sample pairs by stochasticity in specially designed models. This can avoid issues of changing certain information of original data.

Subsequently, an LBP-based self-attention mechanism is introduced to the en-

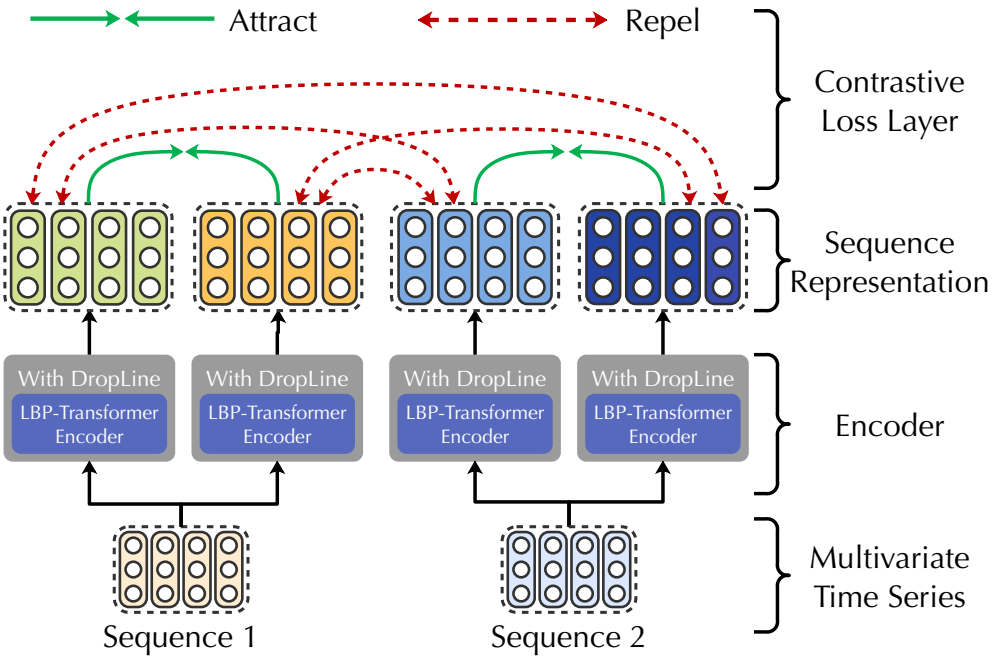


Figure 3.3. Structure of proposed unsupervised representation learning for multivariate time series.

coder of transformer architecture as a representation learning model. It uses a specially designed LBP module, multivariate LBP, to extract local features of multivariate time series. Inspired by 1D-LBP and other LBP methods, such as TTLBP, The multivariate LBP module is designed for calculating the local feature relationship matrix of tensors.

Furthermore, a novel Dropout method, DropLine, is proposed. It can be regarded as a one-dimension version of DropBlock [49]. Like DropBlock, DropLine also obstructs the transfer of pertinent information from units adjacent to the dropped unit to the subsequent layer. Then, a contrastive loss is employed to train the representation of multivariate time series.

3.3.2 Problem Definition

Given a training sample $X \in \mathbb{R}^{n \times m}$, which is a multivariate time series of length n and dimensions m, the input sequence with n vector is $x_t \in \mathbb{R}^m : X \in \mathbb{R}^{n \times m} = [x_1, x_2, \ldots, x_n]$. The proposed unsupervised representation learning model aims to train a mapping function that transforms each input data point x_t into its corresponding representation r_t. Such a representation is designed to capture the input data's most informative and distinguishing features, allowing it to describe itself effectively.

Therefore, the representation of is denoted as $R = [r_1, r_2, \ldots, r_n]$, where each vector $r_t \in \mathbb{R}^k$ represents the learned representation of the input at a particular timestamp t. Here, k denotes the dimension of representation vectors. Essentially, the model transforms each input data point x_t into a representation vector r_t of size k, capturing the essential features and characteristics of the input. The resulting representation sequence R consists of these vectors corresponding to the individual timestamps.

3.3.3 LBP-based Transformer Encoder

As previously discussed, the original self-attention mechanism falls short of adequately representing the local characteristics inherent in multivariate time series data. Hence, numerous modifications have been suggested for the original self-attention model to improve its ability to portray local features found in sequential and multivariate time series data. The feature dependence of multivariate time series in local space is similar to that of image data, i.e., for any given encoded

data point, its neighboring data points exert a more significant influence than data points located further away. Thus, convolutional Layers, a widely-used module to extract local information in CV, could be used to improve the performance of self-attention in extracting local information.

A straightforward method to encode local information is to use a convolutional layer before the self-attention mechanism. This allows the model to extract local features in the input sequence. However, both the convolutional layer and self-attention mechanism use learnable structures that are continuously updated during training. This continuous updating can lead to high computational costs, especially for deep networks with many layers and extensive training data. Several studies have opted for using non-learnable modules, like LBP, as substitutes for convolutional layers within a network [52]. These techniques can enhance computational efficiency and reduce susceptibility to overfitting. This motivates us to use LBP in the self-attention mechanism to improve the performance of extracting local features.

LBP for Multivariate Time Series

Our multivariate LBP method is an operator for multivariate time series. Given a training sample of multivariate time series $X = [x_1, x_2, \ldots, x_n]$, for each timestamp of multivariate time series, x_i, multivariate LBP defines the variant M_i as a combined similarity vector between x_i and p timestamp data points before x_i.

$$M_i = [S_{i,i-p}, S_{i,i-p+1}, \ldots, S_{i,i-1}] \tag{3.1}$$

where $S_{i,i-j}$ is the similarity value between x_i and the j^th data point before x_i. It can be expressed as follows:

$$S_{i,i-j} = s(x_i, x_{i-j}), j \in [1, p] \tag{3.2}$$

where $s(\cdot)$ denotes similarity calculation. The similarity determination in multivariate LBP cannot be made directly, as in LBP, by comparing the values of two scalars. There are numerous similarity measures for vectors that can be utilized. The selection of an appropriate similarity measure can be tailored according to the specific situation. A commonly employed measure is Cosine similarity. For any timestamp data point x_i and its neighboring data point x_{i-j}, the Cosine similarity is calculated as follows:

$$cos(x_i, x_{i-j}) = \frac{\langle x_i, x_{i-j} \rangle}{\sqrt{\langle x_i, x_i \rangle} \cdot \sqrt{\langle x_{i-j}, x_{i-j} \rangle}} \tag{3.3}$$

where $\langle \cdot \rangle$ represents the inner product.

Unlike the LBP and most variants, multivariate LBP is not symmetric and has no central point. This asymmetrical design ensures that the multivariate LBP value for each timestamp data point in multivariate time series is solely influenced by its neighboring historical data but not by any future data. Thus, the coefficient

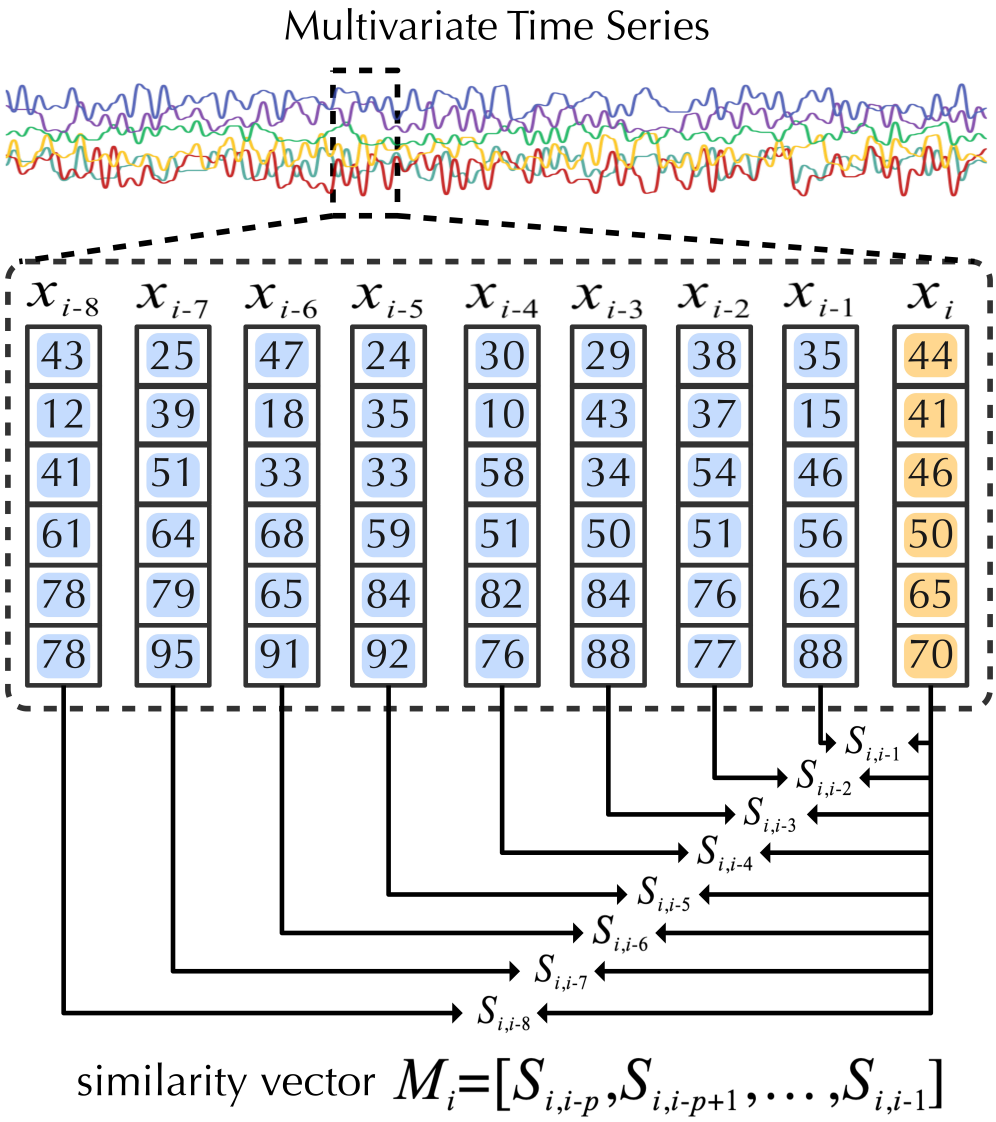


Figure 3.4. Illustration of calculation process of multivariate LBP.

of each timestamp data point, x_i, is influenced by its immediate p neighboring data points, $[x_{i-p}, x_{i-p+1}, \ldots, x_{i-1}]$. Figure. 3.4 illustrates the calculation process of multivariate LBP. As for parameter p, i.e., the number of neighboring historical data, the experiment in TTLBP proved that eight neighboring historical data get the best performance for multivariate time series. Therefore, we also choose eight neighboring historical data in this paper to calculate multivariate LBP value. For the initial timestamp data point input of multivariate time series, i.e., x_i where $i < 9$, we populate their historical data using the constant composition to compute its multivariate LBP operation.

Furthermore, unlike other LBP-based methods for extracting local features from multivariate time series, our approach does not rely on histograms to represent the local information. Instead, we directly employ the computed similarity results to create a similarity vector. This vector is then utilized to calculate an affinity matrix (AM), like the weight matrix in the self-attention mechanism. The resulting affinity matrix captures the local features of the multivariate time series in the encoder layer and is combined with the attention mechanism to enhance the overall representation.

LBP-based Self-Attention Mechanism

Based on the similarity vector calculated by the multivariate LBP method, we propose an LBP-based self-attention mechanism in the encoder of the transformer architecture to add local features to the representation learning of multivariate time series. The diagram of the proposed LBP-based self-attention mechanism is shown in Figure. 3.5, where X represents the entire sequence of multivariate time series. In this mechanism, the local feature is represented by similarity vector M_i calculated by the multivariate LBP method. An affinity matrix is then generated according to the similarity vector to reveal the degree of similarity between inputs. To enhance the robustness of the affinity matrix, we can utilize the Pearson correlation coefficient among similarity vectors of timestamp data points in the multivariate time series. The correlation coefficient serves as a centered version of cosine similarity since it involves subtracting the mean from the data points before computation.

The formula to calculate the Pearson correlation coefficient is as follows:

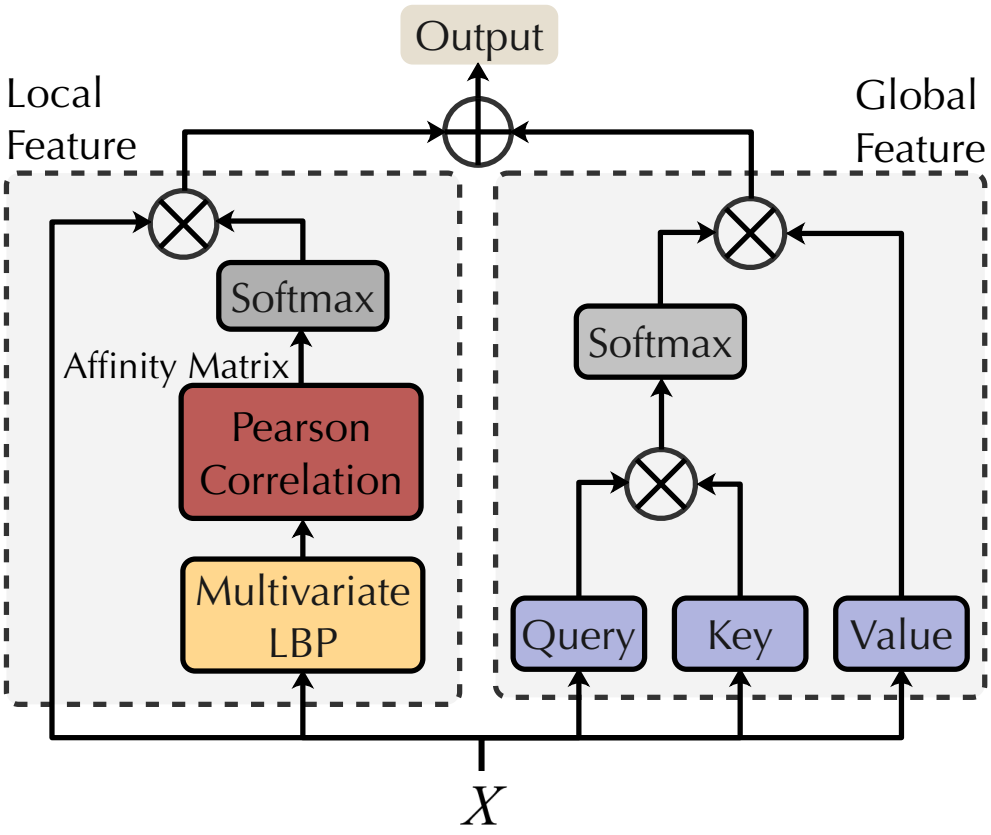


Figure 3.5. Schematic of LBP-based self-attention mechanism.

$$p_{i,j} = \frac{\langle M_i - \overline{M_i}, M_j - \overline{M_j} \rangle}{\sqrt{\langle M_i - \overline{M_i}, M_i - \overline{M_i} \rangle} \cdot \sqrt{\langle M_j - \overline{M_j}, M_j - \overline{M_j} \rangle}} \tag{3.4}$$

where $\overline{M_i}$ is the mean of similarity vector m_i. By calculating the Pearson correlation coefficient of among each timestamp data point in multivariate time series, the affinity matrix is formed as follows:

$$AM(X) = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,n} \\ p_{2,1} & p_{2,2} & \cdots & p_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n,1} & p_{n,2} & \cdots & p_{n,n} \end{bmatrix} \tag{3.5}$$

where AM is an abbreviation for affinity matrix.

In original self-attention, for any input X, the function of self-attention is expressed as follows:

$$Q = XW^Q; K = XW^K; V = XW^V \tag{3.6}$$

$$\text{Attention} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V \tag{3.7}$$

where Q,K,V represent the matrices of queries, keys (with dimension d_k),and values (with dimension d_v), respectively. As shown in equation (3), queries, keys, and values are transformed through linear projections by $W^Q \in \mathbb{R}^{d_m \times d_k}$, $W^K \in \mathbb{R}^{d_m \times d_k}$ and $W^V \in \mathbb{R}^{d_m \times d_v}$, respectively, where d_m is the dimension of the input.

After adding a multivariate LBP module, the function of LBP-based self-attention can be described as follows:

$$\text{LBPAttention} = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V + softmax\left(AM(X)\right) \cdot X \tag{3.8}$$

In Equation (8), the first component $softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$ represents the global feature and the second component $softmax\left(AM(X)\right) \cdot X$ represents local feature.

3.3.4 Unsupervised Training

Unsupervised learning is particularly pertinent to multivariate time series data analysis, given the considerable effort and expense often associated with obtaining labeled data.

Most existing research on unsupervised learning for multivariate time series relies on data augmentation from fields of CV or NLP to generate sample pairs. These techniques might not always be suitable due to the unique characteristics of multivariate time series data, such as temporal dependency. The inductive biases transformation-invariance, like rotating an image in CV, or cropping-invariance, like cropping part of a sentence in NLP, might not hold true in the case of multivariate time series data.

Besides utilizing data augmentation to create sample pairs during the pre-processing phase, a rising number of studies are now turning to apply model stochasticity, like the Dropout layer, in the training phase to generate positive training pairs [51]. This strategy aims to avoid the potential negative impacts that data augmentation could impose on the input data. Inspired by these, a

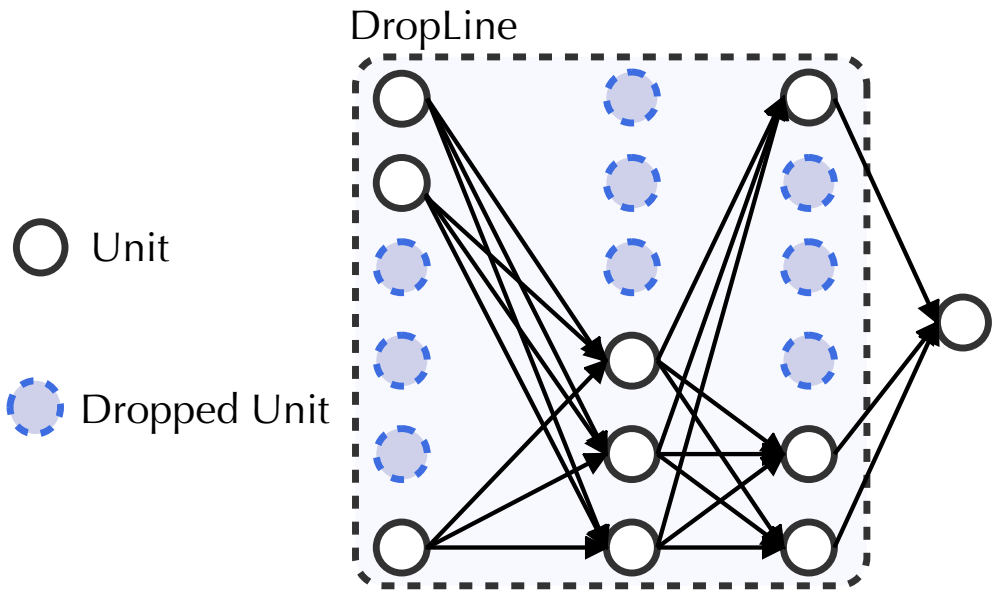


Figure 3.6. Schematic of DropLine method in standard neural network.

variate of Dropout is proposed for multivariate time series to generate training sample pairs without traditional data augmentation methods. This method is named DropLine and can be added to most training models for multivariate time series. The diagram of DropLine is shown in Figure. 3.6.

Compared with standard Dropout, DropLine randomly discards continuous neuron units within the layer, i.e., a line of neuron units is dropped. This design is based on the understanding that for any given timestamp data point, neighboring neural nodes could potentially hold similar information because of temporal continuity. Essentially, it suggests that random deactivation of individual nodes does not necessarily lead to a total loss of relevant information.

After the DropLine operator, the sample pairs of unsupervised training are obtained. Then, the training object of contrastive learning is to learn an encoder such that:

$$score\left(f(x), f(x^+)\right) \gg score\left(f(x), f(x^-)\right) \tag{3.9}$$

where x^+ is the positive sample, and x^- is the negative sample. *score* is often expressed as a distance function, which means that the training object can be formulated by computing the distance among the anchor, positive, and negative samples. It is articulated as follows:

$$max(d(x, x^+) - d(x, x^-) + margin, 0) \tag{3.10}$$

where $d(\cdot)$ is the distance between the sample pairs, and the margin is a hyper-parameter to control the distances. The loss function can be defined as follows:

$$-\log \frac{e^{\cos(x_i, x_i^+)/\tau}}{\Sigma_{j=1}^{n} \left(e^{\cos(x_i, x_j^+)/\tau} + e^{\cos(x_i, x_j^-)/\tau} \right)} \tag{3.11}$$

where τ is a temperature hyperparameter and n is training batch size.

3.4 Experiments

In this section, we assess the performance of our model by analyzing its performance across various tasks. We employ classification and regression tasks as downstream tasks to evaluate the value of local features in the representation learning of multivariate time series.

In the subsequent experiments outlined below, we employ the predefined training-test splits of the benchmark datasets and ensure all models are sufficiently trained to achieve convergence. An initial adjustment of the hyper-parameters (such as the number of training batch sizes, the number of encoder blocks, or the representation dimension) for each distinct dataset can contribute to enhanced performance. After the hyper-parameters were determined, the complete training set was leveraged for model training, which was ultimately assessed using the test set.

To more accurately assess the performance of our algorithm, we employed K-fold cross-validation (ten-fold cross-validation) on each dataset and repeated the experiment 5 times for each fold.

3.4.1 Classification

In this subsection, we report the experiments conducted to evaluate the effectiveness of our proposed model on the UEA dataset [53], using the classification task as a downstream task. The UEA dataset is significant for researching and analyzing multivariate time series time data. Benefited from its expansive collection of real-world multivariate time series data, the UEA dataset provides a consistent benchmark for researchers. Its ongoing updates and expansions not only ensure

its enduring relevance in the ever-evolving research landscape but have also led to its increasing adoption in a multitude of time series studies worldwide. It currently has 128 univariate and 30 multivariate time-series classification datasets. We conducted repeat experiments on ten multivariate time series datasets to verify the performance, providing multiple datasets from different domains, with varying dimensions, unequal length dimensions, and missing values. The summary of these datasets is shown in Table 3.1.

In the classification task, the output vector of our model was passed through a SoftMax function to obtain a distribution over classes, and its cross-entropy with the categorical ground truth labels was considered as the sample loss. This experiment can directly verify the performance of the proposed representation learning model.

The UEA archives also provide an initial benchmark for the existing models, with accurate baseline information including classification accuracy. The benchmarks facilitate consistency in evaluations, ensuring that methodologies are compared under standardized conditions. Based on these information, we chose these four models as our baseline for multivariate time series classification: dimension-dependent dynamic time warping (DTW_D) [54], TST [24], XGBoost [55] and TS2Vec [4]. Adhering to the approach outlined by the TST model, we utilize the best-performing method, DTW_D that the authors of the UEA archive examined, as our benchmark for comparison. Meanwhile, as the first and the most

Table 3.1. Summary of UEA multivariate classification datasets.

Dataset	Train Size	Test Size	Length	Classes	Dimensions	Type
EthanolConcentration	261	263	1751	4	3	Other
FaceDetection	5890	3524	62	2	144	EEG
Handwriting	150	850	152	26	3	HAR
Heartbeat	204	205	405	2	61	AUDIO
JapaneseVowels	270	370	29	9	12	AUDIO
PEMS-SF	267	173	144	7	983	MISC
SelfRegulationSCP1	268	293	896	2	6	EEG
SelfRegulationSCP2	200	180	1152	2	7	EEG
SpokenArabicDigits	6599	2199	93	10	13	SPEECH
UWaveGestureLibrary	2238	2241	315	8	3	HAR

Table 3.2. Accuracy results of classification of the proposed and baseline methods.

Dataset	LBP4MTS	DTW_D	XGBoost	TST	TS2Vec
EthanolConcentration	**0.429**	0.305	0.417	0.258	0.288
FaceDetection	**0.661**	0.526	0.635	0.535	0.500
Handwriting	0.361	0.278	0.175	0.215	**0.479**
Heartbeat	0.725	0.727	0.732	**0.739**	0.694
JapaneseVowels	0.951	0.909	0.917	**0.980**	0.943
PEMS-SF	0.692	0.703	**0.967**	0.737	0.677
SelfRegulationSCP1	**0.845**	0.753	0.823	0.714	0.818
SelfRegulationSCP2	**0.597**	0.528	0.489	0.550	0.570
SpokenArabicDigits	**0.997**	0.959	0.712	0.931	0.973
UWaveGestureLibrary	0.910	0.907	0.772	0.900	**0.912**
Average Accuracy	**0.717**	0.660	0.664	0.656	0.686
Average Rank	**1.9**	3.5	3.3	3.2	3.1

famous model that introduces transformer architecture to representation learning of multivariate time series, TST is also considered as the baseline. Additionally, XGBoost is among the most frequently utilized models for both univariate and multivariate time series analysis, which can also be used as a baseline to evaluate the performance of our model. Finally, TS2Vec is currently the most advanced representation learning model for multivariate time series, which also be included for comparison. These methods are the best-performing methods studied by the creators of the archive. Among these four methods, TST and TS2Vec are neural network-based models, while DTW_D and XGBoost are traditional methods. Table 3.2 presents our model's and baseline models' classification results for the multivariate time series, where bold indicates the best values. The Critical Difference diagram for the Nemenyi test applied to these datasets is depicted in Figure. 3.7. Classifiers not linked by a bold line exhibit significant differences in their average ranks. This provides strong evidence that our algorithm notably surpasses other methods.

Table 3.2 reveals that our proposed model exhibited superior performance on five out of the ten datasets, achieving an average ranking of 1.9^{th}. This was followed by TS2Vec and TST, which outperformed the remaining two datasets and achieved average ranks of 3.1^{th} and 3.2^{th}, respectively. XGBoost performed

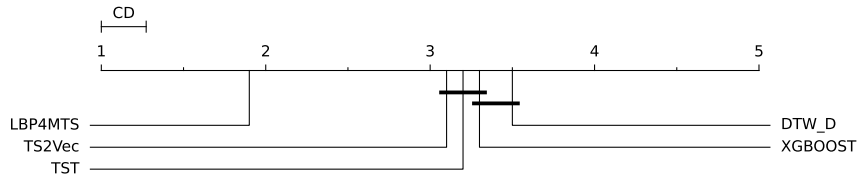


Figure 3.7. Critical Difference (CD) diagram of representation learning methods on time series classification tasks with a confidence level of 95%.

best on the remaining 1 dataset, ranked 3.3^{th} on average. The table clearly indicates that methods based on neural networks generally yield superior results, aligning with the current understanding of the significant role neural networks play in the advancement of multivariate time series analysis. We note that all datasets where TS2Vec surpassed our model's performance were extremely low-dimensional, specifically 3-dimensional. Compared with other methods, TST achieves the best result of performance in multivariate time series with the type of AUDIO. In terms of XGBoost, it demonstrates robust performance on highly dimensional data, highlighting potential limitations of methods grounded in neural networks.

Interestingly, the data presented in the table also suggests a clear positive relationship between the efficacy of our model and the volume of available data, especially for large-scale training data. This indicates that as the quantity of data increases, the performance of our model also significantly improves. It further underscores the importance of large datasets in enhancing the model's predictive power and generalization capabilities, which is crucial in machine learning and data-driven decision-making. This correlation between data volume and model effectiveness could pave the way for future research and developments in optimizing data collection and utilization methods.

3.4.2 Regression

In this subsection, the regression task is introduced as the downstream task to evaluate the effectiveness of our proposed model. multivariate Time series regression is a statistical method that is used to analyze multivariate time series data. multivariate Time series regression aims to create a mathematical model that can predict future responses based on the behavior observed in past data.

This method can be used to forecast trends, cycles, or other patterns in the data that tend to repeat over time.

We chose various datasets from UEA&UCR Time Series Regression Archive [56]. . Table 3.3 presents detailed characteristics of these datasets. As mentioned in experiments of TST, this selection was made to ensure a diverse representation concerning the dimensionality and length of multivariate time series samples and the number of samples.

In the regression task, we choose root mean square error (RMSE) to evaluate the performance of different models. RMSE is a commonly used metric in regression analysis and forecasting to measure the model's prediction error. The RMSE represents the sample standard deviation of the differences between predicted and observed values. Essentially, it tells you how concentrated the data is around the line of best fit. RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(P_i - O_i)^2} \tag{3.12}$$

where n is the number of observations, P_i is the predicted value for observation i and O_i is the observed value for observation i.

Meanwhile, inspired by the TST paper, we also incorporate the "average relative difference from the mean" evaluation criterion. This addition can help RMSE in mitigating the impact of different magnitudes across various datasets, thereby providing a more accurate measure of different models' performance across diverse datasets. The metric average relative difference from the mean (represented as r_j for each model j) can be defined as follows:

Table 3.3. Details of multivariate regression datasets.

Dataset	Train Size	Test Size	Length	Dimensions	Missing Values
AppliancesEnergy	96	42	144	24	No
BenzeneConcentration	3433	5445	240	8	Yes
BeijingPM10Quality	12432	5100	24	9	Yes
BeijingPM25Quality	12432	5100	24	9	Yes
LiveFuelMoistureContent	3493	1510	365	7	No
IEEEPPG	1768	1328	1000	5	No

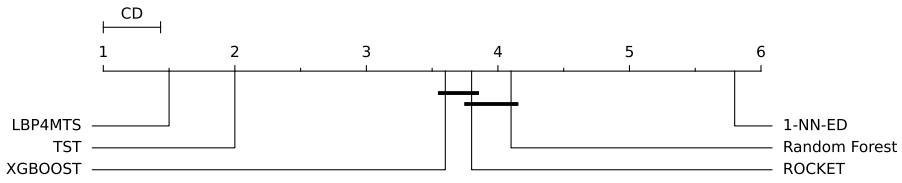


Figure 3.8. Critical Difference (CD) diagram of representation learning methods on time series regression tasks with a confidence level of 95%.

$$r_j = \frac{1}{N} \sum_{i=1}^{N} \frac{R(i,j) - \bar{R}_i}{\bar{R}_i} \tag{3.13}$$

$$\bar{R}_i = \frac{1}{M} \sum_{k=1}^{M} R(i,k) \tag{3.14}$$

where $R(i,j)$ is the RMSE of model j on dataset i, N is the number of datasets, and M is the number of models. Upon analyzing this particular metric, it is obvious that a smaller value of the average relative difference from the mean corresponds to superior model performance.

As same as the classification task, The UEA&UCR Time Series Regression Archive also provides an initial benchmark for the existing models, with accurate baseline information. Based on the performance metrics provided by the archives, we chose these five models as our baseline for multivariate time series classification: ROCKET [57], XGBoost [55], 1-NN-ED [58], Random Forest [59], and TST [24]. According to the results reported in the archive, these methods emerge as the top five-performing algorithms. Table 3.4 presents the RMSE of regression results of our model and baseline models for the multivariate time series, where bold indicates best values. The Critical Difference diagram illustrating the results from the Nemenyi test for various datasets can be seen in Figure. 3.8. If algorithms are not connected by a bold line, it indicates significant disparities. Such evidence compellingly underscores the superiority of our algorithm over the other methods.

As the results in Table 3.4 indicate, our model yields the best performance on three datasets, outperforming all other models. On the remaining three datasets, where our model didn't achieve optimal performance, it secured the second position. The second one is the TST model, which proves optimal on two datasets,

while the ROCKET model, securing the third position, is optimal on one dataset. Thus, the overall ranking for our model stands at 1.5. The outcomes from both TST and our model underscore the efficacy of deep learning models in the representation learning of multivariate time series. Even though our model managed to achieve second rank on three datasets, the analysis of these datasets uncovers a limitation in our model's capability to utilize local features when dealing with multivariate time series data of shorter lengths (such as BeijingPM10Quality and BeijingPM25Quality datasets). Moreover, deep learning-based models tend not to perform well with smaller sample datasets (such as the Appliances dataset). Several factors might contribute to these limitations. For one, smaller datasets limit the diversity and variability within the data, constraining the model's learning process. Without a broad range of data to train on, the model might miss subtle patterns or nuances. When working with compact datasets, the model may not have sufficient information to train effectively, potentially leading to overfitting or reduced generalization capabilities. Meanwhile, by comparing the results of our model with those of the TST model, it can be seen that generative unsupervised learning could potentially outperform contrastive unsupervised learning when it comes to learning representations of shorter sequences. This insight outlines our prospective direction for enhancement.

Table 3.4. Performance of regression task for our and baseline models on multivariate regression datasets (RMSE).

Dataset	LBP4MTS	ROCKET	XGBoost	1-NN-ED	Random Forest	TST
AppliancesEnergy	2.355	**2.240**	3.494	5.273	3.415	2.359
BenzeneConcentration	**0.461**	3.160	0.662	6.296	0.815	0.506
BeijingPM10Quality	84.783	113.943	94.589	130.583	96.946	**82.996**
BeijingPM25Quality	55.789	60.874	60.352	84.806	65.905	**53.153**
LiveFuelMoistureContent	**43.795**	44.651	48.897	57.901	47.685	44.785
IEEEPPG	**23.909**	35.115	30.877	31.685	30.879	26.469
Ave Rel. diff. from mean	**-0.280**	0.095	-0.110	0.650	-0.084	-0.270
Average Rank	**1.5**	3.8	3.6	5.8	4.1	2

Table 3.5. Ablation results for LBP4MTS and its variants.

	Average Accuracy	Accuracy Decline
LBP4MTS	**0.719**	-
w/o LBP	0.681	-3.8%
w/o DropLine	0.693	-2.6%
w/o LBP & DropLine	0.670	-4.9%

3.4.3 Ablation Study

To validate the efficacy of the proposed components in our model, i.e., the LBP-based self-attention and DropLine, we compare the full LBP4MTS model against its three variants across ten UEA datasets outlined in Table 3.1. To swiftly and effectively demonstrate the efficacy of each module within the proposed LBP4MTS model, a classification experiment is adopted for the ablation study. The results of the ablation study were evaluated based on the accuracy of the classification results and their percentage change.

Table 3.5 presents the results of this ablation study, where (1) **w/o LBP** removes the LBP-based self-attention module and employs original self-attention mechanism, (2) **w/o DropLine** removes DropLine designed in this paper and applies Dropout to unsupervised train the model, (3) **w/o LBP & DropLine** remove both LBP-based self-attention module and DropLine. The results demonstrate that every component within the LBP4MTS structure is essential and irreplaceable.

Meanwhile, the comparison of the results from LBP4MTS and its variate without LBP-based self-attention suggests that local features play a pivotal role in the representation learning of multivariate time series. This is attributed to the fact that the trends of its neighboring data heavily influence the timestamp data points within a multivariate time series. Capturing local features enables the model to more accurately depict the underlying pattern of change within the multivariate time series. In addition, by comparing the difference in results between LBP4MTS and its variate without DropLine, we can observe that the DropLine module is more adept at constructing sample pairs for unsupervised training of multivariate time series. This outcome is credited to DropLine's capacity to avoid the leakage of information from neighboring timestamp data points, resulting in a more effective unsupervised model training process.

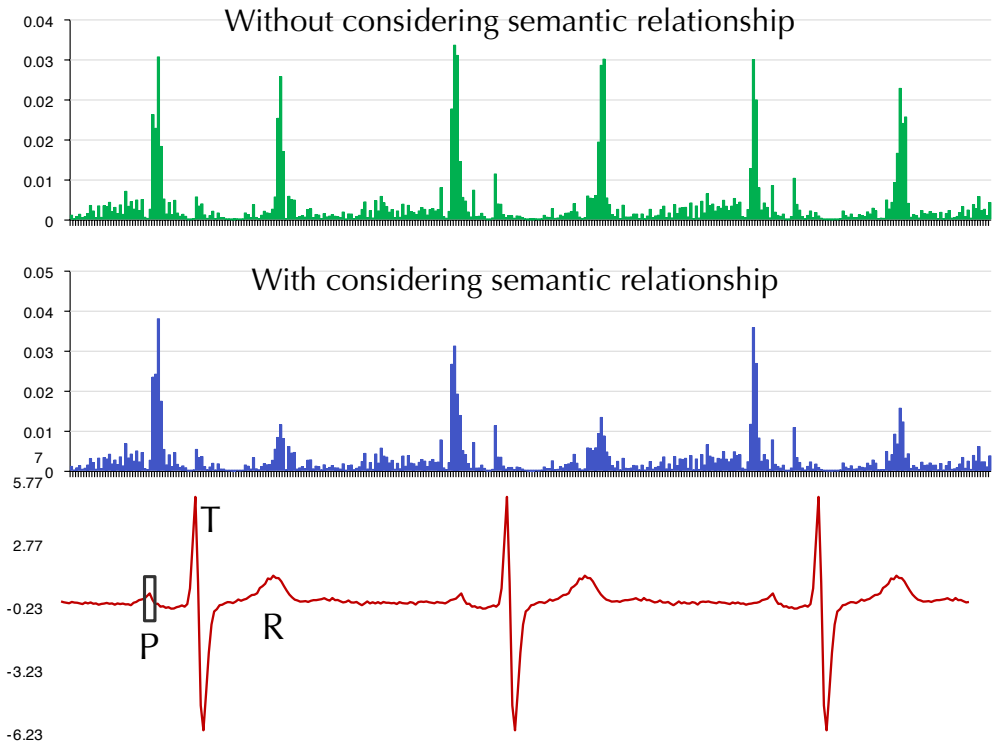


Figure 3.9. Example of without and with considering semantic relationship in representation learning of ECG.

3.4.4 Case Study

In this case study, the example of ECG time series was considered more comprehensively to clarify the effect of semantic relationship. ECG time series has long been used as a testbed for algorithms of time series. Researchers studying ECG complexity note that each status of ECG can be more intricate. As shown in Figure 3.9, the status P and T have quite similar shapes. This kind of similarity may lead to confusion of representation of ECG.

If we attempt representation learning based on classic attention mechanism (i.e., without considering semantic relationship) , the attention matrix can not identify status P and status R. They have similar attention weights. Meanwhile, in our model, which considering semantic relationship, status P and status R have different attention weight.

Figure 3.9 vividly illustrates the impact of semantic relationships in the context of time series representation learning. This visualization highlights how semantic relationships facilitate the representation learning algorithm's ability to incorpo-

rate information from neighboring sequences when analyzing segments of similar time series. By doing so, it significantly enhances the algorithm's capacity to generate more precise and meaningful representations. This approach allows for a deeper understanding of the underlying patterns and connections within the data, leading to more effective and insightful analysis of time series.

3.5 Conclusion and Future Work

Given the inherent nature of multivariate time series data, local features play a crucial role in the representation learning process. The identification of local patterns and trends can provide a wealth of insights that global analysis might miss. However, in the original self-attention mechanism, these local aspects were not effectively captured, potentially losing important information. In this study, our LBP-based transformer encoder is proposed as a mechanism to represent multivariate time series. This model aims to overcome the shortcomings of the original model in local feature extraction. In addition, a variate of Dropout, DropLine, is designed to construct the sample pairs of multivariate time series and to achieve unsupervised contrastive learning. DropLine is based on the understanding that neighboring neural nodes could potentially hold similar information because of temporal continuity. The conducted experiments reveal that the proposed model exhibits substantial improvement in the representation learning of multivariate time series. An ablation study proves the effectiveness of components within the LBP4MTS structure. Consequently, it can be employed in various downstream tasks, such as classification and regression.

In future research, our efforts will be devoted to improving the performance of our model in datasets with small data sizes and short data lengths. These include leveraging transfer learning from pre-trained models, integrating domain-specific or external data sources for added context, and exploring hybrid models to bolster the model's adaptability to short data sequences. By harnessing these approaches, we anticipate marked improvements in model efficacy across diverse data scenarios. Meanwhile, we find the design of the loss function to be a captivating aspect of unsupervised representation learning of multivariate time series. So far, a variety of loss functions have been engineered to cater to diverse applications. Thus, we believe that optimizing the loss function could further enhance the performance of our model.

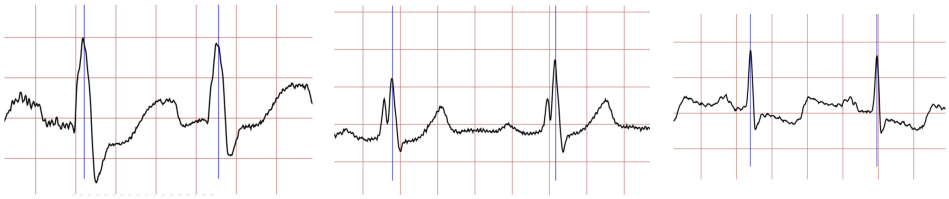
CHAPTER 4

Segment-level Representation Learning

4.1 Introduction

The significant progress of the Internet and widespread use of sensors has driven the remarkable development of multivariate time series, such as electrocardiograms [60] and daily stock prices [61]. This plays an important role in the field of data engineering. As the application scenarios and downstream tasks of multivariate time series become increasingly complex, representation learning can advance the analysis of multivariate time series and become a universal tool for feature detection and preprocessing of raw data. Representation learning replaces manual feature engineering and enables the learning and use of features to perform a specific task.

Conversely, several semantic-based methods and algorithms have recently demonstrated good performance in the areas of natural language processing (NLP) and computer vision (CV). Extracting semantic-based features is the first step of almost all CV models [62]. Inspired by this progress, an increasing number of algorithms for multivariate time series have used semantic information in a wide variety of tasks, especially in a data search of time series [63]. These algorithms convert multivariate time series data into several subseries based on semantic



	Abnormal	Abnormal	Normal
Shapelet	Class 1	Class 2	Class 1
DTW	Class 1	Class 2	Class 2
Proposed method	Class 1	Class 1	Class 2

Figure 4.1. Example of the issue in semantic-based time subseries. These figures represent different states of the heart. Class 1 and class 2 represents abnormal state and normal state respectively.

information. Semantic information can be revealed by the shape of the curve of time series data (e.g., the shapelet learning method [64]) or other statistical information, such as the mean and maximum values of time series data [65]. Semantic-based methods have a natural advantage, i.e., they can convert a time series into several subseries according to their semantic information; this characteristic can be beneficial for data storage and search.

These algorithms have shown good results in information retrieval and classification tasks. However, there are still some limitations and weaknesses in previous studies. Most of these semantic-based algorithms focus on obtaining accurate semantic subseries rather than the relationships among different subseries. They require another algorithm to learn the relationships among subseries, which may increase the computational cost of downstream algorithms and affect the performance. This can be illustrated by analyzing the incorrect results in the experiments with these semantic-based algorithms. Figure. 4.1 shows a typical example of this issue in electrocardiogram (ECG) classification tasks.

As can be seen in the Figure. 4.1, the three curves of the ECG time series have similar shapes, although they represent different states of the heart. In such a situation, the traditional semantic-based algorithms classified them into incorrect classes. This is a common phenomenon in the real world, which is

mainly caused by disregarding the relationships between different subseries. We refer to these semantic relationships as high-level semantics. High-level semantics is a fundamental concept in CV [66] that can distinguish an object in an image by considering the surrounding information of the neighbors of the target object. Given this definition, real-world time series also have high-level semantics, i.e., the relation between neighbor subseries and target subseries that can enhance the performance of semantic-based time series methods. Thus, our motivation is to design a representation learning model by representing subseries of multivariate time series with high-level semantics.

In addition, there are other problems with traditional time series algorithms that are challenging for various reasons. First, most real-life time series are unlabeled. Therefore, unsupervised algorithms are strongly preferred because of their broader application scenarios, i.e., unlabeled time series data can be used, and more adaptive features can be learned. Second, the methods should deliver compatible representations while allowing the input time series to have unequal lengths. Given that the algorithm divides the entire time series into several subseries according to semantic information, the length of each subseries may differ.

In this study, we propose a novel unsupervised learning framework to learn the representation of semantic-based subseries of multivariate time series. The proposed model represents the subseries by considering the covariance calculated by the Gaussian process (GP) to reveal their high-level semantics (HLS) and is named GP-HLS. First, a Gaussian process-based attention mechanism is introduced to the encoder of the transformer [38] as the representation learning model. It uses the covariance calculated by the GP as the external information to consider the high-level semantics of each subseries of the multivariate time series. Subsequently, a Gaussian drop-based triplet network is designed for multivariate time series to construct the positive and negative sample pairs of unsupervised training. In addition, we use an advanced segmentation algorithm named greedy Gaussian segmentation (GGS) [67] to generate several subseries of multivariate time series. And a widely used input regularization method, named temporal pyramid pooling (TPP) [68], is considered to generate regular inputs for time series subseries with unequal lengths.

In summary, the main contributions of our work are as follows:

- We propose a transformer encoder-based architecture with the GP in the

self-attention mechanism (Section 4.3.2) that uses covariance information to learn high-level semantic features in subseries inputs.

- We develop an unsupervised training method (Section 4.3.3). Triplet sample pairs for multivariate time series data based on the Gaussian drop are also designed to construct the unsupervised sample pairs of multivariate time series.

- We conduct extensive experiments on several datasets from different fields (Section 4.4). In comparison to other baseline algorithms, the proposed GP-HLS model achieves better results and is applicable different tasks.

The remainder of this paper is organized as follows. Section 4.2 outlines previous studies on representation learning for multivariate time series and self-attention mechanisms from existing literature. Section 4.3 describes the architecture of the proposed model in detail. Finally, Section 4.4 presents the experimental results, and the study conclusions are summarized in Section 4.5.

4.2 Related Work

Self-attention (also called intra-attention [69]) is an attention mechanism that constructs attention models using the relationship between the input samples

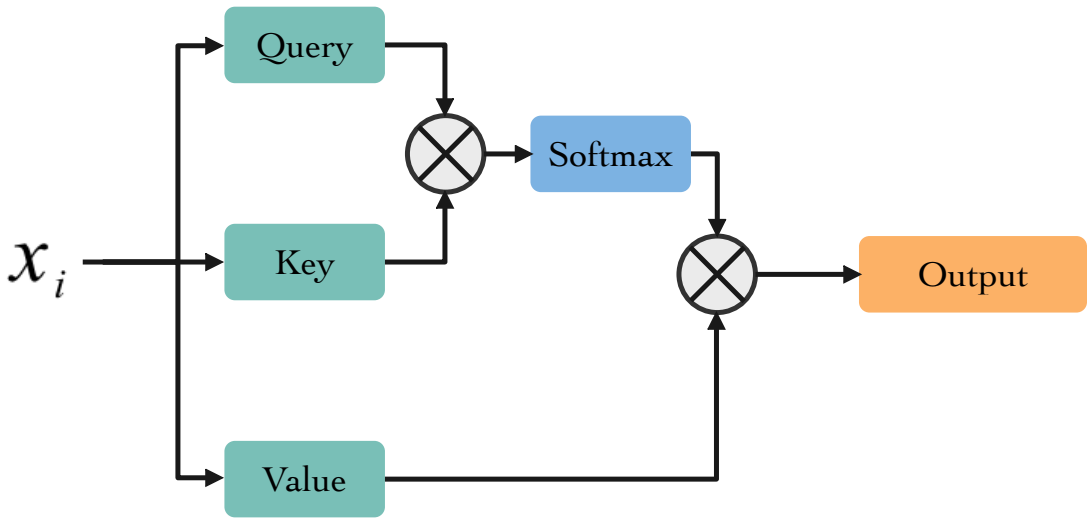


Figure 4.2. Schematic of original self-attention of transformer.

themselves. It is useful in a wide field of machine learning, such as image processing, text representation and data prediction. The most well-known application of self-attention is the transformer [38] proposed for NLP tasks.

Assuming that x_i represents a certain training batch consisting of several subseries of multivariate time series, the original self-attention can be described as shown in Figure. 4.2.

The function of self-attention is expressed as

$$Q = x_i W_i^Q; K = x_i W_i^K; V = x_i W_i^V \tag{4.1}$$

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4.2}$$

where Q,K,V are the matrices of queries, keys of dimension d_k,and values of dimension d_v, respectively. As shown in equation (1), queries, keys, and values are projected by the linear transformations $W_i^Q \in \mathbb{R}^{d_m \times d_k}$, $W_i^K \in \mathbb{R}^{d_m \times d_k}$ and $W_i^V \in \mathbb{R}^{d_m \times d_v}$, respectively, where d_m is the dimension of the input.

As shown in the paper on ordinary transformers [38], the self-attention mechanism represents inputs by calculating their similarity, which can generate a suitable representation for words. However, this is not sufficient for representing subseries with high-level semantics. It not only requires similarity information among different subseries, but also the correlation among each subseries that plays a significant role in the representation learning of time series.

4.3 Methodology

4.3.1 Overview

In this section, the proposed GP-HLS model structure and the relevant algorithms are described. The structure of GP-HLS is shown in Figure. 4.3. First, an input regularization method of one-dimensional data is considered to generate regular inputs for time series subseries with unequal lengths. Subsequently, a GP-based attention mechanism is introduced to the encoder of a transformer as a representation learning model. It uses covariance calculated by the GP as the external information to consider the high-level semantic features of each subseries of the multivariate time series. Then, a Gaussian drop-based triplet loss function

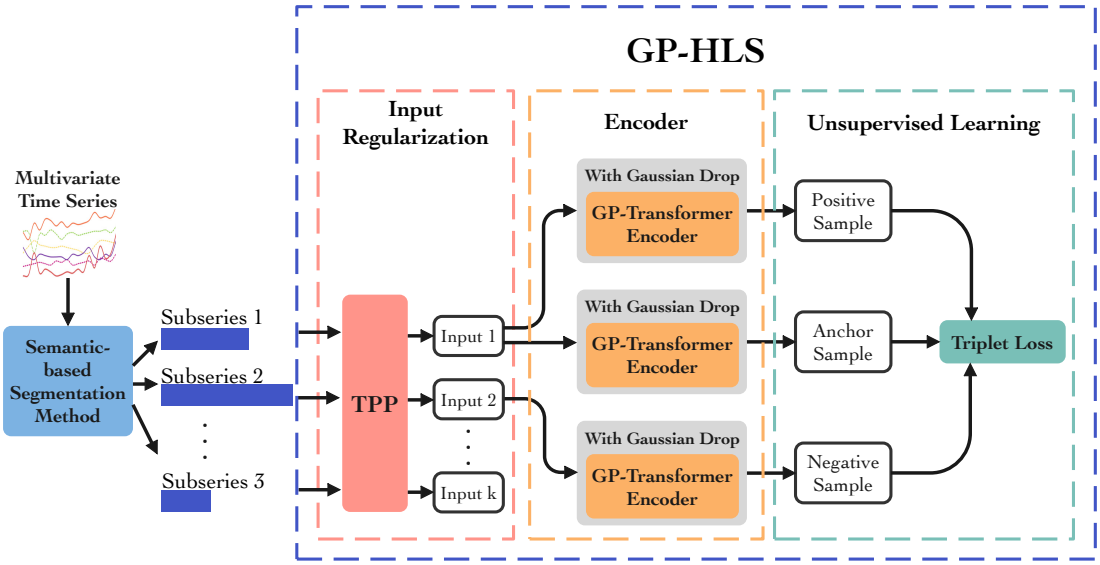


Figure 4.3. Structure of unsupervised representation learning for time series with high-level semantic features.

is designed for multivariate time series to construct the positive and negative sample pairs of unsupervised training.

Because most semantic-based segmentation methods divide the entire time series into several subseries with varying lengths, we must reshape them with unequal lengths. Then, the model learns their representation. We apply TPP [68] to regularize the subseries input generated by the segmentation method, which was proposed to deal with the varying length issue of the input for one-dimensional data. The TPP method is illustrated in Figure. 4.4.

4.3.2 Gaussian Process-based Self-Attention Mechanism

As introduced earlier, the original self-attention mechanism is not sufficient to represent subseries with high-level semantics. The correlation among each subseries is necessary for the representation learning of time series, and especially for revealing the high-level semantics in time series.

Based on this concept, we propose a GP-based self-attention mechanism in the encoder of the transformer architecture to add correlation information to the representation learning of multivariate time series. The diagram of the proposed model is shown in Figure. 4.5, where X represents the entire sequence of the

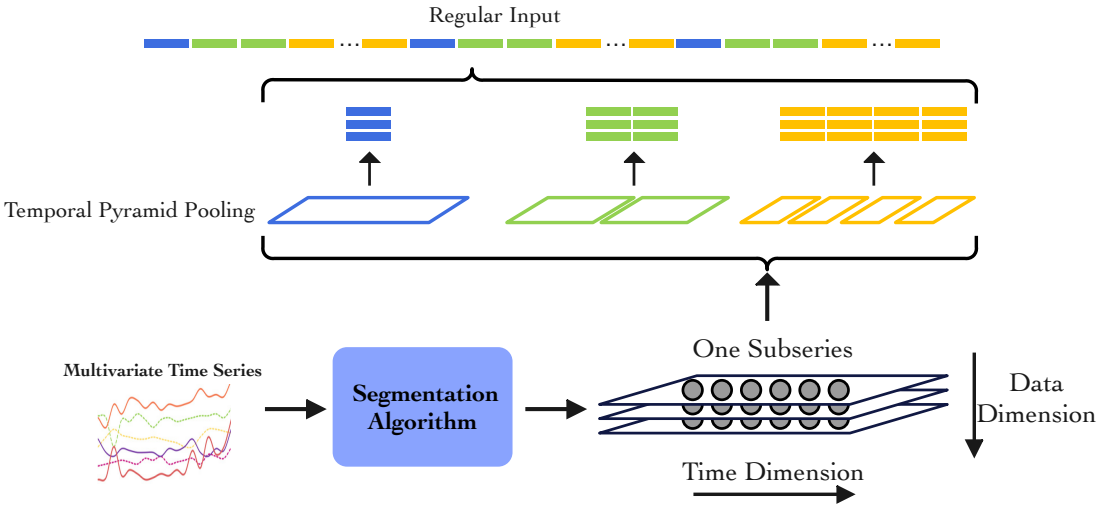


Figure 4.4. Detailed diagram of input regularization method.

multivariate time series. In our model, the covariance function is learned from the GP and the covariance matrix is then generated according to the subseries in the batch. The covariance matrix can reveal the correlation among each subseries in the input batch, which can be used as the correlation matrix for subsequent calculations.

After adding the GP part to the self-attention mechanism, the function of self-attention described in equation (2) can be rewritten as:

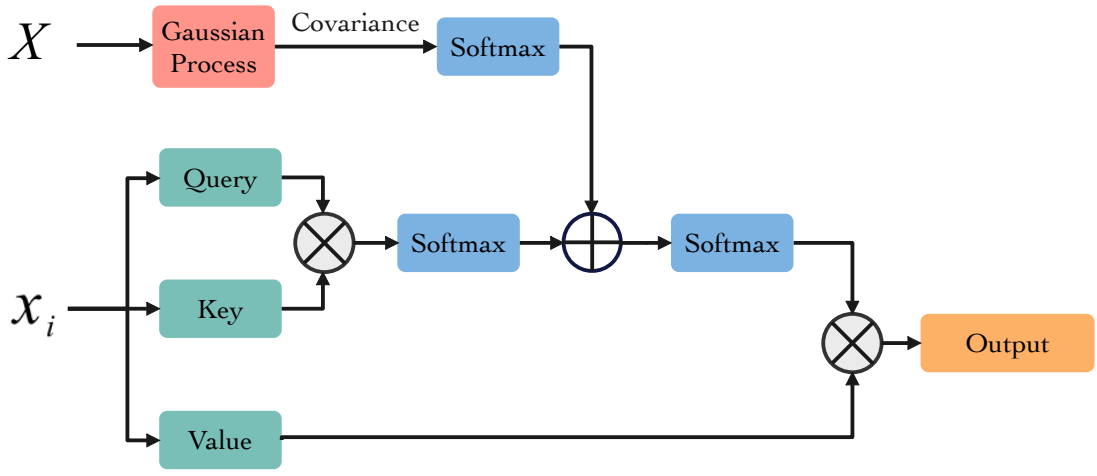


Figure 4.5. Schematic of Gaussian process-based self-attention mechanism.

$$Attention(Q, K, V) = softmax\left(softmax\left(\frac{QK^T}{\sqrt{d_k}}\right) + softmax\left(Cov(x_i)\right)\right)V \tag{4.3}$$

where $Cov(x_i)$ represents the covariance matrix of subseries in the input batch. In equation (3), the first component $\frac{QK^T}{\sqrt{d_k}}$ represents the similarity relationship of the input, and the second component $Cov(x_i)$ represents the correlation relationship of the input.

For covariance, a fundamental fact of GP is that it can be defined entirely by second-order statistics [70]. Thus, if a GP is assumed to have a mean of zero, the covariance function ultimately defines the behavior of the process.

Covariance is the core of GP, which can be determined by the different kernel functions. This also expands the scope of the application of our model. For other types of data, we can choose different kernel functions to obtain a better representation of data correlations. In this study, we chose a radial basis function kernel (RBF). It is also known as the squared-exponential kernel.

The RBF kernel is stationary and parameterized by a length scale $l > 0$, which can be either a scalar (an isotropic variant of the kernel) or a vector with the same dimensions as the inputs x (an anisotropic variant of the kernel). The kernel is expressed as

$$k(x_i, x_j) = \sigma^2 exp\left(-\frac{d(x_i, x_i)^2}{2l^2}\right) \tag{4.4}$$

where σ^2 is a hyperparameter, l is the kernel length scale, and $d(\cdot, \cdot)$ is the Euclidean distance.

4.3.3 Unsupervised Training

The triplet network was developed from the Siamese network [71], which is an artificial neural network that uses the same weights while working in tandem on two different input vectors to compute comparable output vectors. In comparison with the Siamese network, the triplet network uses both positive and negative samples. This joint training of positive and negative pairs could help the model easily distinguish the input from the same class and different classes. To use the triplet network, labeled data are necessary. However, most real-life time series

are unlabeled. Therefore, unsupervised representation learning was suitable for training.

In this section, we introduce our simple unsupervised training method. The key point of unsupervised representation learning is to ensure that similar time series obtain similar representations with no supervision to learn such similarities. While there are some unsupervised methods for time series representation learning, most of them require manual training pair design. This not only increases the complexity of the algorithm but also makes the training pairs rely on the precision of manual methods, which cannot generate universal training pairs for most training models. Hence, we design an unsupervised method for time series to select pairs of similar time series inspired by the recent development of unsupervised methods and contrast learning in CV [72] and NLP [51]. This is a sample method that can be added to most training models.

Dropout is a relatively general and straightforward method for machine learning models. Owing to the random characteristic of dropout, one input will have two different eigenvectors when going through a model with a dropout layer. We

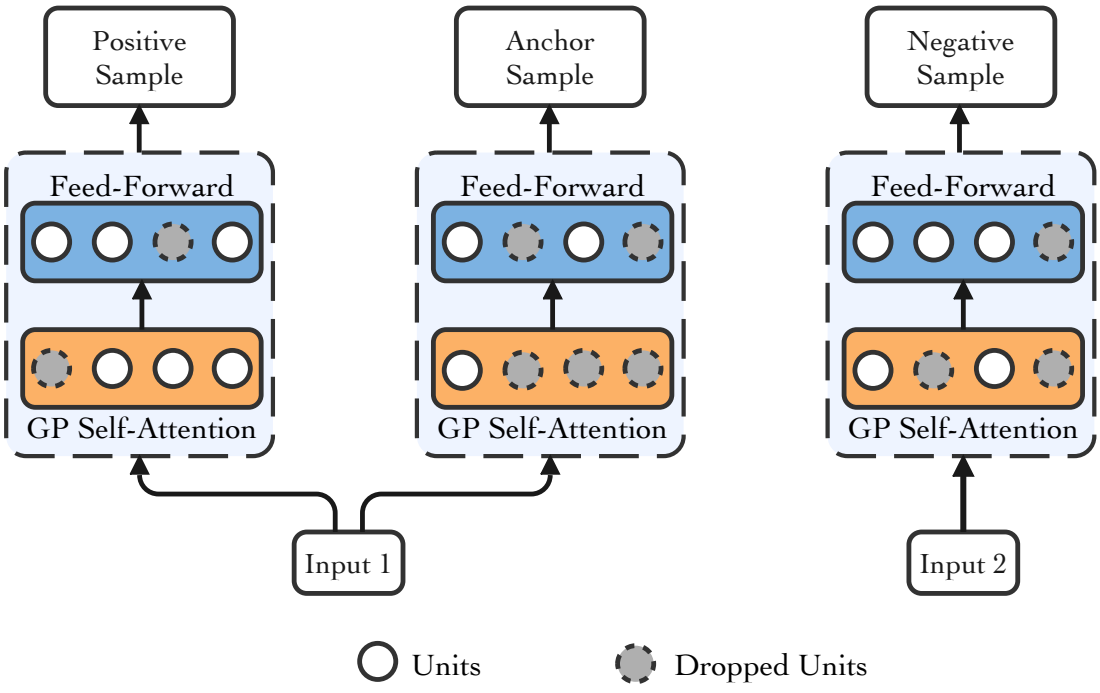


Figure 4.6. Schematic of generating training pairs for triplet network of representation learning.

develop an unsupervised method for converting the Siamese network into a triplet network, which achieves data enhancement without changing the original high-level semantic features and information of the data. While some contrast learning models in CV and NLP use the standard dropout layer to generate positive pairs, we choose the Gaussian dropout for representation learning of multivariate time series. A diagram of the generation of the training pairs (anchor, positive, and negative samples) for the triplet network of representation learning is shown in Figure. 4.6.

In comparison with the standard dropout layer, Gaussian dropout discards neurons using a probability that fits a Gaussian distribution. This is equivalent to adding multiplicative noise to the input signal that obeys a Gaussian distribution. This Gaussian noise does not change the original distribution of the multivariate time series and can maintain the consistency of the data distribution in the model.

The original training object of the triplet loss is calculated by the distance between the positive, anchor, and negative samples expressed as:

$$max(d(x, x^+) - d(x, x^-) + margin, 0) \tag{4.5}$$

where x^+ is the positive sample, and x^- is the negative sample; $d(\cdot)$ is the distance between the input pairs, and margin is a hyperparameter to control the distances. Considering the dropout unsupervised method in NLP [73] and the process of generating the sample pairs in our model, the training objective can be defined as follows:

$$-\log \frac{e^{\cos(x_i, x_i^+)/\tau}}{\Sigma_{j=1}^{N} \left(e^{\cos(x_i, x_j^+)/\tau} + e^{\cos(x_i, x_j^-)/\tau} \right)} \tag{4.6}$$

where $\cos(\cdot, \cdot)$ is the cosine distance, N is the mini-batch size, and τ is a temperature hyperparameter. Based on this principle, the convergence speed of the triplet network can be improved.

4.4 Experiments

In this section, we test the effectiveness of our model by analyzing its performance on different tasks. Classification and retrieval tasks are used as downstream tasks to prove the effectiveness of high-level semantic information in representation

learning. In addition, a case study is conducted to recall the example introduced in Section 4.1.

4.4.1 Classification

In the classification task, the output vector of our model was passed through a softmax function to obtain a distribution over classes, and its cross-entropy with the categorical ground truth labels was considered as the sample loss. In this task, we show that our model performs better than other unsupervised methods.

We used the following six multivariate datasets from the UEA time series classification archives [54], which provide multiple datasets from different domains with varying dimensions, unequal lengths, and missing values. A summary of these datasets is presented in Table 4.1.

Meanwhile, the UEA archives provide an initial benchmark for existing models that provided accurate baseline information. Based on the performance metrics provided by the UEA archives, we chose the following three models as our baseline:

- Dimension-dependent dynamic time warping (DTW_D) [74]: it uses a weighted combination of raw series and first-order differences for neural network classification with either Euclidean distance or full-window dynamic time warping (DTW). It combines two distances, i.e., the DTW distance between two series and two different series, using a weighting parameter. It develops the traditional DTW method and suits every series of data.

- ROCKET [57]: it is based on a random convolutional kernel similar to a shallow convolutional neural network. It can achieve fast and accurate time series classification using random convolutional kernels.

- Time series transformer (TST) model [24]: it largely fills the gap in the application of the transformer model to the representation learning of time series. This model achieves a better learning performance by introducing a transformer-based pre-training model.

Table 4.2 presents the classification results for the multivariate time series,where bold indicates best values. As shown in Table 2, the proposed model demonstrated the best performance among the four datasets. From the data presented

Table 4.1. Summary of UEA multivariate datasets.

Dataset	Train Size	Test Size	Length	Classes	Dimensions
EthanolConcentration	261	263	1751	4	3
Handwriting	150	850	152	26	3
Heartbeat	204	205	405	2	61
PEMS-SF	267	173	144	7	983
SpokenArabicDigits	6599	2199	93	10	13
HJapaneseVowels	270	370	29	9	12

in the table, it can be concluded that the effectiveness of our model is significantly enhanced as the amount of data increases.

However, our model is relatively more advantageous for small datasets than baselines. The results for the Heartbeat datasets revealed that binary classification is more likely to exploit contrastive learning. Conversely, our model yielded better results for datasets with trend changes. In general, the results of the SpokenArabicDigits data indicate a relative weakness of our model, i.e., it has no significant advantage when dealing with large scale data. And for handwriting, our proposed method and TST both have undesirable results. We can draw a conclusion that attention mechanism has a weak ability in dealing with the low-dimensional data, especially when the training data is obviously less than test data. To mitigate these issues, we intend to set new feature parameters and try other mechanism to increase the sensitivity of the model to such data in our future work.

Table 4.2. Accuracy results of proposed and other methods.

Dataset	GP-HLS	DTW_D	ROCKET	TST
EthanolConcentration	**0.467**	0.452	0.326	0.326
Handwriting	0.312	0.286	**0.588**	0.309
Heartbeat	**0.781**	0.717	0.756	0.776
PEMS-SF	**0.919**	0.711	0.751	0.896
SpokenArabicDigits	0.968	0.963	0.712	**0.993**
HJapaneseVowels	**0.997**	0.949	0.962	0.994

Table 4.3. The details of two multivariate time series datasets in experiment. N.A. denotes not available.

Dataset	Number of Attributes	Number of Instances	Classes
EEG Eye State	15	14980	2
Twitter	77	583250	N.A.

4.4.2 Retrieval

For the time series retrieval task, we evaluate the effectiveness of the proposed model for unsupervised time series retrieval tasks based on two different datasets. Table 4.3 The statistics of two multivariate time series datasets in experiment. N.A. denotes not available.

The EEG Eye State dataset was collected from one continuous EEG measurement using the Emotiv EEG Neuroheadset [75]. All data has 117 seconds duration of the measurement. The eye state was detected using a camera during the EEG measurement. '1' represents a closed eye, and '0' the eye-open state. In this experiment, we generate 6012 segments by GGS algorithm.

The Twitter dataset was collected to predict Buzz from the Buzz in social media Dataset [76]. It contains examples of buzz events from Twitter. And it does not have any label of class information. In this experiment, we generate 49803 segments by GGS algorithm.

The details of two datasets are shown in Table 4.3. For those segments in these two datasets, we select 50% as the training data, next 10% as the validation data, and the last 40% as the test data.

We compared our model with three typical baseline methods in time series retrieval. All these methods are unsupervised. DeepBit [77] is an unsupervised deep learning approach. It can learn binary descriptors in an unsupervised manner. HashGAN [78] is a deep unsupervised hashing function, which is also designed for image retrieval. The last baseline is Long Short-Term Memory (LSTM) encoder-decoder (LSTM-ED) [79]. It uses an encoder LSTM to map an input sequence into a fixed length representation.

To evaluate the performance of proposed model and baseline models in the task of unsupervised multivariate time series retrieval, we calculate the K nearest neighbors (KNN) based on Euclidean distance (ED). For each query segment, we first calculate its KNN as the ground truth (KNN=100 for EEG Eye State

Table 4.4. Unsupervised multivariate time series retrieval performance (MAP).

Dataset	EEG Eye State			Twitter		
Hidden size	64	128	256	64	128	256
GP-HLS	**0.282**	**0.336**	**0.395**	**0.108**	**0.144**	**0.171**
DeepBit	0.225	0.284	0.325	0.040	0.089	0.102
HashGAN	0.206	0.299	0.320	0.051	0.101	0.101
LSTM-ED	0.245	0.325	0.357	0.077	0.113	0.143

and KNN=500 for Twitter dataset). Then, we search the representation of similar segments based on the Hamming distance. Finally, the mean average precision (MAP) is reported for comparison purposes. Meanwhile, to evaluate the performance of each model more comprehensively, we use three different hidden size of each model, 64, 128 and 256. The MAP results of each model are shown in Table 4.4. We notice that our model has a strong advantage compared to baseline models. It is mainly owing to the use of semantic-based segments and our high-level semantics representation learning algorithm. Meanwhile, we observed that LSTM-ED consistently outperformed DeepBit and HashGAN. The reason may be the DeepBit and HashGAN are specifically designed for images and cannot represent the temporal information in the input segment. These two algorithms could need more necessary improvement for use in time series tasks.

4.4.3 Case Study

In this case study, we revisit the example in Section 4.1 at greater depth to explain the motivation for this work. As introduced in Section 4.1, by analyzing the wrong results in experiments of some semantic-based algorithms, we conclude that the relationship among subseries plays a significant role in representation learning of multi-variate time series. In this section, we design an experiment to further address this issue.

Most time series datasets are carefully designed and selected with a perfect distribution or measure precision. However, time series from the real world may have many problems, such as noise, loss, or measurement errors. Additionally, measuring data from different equipment or sources interferes with each other. These issues can significantly affect the performance of models. Therefore, we combined two other datasets from the same subject that were obtained from var-

Table 4.5. Summary of ECG200 and TwoLeadECG.

Dataset	Train Size	Test Size	Length	Classes	Dimensions
ECG200	100	100	96	2	1
TwoLeadECG	23	1139	82	2	1
Combined ECG	123	1239	82	2	1

ious sources. Specifically, we used ECG 200 and TwoLeadECG as the datasets from UCR time series classification archive [58]. Both datasets trace the recorded electrical activity and contain two classes: normal heartbeat and myocardial infarction (MI). We randomly combined these two datasets and reshaped the length of the combined ECG dataset to obtain a regular length of time series. The details of these datasets are listed in Table 4.5.

We chose the DTW_D and Shapelet Transform as the baseline algorithms. The Shapelet Transform (ST) [80] is based on the shapelet method, which separates the shapelet discovery from the classifier by finding the top k shapelets in a single run. Shapelets were used to transform the data, and each attribute in the new dataset represented the distance of a series to one of the shapelets. This is a semantic-based method for time series. First, we conducted experiments on these two datasets separately. Then, we conducted an experiment using the combined dataset. The results of the experiments in the case study are listed in Table 4.6.

4.5 Ablation Study

4.5.1 Ablation of Gaussian Dropout

In this section, we discuss the performance difference between standard dropout and Gaussian dropout in unsupervised representation learning. As introduced

Table 4.6. Accuracy results of proposed and other methods.

Dataset	GP-HLS	DTW_D	ST
ECG200	**0.902**	0.880	0.840
TwoLeadECG	**0.991**	0.868	0.984
Combined ECG	**0.752**	0.442	0.510

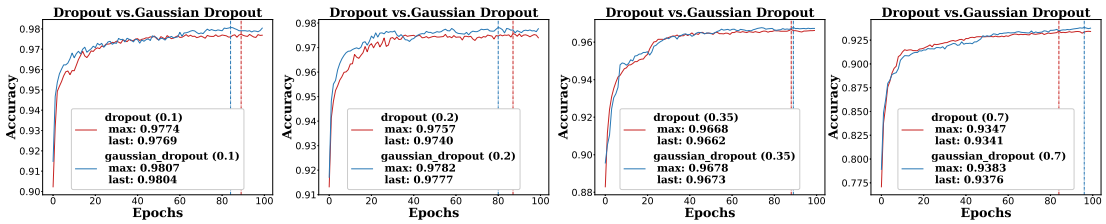


Figure 4.7. Training curves of standard dropout and Gaussian dropout respectively in training phrase.

in Section 4.3.3, Gaussian Dropout can achieve smoother gradients and improve training performance compared to standard Dropout, especially in scenarios where the dropout probability is high.

To compare the training performance of standard dropout and Gaussian dropout, both these two dropout layer are applied in training model. To better evaluate the performance of standard dropout and Gaussian dropout, we selected dropout rates of 0.1, 0.2, 0.35, and 0.7 to verify their performance on the dataset. The training curve and accuracy are shown in Figure. 4.7.

As shown in Fig. 4.7, Gaussian dropout has a better training curve with higher maximum accuracy and higher last accuracy in all four types of dropout rates. Typical dropout rate values range from 0.2 to 0.5. Considering input layers, the choice of dropout rate depends on the type of input. For real-valued inputs, a typical value is 0.2. In the case of hidden layers, the dropout rate selection is associated with the number of hidden units. Choosing a higher dropout rate requires a larger number of hidden units, which slows down the training process and may result in under-fitting. Conversely, selecting a smaller rate may not provide enough dropout to prevent over-fitting. By comparing the maximum and final accuracy of the training phase at different dropout rates, it is evident that as the dropout rate increases, the accuracy tends to decrease. Moreover, the distance of accuracy results between standard dropout and Gaussian dropout also tends to decrease. In addition, with a higher dropout rate, the time to reach the maximum accuracy during training gradually increases, and when the dropout rate is higher than 0.3, the training time of Gaussian dropout is no longer faster than that of the standard dropout. In conclusion, all four comparative experiments of standard dropout and Gaussian dropout demonstrate that Gaussian dropout is more suitable for unsupervised representation learning of multivariate time series.

4.5.2 Ablation of GP Component in Self-Attention

To verify the effectiveness of the covariance considered in our proposed model, a comparison between the full model and the model without the GP component in self-attention mechanism on UEA multivariate datasets described in Table 4.1 is shown in Table 4.7. The results of classification show that the GP component is indispensable.

The ablation study results suggest that ignoring or removing covariance information may lead to sub-optimal representations of time series data. After removing the GP component, the classification accuracy of all datasets decreased to varying degrees. The experiments without the GP component show that the results on average decreased by 2.6%, with the maximum reduction being 9.3% and the minimum reduction being 0.3%. This is sufficient to demonstrate the importance of the GP component in representation learning of multivariate time series. Instead, considering the covariance structure can lead to better and more informative representations, which can improve the performance of downstream tasks such as prediction or classification. Covariance refers to the degree to which two variables change together over time. In the context of time series data, the covariance structure captures the dependencies and relationships between differ-

Table 4.7. Accuracy results of the full model and the model without the GP component.

Dataset	Full model	Without GP Component
EthanolConcentration	0.467	0.374 (-9.3%)
FaceDetection	0.717	0.685 (-4.4%)
Handwriting	0.302	0.296 (-1.9%)
Heartbeat	0.781	0.770 (-1.1%)
JapaneseVowels	0.997	0.994(-0.3%)
PEMS-SF	0.919	0.881 (-3.8%)
SelfRegulationSCP1	0.955	0.934 (-2.2%)
SelfRegulationSCP2	0.626	0.0.580 (-7.3%)
SpokenArabicDigits	0.968	0.960(-0.8%)
UWaveGestureLibrary	0.903	0.895 (-0.9%)
Average Accuracy	0.76	0.74 (-2.6%)

ent subseries of time series. By considering the covariance structure, a model can learn to extract more meaningful and informative features that capture the dynamics of the data.

4.6 Discussion

4.6.1 Summary of Contributions

This paper introduces a novel attention mechanism for time series representation learning based on Gaussian Processes. Unlike traditional methods focusing only on timestamp-level and instance-level representations, our approach aims at representation of subseries-level of time series. Our model can capture semantic relationship from time series subseries, broadening the applicability of time series representation learning across various scenarios.

4.6.2 Comparison to Related Work

Our focus on subseries-level representation adds granularity to the time series learning landscape, potentially enriching its applicability across various scenarios. This level of representation captures semantic information within the segments, thereby improving the overall representation quality. This is particularly valuable when dealing with time series data that requires interpretation of segments as cohesive units rather than disjoint time-stamps or entire instances. In addition, the introduction of an unsupervised training methodology eliminates the need for manually crafted training pairs, often a significant bottleneck in unsupervised learning. This makes the method more autonomous and possibly easier to deploy in real-world applications.

4.6.3 Limitations

- **Segmentation algorithm**: One primary limitation is the absence of a specialized algorithm for subseries segmentation. Currently, our model does not adaptively select or generate time segments for representation learning, which could be a vital feature for optimizing performance.

- **Datasets limitation**: Our model seems to excel on small datasets but falters when the dimensions are low or when there are fewer training samples. This suggests that the model may require sufficient variability and complexity in the data for effective representation learning.

- **Granularity constraints**: Our focus on segment-level representation could be a limitation when applied to tasks that require finer-grained information. For instance, certain applications may require millisecond-level data interpretation, which our current model might not sufficiently capture.

4.7 Conclusion and Future Work

4.7.1 Conclusion

High-level semantics is essential for representation learning of time series data. This is particularly true in data search of time series. Our high-level semantic methods can represent time series by converting them into several subseries according to their high-level semantics information. This characteristic is beneficial for data storage and search. In this study, we propose a novel unsupervised representation learning model with high-level semantic features of multivariate time series. A Gaussian process-based self-attention mechanism was introduced to the encoder of the transformer as the representation learning model. In addition, a Gaussian drop-based triplet net-work was designed for multivariate time series to construct positive and negative sample pairs of unsupervised training. The experiments show that the proposed model demonstrates significant improvement in multivariate time series representation learning and can be used in various downstream tasks such as classification and retrieval. In future research, our efforts will be devoted to the design of the triplet loss function. So far, many different loss functions have been designed for various applications. Thus, we believe that the loss function may improve the performance of our model.

4.7.2 Future Work

- **Adaptive time-segment partitioning**: Incorporating a dynamic time-segment partitioning algorithm could potentially improve representation quality and model robustness.

- **Application scenarios**: Given the model's limitations on small training samples and low-dimensional datasets, future work should explore techniques to improve performance under these conditions. This could involve regularization methods, data augmentation techniques, or leveraging transfer learning.

- **Cross-granularity representation**: One intriguing direction is to extend the model to handle different levels of granularity simultaneously. By developing a multi-scale approach, the model could become more versatile and applicable to a wider range of tasks.

CHAPTER 5

Representation Learning for Streaming Time Series

5.1 Introduction

The proliferation of digital technologies and the Internet of Things has generated an unprecedented volume of time series. This wealth of information are collected from a wide range of domains, including finance [61] and healthcare [60]. As a result, researchers and practitioners have recognized the immense potential and value inherent in extracting insights and patterns from time series. With the increasing complexity of application scenarios and downstream tasks involving time series, representation learning has emerged as a powerful technique for advancing their analysis. Representation learning allows for the learning and utilization of task-specific features, thus enhancing the performance of various data analysis tasks.

However, most representation learning are designed to represent each timestamp of time series. They cannot represent the state of subsequences,i.e., the semantic information. This makes these timestamp-level methods not suitable for certain downstream tasks, like retrieval. Therefore, semantic-based methods has been widely concerned in representation learning of time series [81]. Figure. 5.1 shows a typical example of semantic information in electrocardiograms

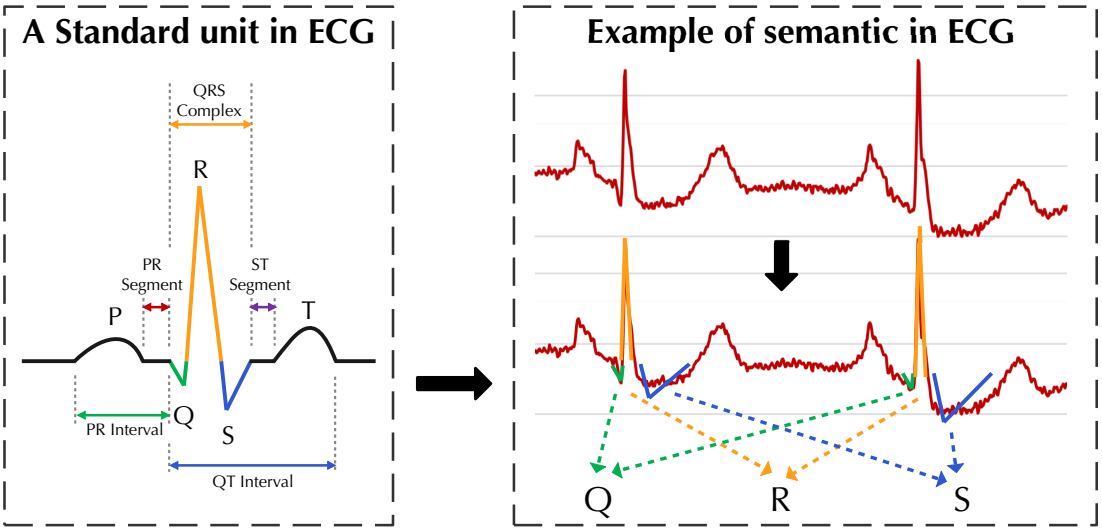


Figure 5.1. Example of semantic information in ECG.

(ECG). As shown in the left side of Figure. 5.1, a standard unit in ECG can be divided into several parts with different states. These different states can be regarded as the semantic in time series. And according to different semantic, a ECG time series data can be represented more efficiently (as shown in the right side of Figure. 5.1).

In addition, as more and more devices become smarter and ubiquitous, they generate time series data with characteristics of large volume and continuous accumulation. This type of time series is referred to as streaming time series. It differs from traditional time series data in that it is constantly updated. The efficient analysis of streaming time series holds practical significance. For instance, the increasing popularity of smartwatches has enabled the collection and analysis of streaming ECG. It is benefit for detecting heart diseases more promptly and accurately.

Nevertheless, the issue of representation learning in streaming time series remains a huge challenge. In particular, few studies have focused on both streaming time series and semantic information in representation learning. Given the continuous and frequent updates in streaming time series data, it is impractical to apply established studies on the semantic representation of time series, with ignoring the dynamic nature of streaming time series. This requires the design of representation learning algorithms specifically suitable for characteristics of streaming time series.

In this study, we first propose a unsupervised representation learning framework to provide a paradigm for calculating representations of semantic information for various types of time series. Subsequently, for streaming time series, a novel representation learning algorithm is designed according to the framework. This proposed algorithm introduces the recursive covariance estimation in a simplified Transformer structure, PoolFormer, and is named CPFormer. Some research has already proved that covariance can reveal the semantic information in time series [82]. Furthermore, a stochastic Pooling-based triplet network is designed specifically for streaming time series to generate positive and negative sample pairs for unsupervised training.

In summary, the main contributions of our work are as follows:

- This paper presents a unsupervised representation learning framework for representing the semantic information of time series (Section 5.4).
- Subsequently, a novel representation learning algorithm, CPFormer, is designed to learn the semantic-based representation of streaming time series (Section 5.5).
- We conducted extensive experiments on several public datasets from different fields (Section 5.6). In comparison with other baseline algorithms, the proposed CPFormer algorithm achieved an improved performance.

The rest of this paper is organized as follows. Section 5.2 outlines previous research on representation learning of streaming time series, as well as some variants of Transformer architecture. Section 5.3 describes some preliminaries. Section 5.4 proposes the framework for representing the semantic information of time series. Section 5.5 presents the architecture of the proposed algorithm CPFormer in detail. Thereafter, Section 5.6 discussed the experimental results. Finally, Section 5.7 gives conclusions and future work.

5.2 Related Works

5.2.1 Representation Learning of Streaming Time Series

Basically, there are three types of representation learning for streaming time series: traditional feature engineering methods, symbolic representation methods

and neural networks-based methods.

Traditional feature engineering methods include statistical measures and transform-based methods. These techniques aim to extract relevant features from streaming data to represent the temporal patterns and variations. Statistical measures relies on statistical indexes such as mean, variance, and standard deviation over a sliding window of data. Transform-based methods (such as Fourier transform-based [83] and Wavelet transform-based [84]) decomposes the streaming time series into different scales and time-frequency components. These traditional feature methods are based on pre-defined mathematical formulas or transformations. They may not be able to capture complex patterns or dependencies present in the streaming time series data, leading to sub-optimal representations.

Symbolic representation methods is an alternative approach for representing streaming time series. These methods transform the data into a symbolic form using discrete symbols or patterns. Symbolic Aggregate Approximation (SAX) [85] represents a time series by mapping it to a sequence of symbols based on breakpoints derived from the data distribution. Symbolic Dynamic Time Warping (S-DTW) [86] approximates the original time series by aligning and comparing subsequences based on symbolic representations. Generally speaking, these symbolic representation methods can be regarded as the semantic-based methods. However, these methods often rely on human expertise to select relevant symbolic. This process can be time-consuming, subjective, and may not capture the suitable semantic information of the streaming time series data.

Neural networks-based methods is truly a learning approach of representation of streaming time series. Neural network models have shown great promise for representation learning of streaming time series data. These models leverage the power of deep learning to automatically learn meaningful representations from the raw data. Many classical neural network methods have been applied to this issue (such as recurrent neural networks (RNN) and convolutional neural networks (CNN)). However, most of these representation learning methods are focus on timestamp-level representation, which can not represent the state of subsequences and relationship between different semantic patterns.

5.2.2 Variants of Transformer Architecture

Transformer architecture, initially introduced for natural language processing (NLP) tasks, has been adapted and extended for various domains. This famous architecture has already developed several variants, which demonstrate the versatility and adaptability of the original model.

Reformer [87] addresses the computational efficiency of Transformer by introducing a set of optimizations. It leverages reversible layers, chunked processing, and locality-sensitive hashing to reduce memory requirements and enable training and inference on longer sequences.

Performer [88] is another variant of the Transformer architecture that approximates the self-attention mechanism with a faster and more memory-efficient approach. It uses the kernelized self-attention to significantly reduce the computational complexity of the attention mechanism.

MetaFormer [89] is designed based on the observation that attention-based module in Transformer can be replaced by spatial multilayer perceptron (MLP) and the resulted models still perform quite well. This research proposed a token mixer component to replace the self-attention in original Transformer architecture. In the paper of MetaFormer, a Pooling-based token mixer is applied in MetaFormer, which is named PoolFormer. PoolFormer is used to illustrate the performance of the model. MetaFormer architecture allows subsequent studies to develop different designs and studies for different application scenarios.

5.3 Preliminaries

Definition 1 Streaming Time Series Streaming time series T is a discrete and growing continuously, which is obtained from collecting or sampling a data stream at certain timestamp. It can be expressed as $T = \{x_1, x_2, ..., x_n, ...\}$, where x_n represent the data arriving at the n-th timestamp.

In contrast to time series, streaming time series can continuously grow over time. Therefore the definition of streaming time series is unbounded in the right side. To fit the requirement of representing streaming time series with semantic information, the updated streaming time series should be divided into a series of subsequences with different semantic patterns. Therefore, the subsequence of streaming time series can be expressed as follows.

Definition 2 Subsequence of Streaming Time Series A subsequence of a streaming time series T is defined as a finite sequence of contiguous real numbers extracted from the original time series. A subsequence S of length l with the start time k can be expressed as $S = \{x_k, x_k + 1, ..., x_{k+l-1}\}$, where $1 \leq k \leq n - l + 1$. It can also be simplified expressed as $S = \{x_k : x_{k+l-1}\}$.

For the sequences that have already segmented, the various indexes, such as length, are already determined. Therefore, it is necessary to define the incomplete subsequence accordingly, taking into account the newly collected timestamp data. This incomplete subsequence consists of a subset of the most recent contiguous real numbers from the streaming time series.

Definition 3 Newly Incoming Subsequence At timestamp m, if there is a new collected data point x_m and the last h timestamp data have not been segmented, the newly incomplete subsequence $C = \{x_{m-h+1} : x_m\}$, where $h \leq m$.

Under definition 2 and 3, a streaming time series can be divided into a group of complete subsequences and one incomplete subsequence. In other words, a streaming time series can be expressed as $T = \{T_S, C\} = \{S_1, S_2, ..., S_i, C\}$, which means this streaming time series has i complete subsequences and one incomplete subsequence with semantic information. And T_S represents the complete subsequences group.

From these definitions, it follows that the subsequences are determined by segmentation algorithm. Different segmentation algorithm can generate different sets of subsequences with equal or unequal lengths.

5.4 Framework

To achieve a general framework of representation learning with semantic information for time series, it is crucial to generalize the process of representation learning. This involves developing methods and techniques that can effectively capture the underlying patterns and characteristics of diverse time series data.

The general framework of representation learning of time series with semantic information is shown in Figure. 5.2. The framework has three structural layers. First layer is time series pre-processing layer. This layer is designed to construct the input of training model. Basically, this layer consist with two components: semantic-based segmentation method and input normalization method. Semantic-based segmentation method is responsible for segmenting the input data

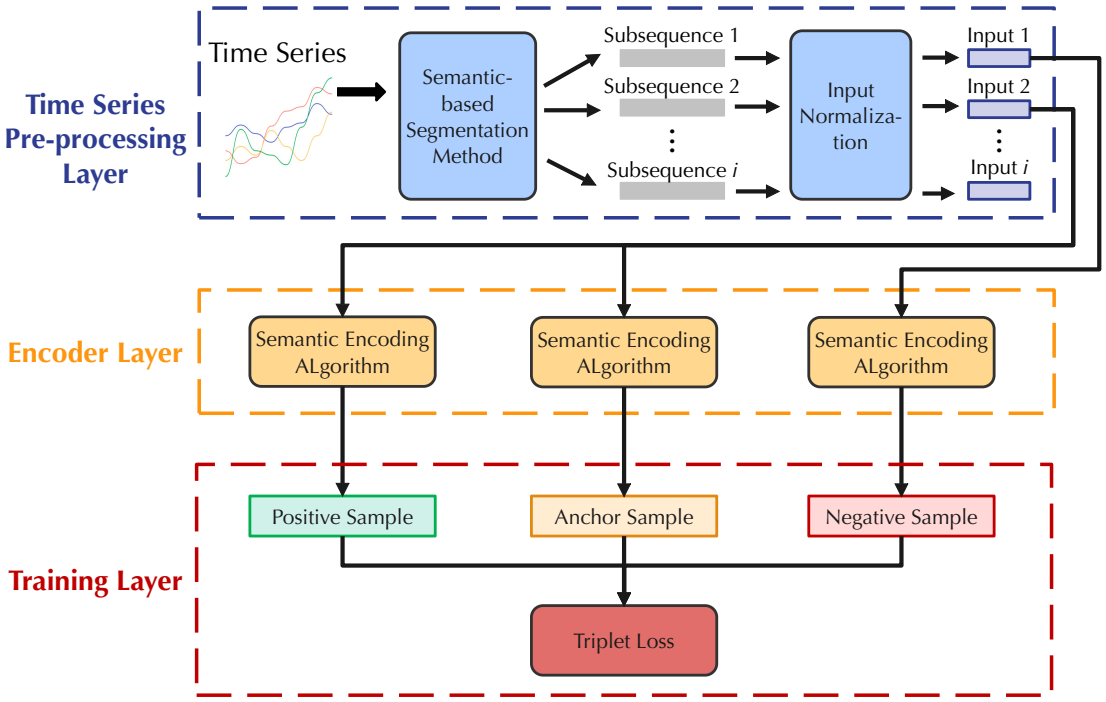


Figure 5.2. Framework structure of representation learning of time series with semantic information.

into meaningful segments with semantic information. It aims to identify and extract relevant patterns or structures in the data. Meanwhile, input normalization method focuses on normalizing the input data. It aims to re-scale subsequences to common scale, making it easier for training.

The second layer, encoder layer, serves as the core component of the framework. The encoder layer takes in the pre-processed input data and learns the compressed and meaningful representation. It can extract relevant features and capture the underlying patterns in time series, enabling effective representation learning. More importantly, with encoding algorithm designed in this layer, the semantic information in the time series subsequences is extracted and embedded in the representation results.

The third layer is the training layer. In this framework, triplet network is selected as the training network. In representation learning, triplet network is a popular choice for training models. It aims to make similar samples closer to each other, while dissimilar samples are farther apart. This is achieved by utilizing anchor sample, positive sample (similar to the anchor), and negative sample

(dissimilar to the anchor). The objective is to optimize the model's embedding space, ensuring that samples with similar semantic information are mapped closer together in the latent space.

This framework is designed for learning the representation of time series with semantic information. Actually, the model structure of some existing research that proposed for subsequence-level representation learning with semantic can also be summarized by this framework. For example, GP-HLS [82] and ShapeNet [64] are two novel representation learning models with semantic feature for multivariate time series. The structures of GP-HLS and ShapeNet are consistent with our proposed framework. This suggests that our proposed framework can be used as a paradigm for representing the semantic information of time series.

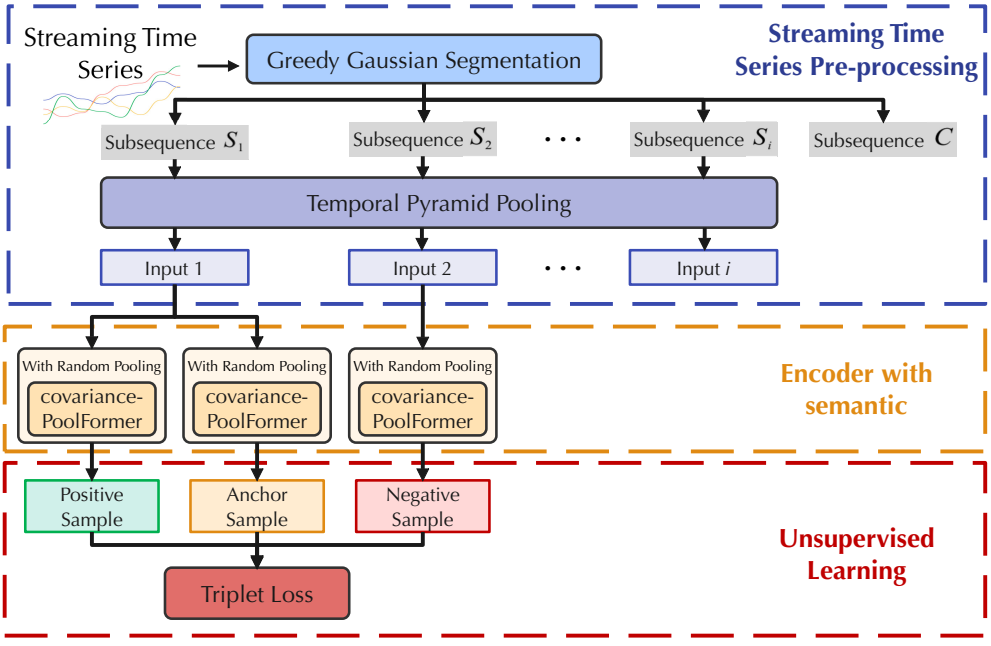


Figure 5.3. Structure of unsupervised representation learning for streaming time series with semantic information.

5.5 Methodology

5.5.1 Overview

In this section, the proposed CPFormer model structure and relevant algorithms are described. The structure of CPFormer is shown in Figure. 5.3.

CPFormer is based on framework mentioned in Section 5.4, designed for unsupervised representation learning with semantic of streaming time series. Firstly, the greedy Gaussian segmentation (GGS) method [67] is applied to generate subsequences with semantic. Basically, GGS aims to divide the whole time series into several regions where the within-segment data points exhibit higher Gaussian likelihood compared to the between-segment data points. Meanwhile, GGS algorithm is particularly suitable for segmentation of streaming time series by its ability to iteratively add breakpoints. In addition, a widely used input normalization method is temporal pyramid pooling (TPP) [68], which is designed to generate regular inputs for time series subseries with unequal lengths.

It is worth noting that the incomplete subsequence C does not go through the TPP normalization method. Because the length and Gaussian distribution of incomplete subsequence has not been determined, which means it can not join the subsequent learning processing. With the gradual incoming of streaming time series, incomplete subsequence C becomes a complete subsequence S_i, it will go through the TPP and join the subsequent learning processing. This iterative process is designed to align with the requirements of micro-batch operations in streaming time series, which are commonly used in various stream time series models [90].

5.5.2 Covariance-based PoolFormer Mechanism

As introduced in section 5.4, the encoder layer is core component of the model. The original encoder architecture of PoolFormer, as depicted in Figure. 5.4 (a), replaces the self-attention mechanism with the Pooling mechanism to address the challenges of trainable parameters and computational complexity. This substitution has resulted in noticeable improvements and positive outcomes. However, Pooling mechanism cannot represent subsequences with semantic information. It can only capture similarity information among different inputs.

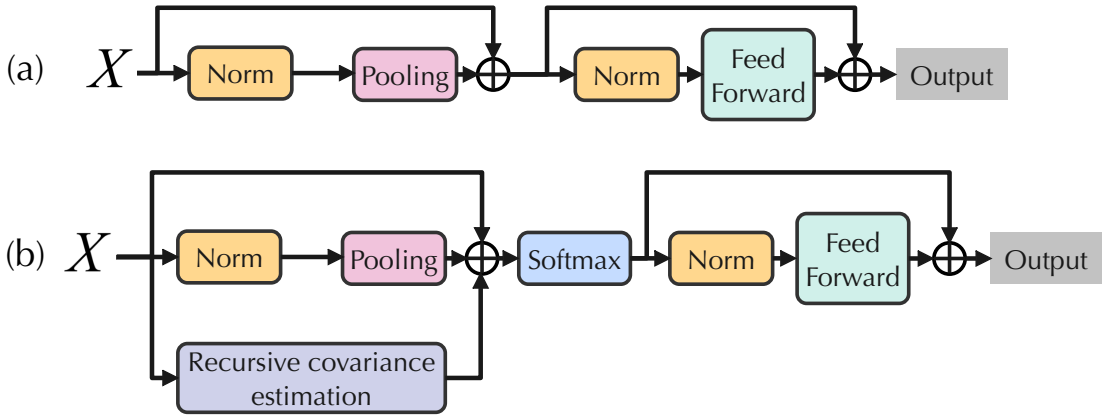


Figure 5.4. Schematic of Pooling-based Transformer architecture: (a) the architecture of PoolFormer; (b)the architecture of proposed CPFormer.

Covariance has already been proved to reveal the semantic information in time series. According to the characteristic of streaming time series, recursive covariance estimation is considered. Recursive covariance estimation is a technique used to estimate the covariance matrix in an iterative manner. It is particularly useful when dealing with streaming data where new observations are continuously added. The structure of proposed CPFormer is shown in Figure. 5.4 (b). As same as PoolFormer, the encoder layer of CPFormer is consist of two sub-block. The first sub-block is designed to calculated the Interaction information of subsequences. The second sub-block is considered to generate the representation.

According to the expression in original PoolFormer paper, the calculated result $Y(PoolFormer)$ of first sub-block can be expressed as:

$$Y(PoolFormer) = Pooling(Norm(X)) + X \tag{5.1}$$

where X represents inputs of complete subsequences group T_S. After adding recursive covariance estimation, covariance is considered in the first sub-block to reveal the semantic information. Therefore, the calculated result $Y(CPFormer)$ of first sub-block in proposed CPFormer can be expressed as:

$$Y(CPFormer) = Pooling(Norm(X)) + Cov(X) + X \tag{5.2}$$

where $Cov(X)$ is calculated by recursive covariance estimation.

Therefore, the representation output of CPFormer can be expressed as:

$$output = \sigma(Norm(Softmax(Y(CPFormer)))W_1)W_2 + Y(CFPormer) \quad (5.3)$$

where W_1 and W_2 are learnable parameters; $\sigma(\cdot)$ is a non-linear activation function.

5.5.3 Stochastic Pooling-based Unsupervised Training

In this section, we present a simple unsupervised training approach. Based on the structure of CPFormer, we employed stochastic Pooling [91] as Pooling component in CPFormer to generate training pairs. In Stochastic Pooling the pooling operation randomly samples values from the pooling window according to a probability distribution. This probabilistic sampling introduces a level of randomness into the pooling process, which meets the requirement of construction of training pairs in unsupervised representation learning.

A diagram of stochastic Pooling in triplet network of proposed CPFormer is shown in Figure. 5.5. Stochastic Pooling is applied to the Pooling component in PoolFormer. Positive sample pairs are constructed by two stochastic Pooling operation for one streaming time series subsequence. These pairs represent similar patterns or instances within the same streaming time series subsequence. For negative sample pairs, subsequences are randomly chosen from different streaming

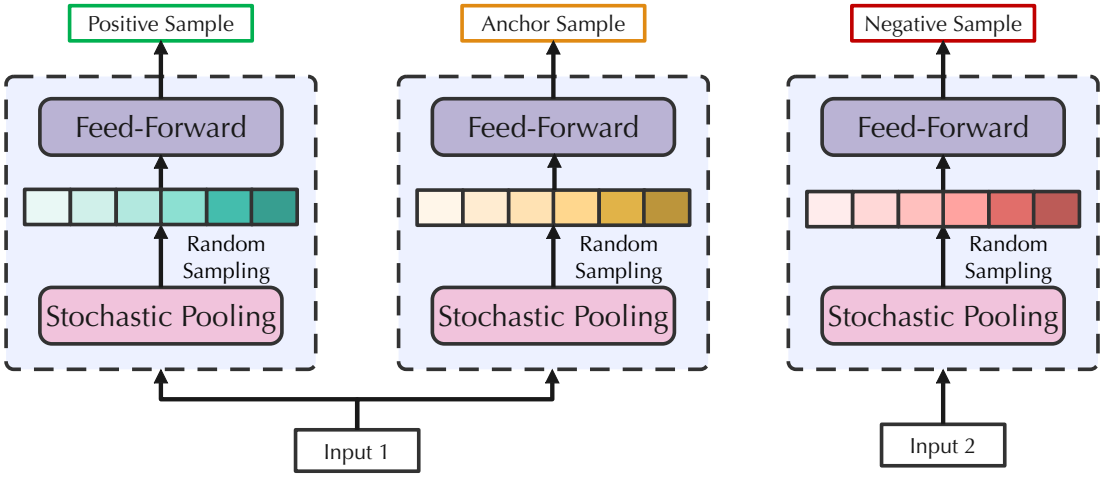


Figure 5.5. Schematic of generating training pairs for triplet network in representation learning of streaming time series.

time series in the dataset. These pairs represent dissimilar patterns or instances from different streaming time series subsequences.

The original training objective of the triplet loss is calculated based on the distances between the positive, anchor, and negative samples. It can be expressed as follows:

$$max(d(a,p) - d(a,n) + margin, 0) \tag{5.4}$$

where $d(a,p)$ denote the distance between the anchor sample (a) and the positive sample (p), and $d(a,n)$ denote the distance between the anchor sample (a) and the negative sample (n). Margin is a hyperparameter that specifies the desired separation or margin between the distances of positive and negative samples. Specifically, based on the cosine distance, the training objective can be defined as follows:

$$\log\left(1 + \exp\left(margin - \cos(a,p) + \cos(a,n)\right)\right) \tag{5.5}$$

5.6 Experiments

In this section, to evaluate our proposed CPFormer more objectively, we examined CPFormer with other algorithms of streaming time series in terms of downstream task of classification and retrieval. Meanwhile, as an important metric for streaming time series analysis, execution time is used as a part of evaluation.

5.6.1 Experimental Setup and Datasets

To generate streaming time series from offline public datasets of time series, we employed Spark streaming [92] to simulate streaming time series. Spark Streaming is a real-time stream processing framework in Apache Spark that enables high-throughput, fault-tolerant processing of live data streams. It allows to develop streaming environment for simulating streaming time data and testing model performance under streaming situation.

As for the time series datasets, we utilized datasets from the UEA&UCR time series classification archives [54]. These datasets were chosen due to their diversity across different domains. The UEA&UCR archives offer an initial benchmark for existing models, providing valuable baseline information on their performance.

Table 5.1. Summary of UEA&UCR datasets in classification task.

Dataset	Train Size	Test Size	Length	Classes
CBF	30	900	128	3
FaceUCR	200	2050	131	14
GunPoint	50	150	150	2
Plane	105	105	144	7
SyntheticControl	300	300	60	6
TwoPatterns	23	1139	82	2
TwoLeadECG	1000	4000	128	4
Wafer	1000	6164	152	2

5.6.2 Classification

In the classification task, to obtain a distribution over classes from the model's output vector, we applied a Softmax function. The cross-entropy between this distribution and the categorical ground truth labels was then calculated as the sample loss.

Following eight datasets from UEA&UCR time series classification archives were chosen to evaluate model performance. These datasets were selected because they are also utilized in evaluation of some advanced algorithms of streaming time series [93]. Table 5.1 provides a summary of these datasets.

Meanwhile, we have selected the following three advanced models as our baseline:

- GP-HLS [82]: It is a unsupervised representation learning algorithm for time series. GP-HLS uses semantic information to represent subsequences of time series, which is consistent with proposed framework in section 5.4.

- ODTW-NN [94]: This research presents a online dynamic time warping (ODTW) for streaming time series. It passively adapts to event changes using a memory forgetting mechanism.

- PED [93]: This study introduces an active adaptation strategy for time series classifiers, which enables them to adjust in real-time to the evolving nature of streaming time series.

Among these three baseline models, GP-HLS is proposed for offline time series. PED and ODTW-NN are designed for streaming time series. Table 5.2 presents the classification results for the streaming time series, where bold indicates best values.

In general, CPFormer model can hold best result in five datasets. Compared with GP-HLS, our model get a better rank, though GP-HLS has a better performance in accuracy. Comparing the experimental results of the four algorithms, we conclude that the representation considering semantic information, GP-HLS and our propose CPFormer, has a better performance than those general models. The results of the SyntheticControl and TwoPatterns data indicate a relative weakness of our model when dealing with shorter time series. Meanwhile, our model is relatively more advantageous for those datasets with longer length than baselines.

Meanwhile, execution time is also a significant evaluation metrics in classification of streaming time series. Figure. 5.6 shows the training time (seconds) of eight datasets in three representation learning models in classification experiment: GP-HLS, PED and our proposed CPFormer. Among three methods, CPFormer provides the shortest training time. Because CPFormer applies iterative training approach, the efficiency of representation learning has been greatly improved. In summary, our proposed CPFormer model has same accuracy performance as advanced GP-HLS models, with less runtime. This makes CPFormer more suitable

Table 5.2. Classification accuracy results of proposed and other methods.

Dataset	GP-HLS	ODTW-NN	PED	CPFormer
CBF	0.79	0.63	0.76	**0.81**
FaceUCR	**0.45**	0.20	0.39	**0.45**
GunPoint	0.73	0.53	0.74	**0.76**
Plane	0.77	0.47	0.73	**0.80**
SyntheticControl	0.81	0.30	**0.82**	0.74
TwoPatterns	**0.75**	0.68	0.69	0.68
TwoLeadECG	0.45	0.40	0.47	**0.50**
Wafer	**0.57**	0.51	0.55	0.55
Average Accuracy	**0.67**	0.46	0.64	0.66
Average Rank	1.8	3.8	2.2	**1.6**

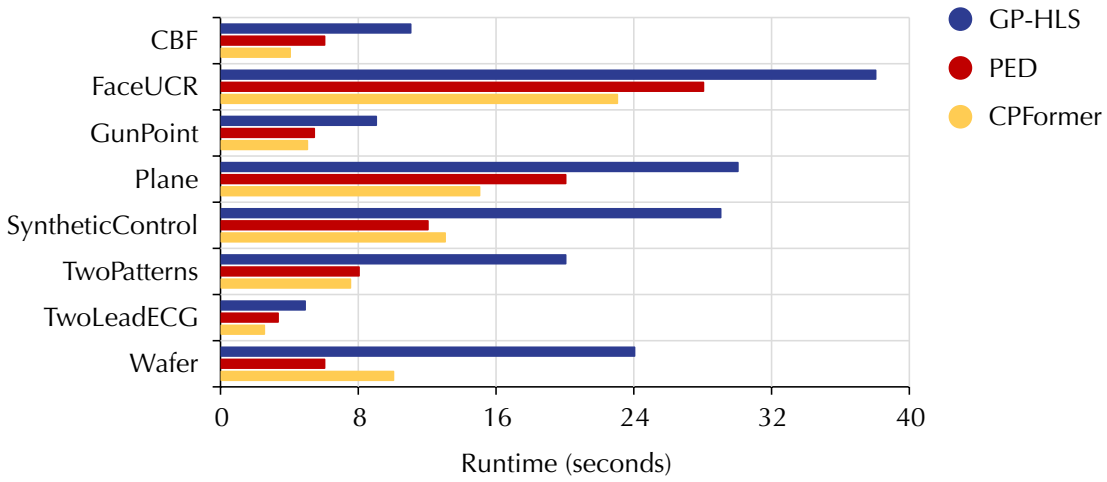


Figure 5.6. Runtime of eight datasets in three representation learning models.

Table 5.3. Summary of UEA&UCR datasets in retrieval task.

Dataset	Train Size	Test Size	Length	Classes
ElectricDevices	8926	7711	96	7
ECG5000	500	4500	140	5
FordA	3601	1320	500	2
Worms	181	77	900	5
ShapesAll	600	600	512	60

for handling the dynamic nature of streaming time series.

5.6.3 Retrieval

For the time series retrieval task, we evaluate the effectiveness of the proposed model for streaming time series retrieval tasks based on five different datasets. Same as described in the section 5.6.2 of classification tasks, these five datasets were selected because they are utilized in evaluation of some advanced retrieval algorithms of streaming times [95]. Table 5.3 presents the details of these datasets.

We compared our model with three advanced baseline methods in streaming time series retrieval:

- multi-step filtering mechanism (MSM) [96]: MSM is used to perform similarity matching over streaming time series. This mechanism allows for the reduction of the search space, leading to faster response times.

- multi-resolution search scheme (MRSS) [95]: This is a variants of MSM, which is based on multi-resolution filtering to perform the similarity search in streaming time series.

- Efficient multi-resolution representation (EMR) [97]: EMR proposes a multi-resolution filtering scheme for incrementally calculating the similarity distance among sequence patterns of streaming time series.

According to the descriptions in baseline research MRSS and EMR, for streaming time series, the objective of retrieval task is to rapidly identify all subsequences in the time series stream data that match the given query sequence. In this context, the retrieval time serves as a metric to evaluate the performance of the search. In addition, both the baseline model and our proposed model select Euclidean distance for evaluating the similarity. The experimental results are shown in Table 5.4, where bold indicates shortest retrieval time. Obviously, compared with baselines methods, our CPFormer model has better performance in retrieval task of streaming time series.

5.7 Conclusion and Future Work

Semantic information is indeed crucial in the representation learning of streaming time series. It allows the model to capture meaningful patterns and relationships within the data. In addition, the iterative training method has proven to be beneficial in the context of streaming time series representation learning. Our proposed CPFormer algorithm combines these two important aspects. In this study, a Covariance-based Pooling mechanism was introduced for representation learning of streaming time series. Meanwhile, stochastic Pooling-based triplet

Table 5.4. Retrieval time of of proposed and other methods (millisecond).

Dataset	MSM	MRSS	EMR	CPFormer
ElectricDevices	4016	**1950**	2001	2011
ECG5000	1369	1224	1230	**1206**
FordA	4330	2745	2566	**2108**
Worms	1442	1452	1410	**1393**
ShapesAll	4125	2102	2527	**1993**

network is designed for unsupervised training of streaming time series. The experiments show that the proposed model demonstrates significant improvement in multivariate time series representation learning.

In future research, we will focus on developing a more comprehensive framework for streaming time series that addresses various aspects such as storage, management, and mining tasks. Our goal is to create an integrated solution that efficiently handles the challenges associated with streaming time series data.

CHAPTER 6

Cross-Granularity Representation Learning

6.1 Introduction

Time series is a traditional and important type of data that is ubiquitous in numerous fields. Significant progress in the widespread use of sensors and social production activities has further promoted the development of time series data such as electrocardiograms (ECG) [98] and daily stock prices [99]. With the development of machine learning and data mining, representation learning, which can reveal hidden information in time series by establishing high-dimensional representations, has been increasingly applied to the field of time series.

However, despite the recent challenges and advancements made by deep learning models in tasks such as prediction and classification, the dominant position of representation learning methods in time series has yet to be established, in contrast to fields such as computer vision (CV) [100] and natural language processing (NLP) [101]. In particular, non-deep learning methods, such as HIVE-COTE [102] and TS-CHIEF [103], provide unique advantages.

Although multiple time series representation methods achieve adequate results on public datasets, in real-world application scenarios, time series data are generally subject to missing data, noise data, and data confusion, among other adverse

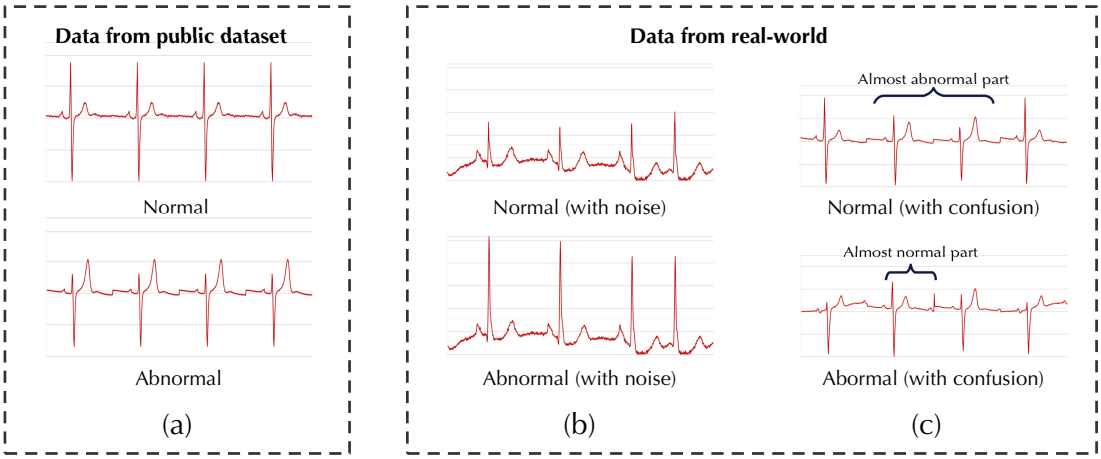


Figure 6.1. Example of ECG data from public dataset and real-world. These figures present several issues of data quality in real-world ECG data.

conditions. Figure. 6.1 presents a typical example of this issue in ECG data. In public datasets designed for model training, the ECG data contain more typical class features (normal and abnormal), without noise or confusion (as shown in Figure. 6.1(a)). However, in practice, the time series data quality is different. Noise is a common problem in real-world applications, as illustrated in Figure. 6.1(b), which can have several negative effects on data analysis, including reduced accuracy and misleading conclusions. In addition, data confusion is a more significant issue, as it can considerably impact the effectiveness and accuracy of data analysis. Data confusion refers to cases wherein data from different categories, sources, or contexts is mixed or entangled, thus making it difficult to discern clear patterns, relationships, or structures within the data. Considering the ECG data as an example, overlapping or ambiguous morphologies frequently appear in real-world data (as shown in Figure. 6.1(c)). Electrocardiograms data from different cardiac conditions may exhibit similar or overlapping morphologies, which makes it challenging to distinguish between them. For example, certain types of arrhythmias may appear similar to a normal sinus rhythm, thus leading to data confusion.

To address this issue, multiple studies comprehensively considered representations of time series at different granularity, i.e., multi-granularity methods [104]. An simple example of a multi-granularity method is sales reports that includes data at both the individual transaction and aggregate levels such as monthly

or yearly totals. By capturing information from multiple scales or levels of detail, these approaches improve the robustness and accuracy of the analysis and interpretation of time series data. Although multi-granularity representations provide more information, information redundancy is generally observed between different granularities. This redundancy can potentially lead to increased computational complexity, and render the analysis and interpretation of time series data more challenging. Moreover, numerous existing multi-granularity methods are focused primarily on the simple fusion of decision results, and generally require the re-design of representation models. Consequently, they cannot utilize existing, well-performing representation methods and lack the flexibility to adapt to different scenarios.

This paper proposes a novel unsupervised learning framework named MUG (for **MU**lti-**G**ranularity), which combines the multi-granularity features of time series based on existing representation learning research. The proposed general framework integrates two different granularities of time series representation methods: a fine-grained representation method, which represents timestamp-level time series data, and a coarse-grained representation method, which represents segment-level time series data. Specifically, for the multiple fine-grained time series representation results, we employed a vector fusion method based on attention mechanism to obtain a comprehensive representation. In addition, based on multi-modal fusion techniques, we employed a cross-granularity attention mechanism to map of coarse-grained representations onto fine-grained representations. This allowed for the fusion of the overall features in the coarse-grained representations with the detailed information in the fine-grained representations. Finally, based on the retrieval task, we designed a more suitable training method for the multi-granularity time series representation learning.

The main contributions of this study are as follows:

- This paper presents a focused study on the transformer-based fusion model of multi-granularity representation for time series data. In particular, this paper proposes a novel unsupervised learning framework (Section 6.3.1) to build association between timestamp-level and segment-level features.

- We developed an unsupervised training method (Section 6.3.3). In particular, a retrieval task for the time series data with a unique loss function was designed to obtain the comprehensive multi-granularity representation

of time series via unsupervised training.

- We conducted extensive experiments on several public datasets from different fields and real-world datasets (Section 6.4). In comparison with other baseline algorithms, the proposed MUG model achieved an improved performance.

The remainder of this paper is organized as follows: Section 6.2 outlines previous studies on representation learning for time series, in addition to multi-granularity representation methods for time series from the existing literature. Section 6.3 presents the architecture of the proposed framework in detail. Thereafter, Section 6.4 presents the experimental results, followed by a summary of conclusions in Section 6.5.

6.2 Related Work

6.2.1 Multi-granularity representation learning of Time Series

The representation learning of time series data has attracted considerable research attention in recent years. The primary objective of these models is to identify spatio-temporal dependencies in the data, which can help uncover the underlying patterns, trends, and relationships that can be used for various tasks, such as forecasting, classification, and anomaly detection.

According to representation granularity, the existing representation learning models of time series can be broadly classified into two categories: coarse- and fine-grained representation methods. The differences between the two types are shown in Figure. 6.2.

Fine-grained representation, i.e., timestamp-level representation learning, is the most traditional concept for the representation learning of time series. The objective of this method is to capture the relationships and dependencies between the different dimensions of the time series data at each point in time. Time2Vec (T2V) [105] is a typical timestamp-level representation learning method developed to capture temporal patterns and dependencies within the data. This method is based on Word2Vec [106]. However, T2V may require detailed hyperparameter tuning to achieve an optimal performance. Selecting the appro-

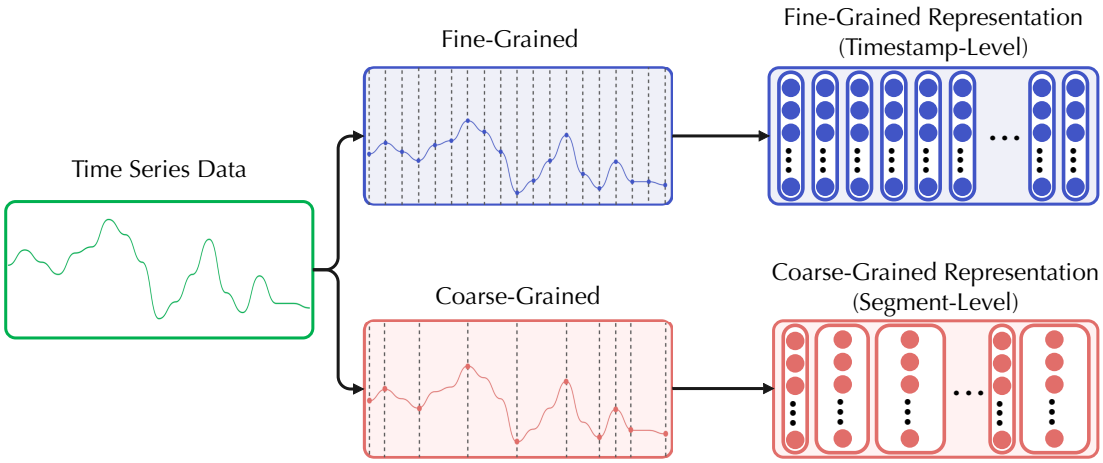


Figure 6.2. Main differences between fine-grained representation learning and coarse-grained representation learning of time series.

priate dimensions for continuous vector representations, an appropriate learning rate, and determining the appropriate context window size can be challenging and time-consuming. Compared with T2V, The Time Series Transformer (TST) model [24] provides more advantages. The TST model is a deep learning-based approach for time series analysis that leverages the transformer architecture [38], originally designed for NLP tasks. The transformer architecture is known for its self-attention mechanism [69], which can capture complex dependencies and patterns within sequences. The TST model can be used for various time series tasks, such as forecasting, classification, anomaly detection, and feature extraction.

Coarse-grained representation is referred to as as segment-level representation learning, i.e., learning representations for segments or subseries within an entire time series. These methods are focused on capturing global patterns and long-range dependencies in time series data, which can be beneficial for various tasks wherein the focus is on understanding local patterns and range dependencies in the data. The symbolic aggregate approximation (SAX)-based method [107] is a widely-used method for time series data representation and dimensionality reduction. In particular, it converts a continuous-valued time series into a discrete, symbolic representation while preserving the essential shape and trends of the original data. The SAX-based method can reduce the storage requirements with lower computational complexity. Additionally, the SAX-based method can be readily extended or combined with other techniques, such as indexable SAX

(iSAX) [108] or multivariate SAX (MSAX) [109]. However, the dimensional reduction and discretization process of the SAX-based method may result in information loss. The Shapelet-based methods, such as ShapeNet [64], may be the most advanced segment-level representation learning method. These techniques are focused on identification of discriminant sub-sequences in time series data, which can be useful for tasks such as classification and anomaly detection. However, the computational complexity of shapelet discovery can be high, particularly for large datasets and long time series.

6.2.2 Multi-Granularity Representation Methods for Time Series

Both coarse- and fine-grained representation learning have advantages and applicability scenarios that render them suitable for different types of time series analysis tasks. Within this context, the majority of existing studies were focused on a single granularity, and methods are developed based on a specific level of detail in time series data with the objective of predicting the labels corresponding to the granularity.

However, in general, selecting the appropriate granularity for different tasks is a challenge that significantly depends on experience. Multi-granularity representations allow for information to be obtained from various perspectives within time series data, thus providing a more comprehensive understanding of the underlying patterns and structures. For example, in the analysis of stock market data, fine-grained representation learning methods can analyze high-frequency data such as intraday price movements. This helps to identify short-term trends and patterns. Coarse-granularity representations, such as daily or weekly price movements, can be useful for identifying long-term trends and patterns in the stock market, such as the overall market direction, support and resistance levels, and seasonal trends. Therefore, an increasing number of previous studies [110] [111] were focused tend on multi-granularity representation learning.

The multi-granularity substructure-aware representation learning algorithm for time series (MS-SRALAT [112]) is an advanced semantic representation of a symbol sequence that is generated corresponding to a time series by an approximation algorithm that can capture the structure of the original data. In particular, it is a quite concise and easily implementable method that utilizes the SAX and pro-

duces the representation of a time series by transforming the target time series into an SAX sentence and aggregating those embeddings of the SAX words in the SAX sentences. However, the SAX information in this framework cannot reveal meaningful semantic information, which limits its performance.

6.3 Methodology

6.3.1 Overview

This section presents the proposed MUG framework and the relevant algorithms are described. The structure of the MUG is shown in Figure. 6.3. Each training sample $X \in \mathbb{R}^{w \times m}$, which is a time series of length w and m different variables, constitutes a sequence of w time series $x_t \in \mathbb{R}^m : X \in \mathbb{R}^{w \times m} = [x_1, x_2, ..., x_w]$. Moreover, for each segment $S_i \in X$ in the time series, $S_i = [x_{i,1}, x_{i,2}, ..., x_{i,j}]$, which implies that segment S_i has j timestamp points in the time series.

First, for each segment S_i, the proposed framework employs two different representation learning algorithms for the coarse- and fine-grained time series data, thus constructing two distinct feature vectors. Using both granularities, the ob-

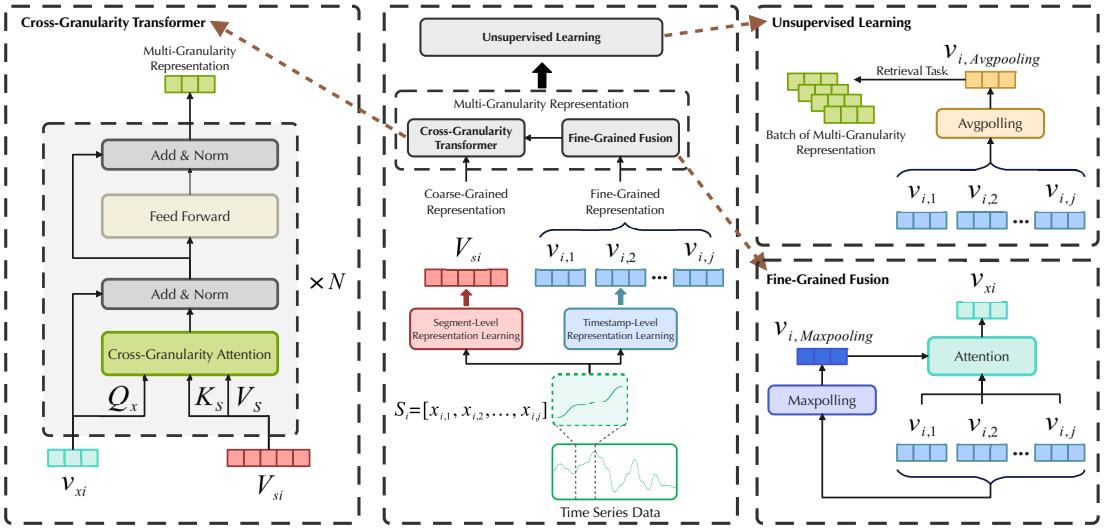


Figure 6.3. **Middle:** The structure of unsupervised multi-granularity representation learning for time series. **Left:** Details of cross-granularity transformer. **Right:** Details of the fine-grained fusion and retrieval-based unsupervised learning.

jectives of the model is to capture the different levels of the information present in the time series data, thus providing a more comprehensive representation. Fine-grained representations focus on local patterns and detailed information within the data, whereas coarse-grained representations capture the high-level patterns and global structures in the data. Thereafter, for the fusion of fine-grained representation of time series, a variant of the attention mechanisms was employed to combined the features of each timestamp-level representation of the time series, and generate a more comprehensive representation vector to represent the fine-grained information in certain segments of the time series. Moreover, for coarse-grained representations, a cross-granularity transformer with cross-granularity attention mechanism was employed to map coarse-grained representations onto fine-grained representations. Finally, with focus on the demand for unsupervised learning in multi-granularity representation learning, a retrieval-based task was selected as the training task for unsupervised learning. Based on the characteristics of the retrieval task, a novel loss function was designed to improve the performance of the training model.

6.3.2 Fine-Grained Fusion

The structure of the fine-grained fusion is shown at the bottom right of Figure. 6.3. This part is based on a variant of the attention mechanism, which was first designed as an NLP model for multi-granularity relation extraction [113]. This type of attention mechanism helps the model combine the feature information from each inputs, which is suitable for the multi-granularity representation learning framework, in the stage of representing the comprehensive feature vector of the fine-grained representation learning of time series.

Based on timestamp-level representation learning methods, the values of timestamp points can be embedded into a fine-grained representation, which can be formalized using Equation (1).

$$v_i = \{v_{i,1}, v_{i,2}, ..., v_{i,j}\} = f_{encoder}(x_{i,1}, x_{i,2}, ..., x_{i,j}) \tag{6.1}$$

Where $v_i = v_{i,1}, v_{i,2}, ..., v_{i,j}$ are the representation vectors of timestamp-level inputs. Index i indicates that these timestamp points are from Segment S_i in the time series.

Moreover, as in the original research, to built a comprehensive representa-

tion vector without any external information, a maximum pooling operation should be employed to obtain the shallow features of each timestamp-level inputs. $v_{i,Maxpooling} = Maxpooling(v_i)$.

Thereafter, to capture the comprehensive information of the fine-grained representations inputs of the time series, the fine-grained fusion part combined the timestamp-level feature and the maximum pooling value of each timestamp-level inputs. Specifically, the maximum pooling representation of these timestamp points can be used as the Query vector in the attention mechanism to obtain the fusion feature of fine-grained representation by Equation (2).

$$v_{xi} = Sfotmax\left(\frac{Q \cdot K^T}{\sqrt{d}}\right) \cdot V = Softmax\left(\frac{v_{i,Maxpooling} \cdot v_i}{\sqrt{d}}\right) \cdot v_i \tag{6.2}$$

Where d denotes the dimension of the representation vector and is used to normalize the vectors. In the remainder of this paper, $d_{(\cdot)}$ is used to represent the dimension of representation vector.

After the fine-grained fusion, the comprehensive representation vector of fine-grained representation learning is computed, which is employed to calculate multi-granularity representation in the subsequent steps.

6.3.3 Cross-Granularity Transformer

Cross-granularity representation is the subsequent step in the proposed framework. Unlike the fusion of fine-grained representation learning to obtain a comprehensive vector, cross-granularity representation has its own challenge.

Cross-granularity representation, which refers to the combination of coarse- and fine-grained information in a unified framework, are generally subject to redundancy. There may be overlapping or redundant information between the different granularities, thus leading to inefficiencies in the representation and potential over-fitting in the learning process. Additionally, complexity is a critical issue. Combining features from different granularities increases the complexity of the model, thus potentially increasing the computational requirements and training time. In addition, determining the optimal method for the fusion or integration of features from different granularities to generate a cohesive representation that effectively captures the underlying patterns in the data can be challenging. Therefore, multi-granularity feature fusion has attracted significant attention with respect to multi-granularity representation.

As mentioned previously, most existing models primarily focus on the simple fusion of decision results and generally require the re-design of representation models. Consequently, they cannot utilize existing, well-performing representation methods and lack the flexibility to adapt to different scenarios. To address this issue and more extensively utilize various existing excellent time series representation learning methods, we designed a cross-granularity transformer architecture based on the cross-granularity attention mechanism. The structure of the cross-granularity attention mechanism is shown in Figure. 6.4.

To introduce the cross-granularity attention mechanism, we considered two representation vectors, namely, v_{xi} and V_{Si} from fine- and coarse-grained representation learning, respectively, where $v_{xi} \in \mathbb{R}^{d_x}$ and $V_{Si} \in \mathbb{R}^{d_S}$. Based on the transformer architecture of multi-modal data fusion [114], we hypothesized that a suitable method for the fusion of cross-granularity information is to provide a latent adaptation across multi-granularity. In the proposed framework, it means V_{Si} to v_{xi} (coarse- to fine-grained).

We defined the Query as $Q_x = v_{xi}W_{Q_x}$, which is a linear transformation of the fine-grained representation input, the Key as $K_S = v_{Si}W_{K_S}$ and the Value as $V_S =$

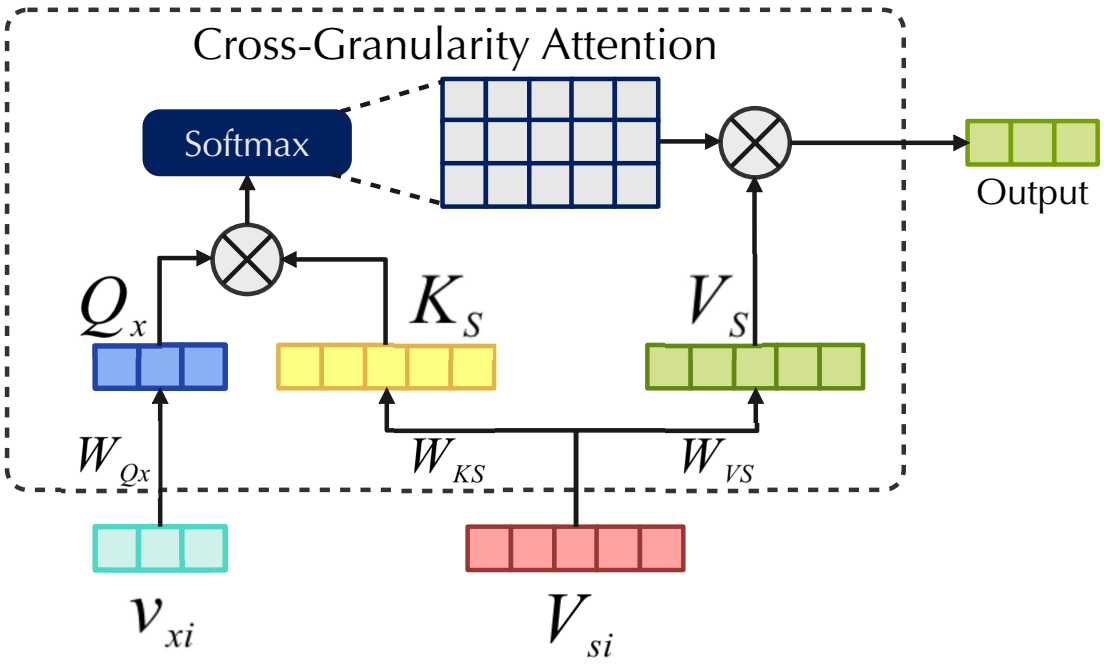


Figure 6.4. Structure of cross-granularity attention mechanism.

$v_{Si}W_{V_S}$, which are the line transformations of the coarse-grained representation input. Moreover, W_{Q_x}, W_{K_S} and W_{V_S} are the weights. The latent adaptation from V_{Si} to v_{xi} is presented as the cross-granularity attention as follows:

$$Attention(Q_x, K_S, V_S) = softmax\left(\frac{Q_x K_S^T}{\sqrt{d_k}}\right) V_S \tag{6.3}$$

The output of the cross-granularity attention mechanism has the same length as v_{xi}. Using this equation, the mapping of coarse- onto fine-grained representations was established.

6.3.4 Unsupervised Learning

The primary objective of unsupervised learning is to learn useful features or representations from the data without using any labeled information. This can be particularly beneficial for time series data analysis because obtaining labeled data can be time-consuming and expensive.

However, the design of algorithm for unsupervised learning is challenging. Several of these difficulties can be attributed to a lack of labeled data, which indicates that the model should identify the underlying structure and relationships within the data without any explicit guidance. However, constructing positive and negative sample pairs for unsupervised training is difficult. The construction of sample pairs is closely related to the selection of the unsupervised training tasks. In this framework, we designed an unsupervised training tasks based on retrieval task.

There were several reasons for us for selecting retrieval task. First, unlike other unsupervised learning models, the proposed MUG is required to accomplish multi-granularity representation vector fusion during the training process, which indicates that constructing positive and negative sample pairs before training is different. The representation vectors of the positive and negative samples are not in the same vector space as the anchor. Therefore, traditional unsupervised contrastive learning methods based on similarity measures cannot be applied in this scenario. Constructing positive and negative sample pairs during the training process is undoubtedly complex and time consuming. In addition, traditional loss functions are subject to several limitations. If the selected triplets are not informative, the triplet loss may rapidly converge to zero, thus leading to the degradation of the model performance.

To solve these issues, we designed an unsupervised learning method using a retrieval training task and applied a novel loss function in the training (as shown at the top right of Figure. 6.3). First, because we used the maximum pooling method in fine-grained fusion, we applied the average Pooling (Avgpooling) method to construct the query vector in the retrieval task. Thereafter, the multi-granularity representation corresponds to the query vector, in addition to other randomly selected multi-granularity representation vectors form the query object together. Assuming that the query vector is y_q, the correct multi-granularity representation corresponding to the query vector is y_t, and the other multi-granularity representation vectors are y_j. The ranking of y_t can be expressed as follows:

$$1 + \sum I\left(\|H(y_q) - H(y_t)\| \geq \|H(y_q) - H(y_j)\|\right) \tag{6.4}$$

where $H(Z)$ is a representation vector value of Z, and $I(a \geq b)$ is a function that is transformed to 1 when $a \geq b$ and 0 in other cases. In the above expression, ranking is used to describe the relative distances.

To convert the ranking situation into a similarity metric that can be used as a loss function, the Spearman correlation coefficient [115] was used to calculate the similarity. The Spearman correlation coefficient is a statistical measure that evaluates the strength and direction of the monotonic relationship between two variables. The equation of the Spearman correlation coefficient of y_t can be expressed as follows:

$$Similarity_t = \frac{n - \pi_t}{n - 1} \tag{6.5}$$

where π_t denotes the ranking of y_t. Using the Spearman correlation coefficient, the relative similarity was used in the loss function to accelerate model convergence and improve model accuracy. Moreover, the traditional loss function is applicable in such cases.

We did not use ranking losses [116] because we found that the binary classification loss [117] demonstrated a superior performance, which was similar to that reported in [118]. The equation of binary classification loss function is expressed as follows:

$$\mathcal{L}_{BCE} = -[ylog(\theta) + (1 - y)log(1 - \theta)] \tag{6.6}$$

where the ground truth labels $y \in (0, 1)$ and θ represent the similarity.

Therefore, by combining Equation 5 and 6, the novel loss function can be expressed as follows:

$$\mathcal{L}_{BCE} = -[ylog(\frac{n - \pi_t}{n - 1}) + (1 - y)log(1 - \frac{n - \pi_t}{n - 1})] \tag{6.7}$$

6.4 Results and discussion

As detailed in this section, we tested the effectiveness of the proposed framework by analyzing its performance on classification task, which was used as a downstream task, to prove the effectiveness of the proposed multi-granularity representation learning framework. Moreover, to highlight the advantages in real-world time series data, comparative experiments was conducted with other multi-granularity representation methods under simulated real-world scenario. Additionally, a case study was conducted to recall the example introduced in Section 6.1.

6.4.1 Classification

In the classification task, the output multi-granularity representation vector of the proposed framework was passed through a SoftMax function to obtain a distribution over the classes. TST is used with ShapeNet, which are introduced in Section 6.2.1, as the fine- and coarse-grained representation parts in our framework. In this task, we demonstrated that the proposed framework demonstrated a superior performance to those of other non-deep learning method and unsupervised methods.

We used the following ten multivariate datasets from the UEA time series classification archives [54], which provided multiple datasets from different domains with varying dimensions, unequal lengths, and missing values. We selected datasets from a diverse range of domains across science and engineering from Monash University, UEA & UCR Time Series Classification Repository. Selection was made to ensure diversity with respect to the dimensionality and length of the time series samples, in addition to the number of samples and classes (when applicable). Furthermore, we included both the "easy" and "difficult" datasets,

Table 6.1. Summary of UEA multivariate datasets.

Dataset	Train Size	Test Size	Length	Classes	Dimensions
EthanolConcentration	261	263	1751	4	3
FaceDetection	5890	3524	62	2	144
Handwriting	150	850	152	26	3
Heartbeat	204	205	405	2	61
JapaneseVowels	270	370	29	9	12
PEMS-SF	267	173	144	7	983
SelfRegulationSCP1	268	293	896	2	6
SelfRegulationSCP2	200	180	1152	2	7
SpokenArabicDigits	6599	2199	93	10	13
UWaveGestureLibrary	2238	2241	315	8	3

where the baselines performance were significantly high or low, respectively. A summary of these datasets is provided in Table 6.1.

The UEA archives provides an initial benchmark for existing models with accurate baseline information. Based on the performance metrics provided in the UEA archives, we selected the following three models as our baselines:

- Dimension-dependent dynamic time warping (DTW_D) [74]: it uses a weighted combination of raw series and first-order differences for the neural network classification with either Euclidean distance or full-window dynamic time warping (DTW). Additionally, it develops the traditional DTW method and suits every data series.

- ROCKET [57]: it is based on a random convolutional kernel similar to a shallow convolutional neural network. It can achieve rapid and accurate time series classification using random convolutional kernels.

- Long short-term memory (LSTM) model [119]: it is a type of recurrent neural network (RNN) architecture that is designed to overcome the limitations of traditional RNNs in capturing long-term dependencies in sequential data.

Table 6.2 presents the classification results for the time series, where bold values indicate the optimal values. As shown in Table 6.2, the proposed framework

Table 6.2. Accuracy results of the proposed and other methods.

Dataset	MUG (TST-ShapeNet)	TST	ShapeNet	DTW_D	ROCKET	LSTM
EthanolConcentration	**0.471**	0.337	0.312	0.323	0.452	0.323
FaceDetection	**0.694**	0.681	0.602	0.529	0.647	0.577
Handwriting	0.366	0.305	0.451	0.286	**0.588**	0.152
Heartbeat	**0.780**	0.776	0.756	0.717	0.756	0.722
JapaneseVowels	**0.997**	0.994	0.984	0.949	0.962	0.797
PEMS-SF	**0.919**	**0.919**	0.751	0.711	0.751	0.399
SelfRegulationSCP1	**0.945**	0.925	0.782	0.775	0.908	0.689
SelfRegulationSCP2	**0.615**	0.589	0.578	0.539	0.533	0.466
SpokenArabicDigits	**0.995**	0.993	0.975	0.963	0.712	0.319
UWaveGestureLibrary	0.905	0.903	0.906	0.903	**0.944**	0.412
Average Accuracy	**0.768**	0.742	0.710	0.669	0.723	0.486
Average Rank	**1.4**	2.4	3.1	4.5	2.9	5.3

demonstrated the highest performance on eight of the ten datasets, thus achieving an average rank of 1.4^{th}, followed by TST, which demonstrated one highest performance with average ranked 2.4^{th}. ROCKET, which demonstrated the optimal performances for the remaining two datasets, and on average, was ranked 2.9^{th}. From the data presented in the table, it can be concluded that the effectiveness of proposed framework significantly increased as the amount of data increased. In addition, comparing the performances of MUG (TST-ShapeNet), TST and ShapeNet, it is evident that multi-granularity representation can achieve a superior performance to that of the single-granularity representation method. This is because multi-granularity methods can capture complex temporal dependencies and patterns that may be presented at different scales or resolutions in the data.

6.4.2 Comparative Experiments

This section presents a discussion on the performance of proposed MUG and other multi-granularity models with respect to real-world time series data. Accordingly, we used several UCR archives [58] and randomly added Gaussian noise and segments of time series data from other classes to the time series data to simulate the real-world cases analyzed in Section 6.1. A summary of these datasets

Table 6.3. Summary of simulated real-world time series data from the UCR datasets.

Dataset	Train Size	Test Size	Length	Classes
Adiac	390	391	200	37
Beef	30	30	500	5
Fish	175	175	480	7
Gun-Point	50	150	170	2
CBF	30	900	160	3
Trace	100	100	300	4

is provided in Table 6.3.

It should be noted that the lengths of these six datasets are longer than the original lengths in UCR archives, which is due to the simulation of the real-world situations.

For comparison, we selected the MS-SRALAT framework introduced in Section 6.2.2. The ROCKET algorithm was used as a control group to further illustrate the performance difference between the single-granularity and multi-granularity methods. The results of the comparative experiments are presented in Table 6.4.

By analyzing the results in Table 6.4, the proposed framework obtained superior results to those obtained by MS-SRALAT. This is because of the more advanced fine- and coarse-grained representations selected in the proposed framework, in addition to the more fixable structure and improved fusion method. The proposed framework can therefore improve the accuracy of downstream tasks.

Table 6.4. Accuracy comparison between single-granularity and multi-granularity methods.

Dataset	MUG(TST-ShapeNet)	MS-SRALAT	ROCKET
Adiac	0.435	0.379	**0.468**
Beef	**0.614**	0.550	0.458
Fish	**0.758**	0.671	0.469
Gun-Point	**0.859**	0.701	0.647
CBF	0.900	**0.934**	0.887
Trace	**0.877**	0.860	0.713

Table 6.5. Summary of ECG200 and TwoLeadECG.

Dataset	Train Size	Test Size	Length	Classes
ECG200	100	100	96	2
TwoLeadECG	23	1139	82	2
Combined ECG	123	1239	82	2

Table 6.6. Accuracy results of proposed and other methods.

Dataset	MUG(TST-ShapeNet)	DTW_D	ST
ECG200	**0.930**	0.880	0.840
TwoLeadECG	**0.993**	0.868	0.984
Combined ECG	**0.800**	0.442	0.510

6.4.3 Case Study

In this case study, the example in Section 6.1 was considered more comprehensively to clarify the motivation for the study. As introduced in Section 6.1, by analyzing the characteristic of real-world ECG data, it can be concluded that the complex temporal dependencies and patterns may exist at different scales or resolutions in the data. This section presents experimental design to further address this issue.

To simulate real-world ECG data, we combined two other datasets from the same subject obtained from various sources. Specifically, we used the ECG 200 and TwoLeadECG as datasets from the UCR time series classification archive. Both datasets traced the recorded electrical activity and contained two classes: normal heartbeats and myocardial infarctions (MIs). We randomly combined these two datasets and reshaped the length of the combined ECG dataset to obtain a regular time series. The details of these datasets are presented in Table 6.5.

We selected DTW_D and Shapelet Transform [80] as the baseline algorithms. The Shapelet Transform (ST) is based on the shapelet method, which represent fine- and coarse-grained representations respectively. First, separate experiments were conducted using these two datasets. We then conducted an experiment using the combined dataset. The experimental results of this case study are listed in Table 6.6.

6.5 Conclusion and Future Work

In this study, we investigated the significance of exploring multi-granularity patterns for time series representation learning and proposed a multi-granularity framework for the unsupervised representation learning of time series. In particular, this paper proposes a novel unsupervised learning framework to build association between timestamp-level and segment-level features. To address the loss function issue in multi-granularity representation learning, a retrieval task for time series data with a special loss function was also designed. Experiments on public datasets and real-world data demonstrated the effectiveness of our MUG framework. In the future, we plan to employ more fine- and coarse-grained representation models in the proposed MUG framework to discuss the generality across different multiple granularity models. Moreover, we will focus on developing a more general framework that can combined more than two multi-granularity representation methods.

CHAPTER 7

Conclusion and Future Work

7.1 Conclusion

This study introduces several time series representation learning models with different granularities. These models can be applied in a variety of downstream tasks associated with time series data. Furthermore, given that a lots of time series data in real world is unlabeled, the value and practical relevance of unsupervised learning are accentuated. Our models are designed around the distinct features inherent in time series data, with an novel unsupervised learning approach, making them particularly well-suited for representation learning of time series data.

We initially develop two representation learning algorithms of different granularities based on the characteristics of time series data: timestamp-level and segment-level. These two levels of time series representation learning algorithms are adaptable to various time series datasets and are compatible with downstream tasks requiring different granularities of time series representation. Moreover, we expand the segment-level representation learning model to streaming time series, which extends its applicability to the field of stream data. In addition, we introduce a cross-granularity time series representation learning model, which is an innovative approach that combines the advantages of multi-granularity of representation.

timestamp-level representation learning: Timestamp-level representation learning focus on fine-grained representation, where we delve into fine-grained nuances of the data. Fine-grained representation learning is designed to capture subtle patterns and minute fluctuations over time, which can be critical for sensitive applications that require high-resolution insights. We introduce a specially designed local binary pattern method to the self-attention mechanism to improve the representation performance of modeling in terms of local information. Meanwhile, a novel unsupervised approach is designed to training the representation learning model. Experiments of classification and regression have been implemented to verify the effectiveness of our proposed approach in tasks that need fine-grained features.

segment-level representation learning: For segment-level representation of time series, another unsupervised representation learning model is proposed to consider the feature of time series subseries. The aim is to understand and encapsulate broader trends and shifts over larger intervals of time, yielding a coarse-grained representation of the time series. This form of representation is beneficial for applications where long-term trends and patterns are of interest, such as retrieval task. In this study, the covariance calculated by the Gaussian process is introduced to the self-attention mechanism, capturing relationship features of subseries. Experiments of retrieval verified the effectiveness of our proposed algorithm in coarse-grained representation of time series.

streaming version of segment-level representation learning: To showcase the versatility and robustness of our model, we extend its application to the domain of streaming time series data. This extension improves the model's practical significance and application value, enabling it can deal with the issue of time series in real-world data processing and analysis. In this extension, we redesign the algorithm, ensuring it is adept at handling continuous, real-time data streams, thereby broadening its applicability and efficacy beyond static time series data, and making it a versatile tool for diverse data environments and application contexts. Experiments in streaming time series data verified the effectiveness of expanded method.

cross-granularity representation learning: To Bridge representation learning models with different granularities, we introduce a novel cross-granularity representation model. This model is adept at integrating both fine-grained and coarse-grained representations, leveraging the strengths of each to provide a more

holistic understanding of time series data. This comprehensive integration ensures an enhanced accuracy in representation learning, making it a significant tool for various datasets of time series and analysis tasks.

Extensive experiments have been conducted in these models and the results have demonstrated the effectiveness of our proposed approach compared with baseline methods.

7.2 Future Work

Given the increasing complexity of data and the ongoing advances in machine learning methodologies, the proposed approach can also be adapted in cross-domain data sources. We will focus on deploying the application and extending the application in cross-domain data sources for future work. Besides, custom granularity levels and causal inference could also become the necessary area of multi-granularity research. Accordingly, we have the following future plan.

Real-world datasets: A pivotal emphasis will be on incorporating a diverse array of real-world datasets for more effective model testing. This strategy surpasses the limitations of using standard public datasets by embracing actual data, which is inherently more complex and reflective of real-world scenarios. Utilizing real-world data is not just a methodological enhancement but a fundamental shift towards ensuring that our models are not only theoretically robust but also practically applicable. Such an approach is essential in demonstrating the models' ability to tackle genuine, real-life problems, thereby making them significantly more relevant and impactful. By focusing on actual data, we aim to develop models that are not only academically credible but also capable of providing practical solutions in various industries and domains, ensuring their utility in solving tangible issues and contributing to real-world advancements.

Cross-domain data sources: Cross-domain adaptation speaks to the ability of models to transfer and apply knowledge learned from one domain to another—say, from healthcare patient data to meteorological time series—leveraging the inherent patterns that persist across different types of temporal data. This capability not only streamlines the process of model development by reducing the need for domain-specific data but also amplifies the utility of models across various applications. It involves developing algorithms that can abstract the core features of time series data at multiple scales, identifying those features that carry

over across domains and those that are domain-specific.

Custom granularity levels: Custom granularity levels in time series representation learning involve tailoring the resolution at which data is analyzed to better suit specific analytical tasks or to accommodate the unique characteristics of the dataset. By adjusting the granularity of the temporal data—ranging from microsecond-level details to broader, aggregated overviews—models can be more precisely fine-tuned to capture the most relevant patterns for a given problem. For instance, in financial markets, high-frequency trading algorithms may require granular millisecond-level data to capture the nuances of rapid market movements, whereas long-term investment strategies might rely on coarser, day-level or month-level data aggregates that highlight broader trends. Customizing granularity not only aids in focusing computational resources on the most informative aspects of the data but also helps in managing the noise-to-signal ratio, as finer granularities often come with increased data variability.

Causal inference: Causal inference in the context of multi-granularity time series learning is about discerning cause-and-effect relationships within temporal data. Traditional statistical methods for causal inference have often relied on controlled experiments or the identification of natural experiments within the data. However, in many real-world scenarios, especially those involving high-dimensional time series data, such methods can be impractical or insufficient. By integrating causal inference methodologies into multi-granularity frameworks, models can not only predict but also understand the underlying causal mechanisms that drive the observed temporal patterns.

Acknowledgements

I would like to send my special thanks to my supervisors, Professor Qiang Ma and Professor Takayuki Ito, for their constant encouragement and guidance. In addition, I would like to thank Professor Takayuki Kanda, and Professor Shinsuke Mori, for their advice and support to conduct this doctoral dissertation and the associated research tasks.

I would like to thank all the teachers at Yoshikawa and Ma Laboratory, Professor Masatoshi Yoshikawa, Associate Professor Yang Cao, and Associate Professor Kazunari Sugiyama, for raising questions and providing valuable comments at the seminar meetings.

I would also like to thank all the Lab members for sharing ideas. Thanks to Mr Yang Zhang, Mr Junjie Sun in our group, and Mr Shuyuang Zheng from the Yoshikawa group.

Special thanks to my wife Mrs. Pei Yu, for her understanding and support.

Chengyang YE, January 2024

References

[1] Dung Bui-Ngoc, Thanh Bui-Tien, Hieu Nguyen-Tran, Magd Abdel Wahab, and Guido De Roeck. Structural health monitoring using handcrafted features and convolution neural network. In *Proceedings of 1st International Conference on Structural Damage Modelling and Assessment: SDMA 2020, 4-5 August 2020, Ghent University, Belgium*, pages 103–112. Springer, 2021.

[2] Tian Han, Qinke Peng, Zhibo Zhu, Yiqing Shen, Huijun Huang, and Nahiyoon Nabeel Abid. A pattern representation of stock time series based on dtw. *Physica A: Statistical Mechanics and its Applications*, 550:124161, 2020.

[3] Marco Cuturi and Mathieu Blondel. Soft-dtw: a differentiable loss function for time-series. In *International conference on machine learning*, pages 894–903. PMLR, 2017.

[4] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. Ts2vec: Towards universal representation of time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8980–8987, 2022.

[5] Qianwen Meng, Hangwei Qian, Yong Liu, Lizhen Cui, Yonghui Xu, and Zhiqi Shen. Mhccl: masked hierarchical cluster-wise contrastive learning for multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9153–9161, 2023.

[6] Xiaochen Zheng, Xingyu Chen, Manuel Schürch, Amina Mollaysa, Ahmed

Allam, and Michael Krauthammer. Simts: Rethinking contrastive representation learning for time series forecasting. *arXiv preprint arXiv:2303.18205*, 2023.

[7] Jon Hills, Jason Lines, Edgaras Baranauskas, James Mapp, and Anthony Bagnall. Classification of time series by shapelet transformation. *Data mining and knowledge discovery*, 28:851–881, 2014.

[8] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. Efficient shapelet discovery for time series classification. *IEEE transactions on knowledge and data engineering*, 34(3):1149–1163, 2020.

[9] Josif Grabocka, Martin Wistuba, and Lars Schmidt-Thieme. Fast classification of univariate and multivariate time series through shapelet discovery. *Knowledge and information systems*, 49:429–454, 2016.

[10] Mingyue Cheng, Qi Liu, Zhiding Liu, Hao Zhang, Rujiao Zhang, and Enhong Chen. Timemae: Self-supervised representations of time series with decoupled masked autoencoders. *arXiv preprint arXiv:2303.00320*, 2023.

[11] Sana Tonekaboni, Danny Eytan, and Anna Goldenberg. Unsupervised representation learning for time series with temporal neighborhood coding. *arXiv preprint arXiv:2106.00750*, 2021.

[12] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1567–1577, 2022.

[13] Aren Jansen, Manoj Plakal, Ratheet Pandya, Daniel PW Ellis, Shawn Hershey, Jiayang Liu, R Channing Moore, and Rif A Saurous. Unsupervised learning of semantic audio representations. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 126–130. IEEE, 2018.

[14] Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. *Advances in neural information processing systems*, 32, 2019.

[15] Markus Bayer, Marc-André Kaufhold, and Christian Reuter. A survey on data augmentation for text classification. *ACM Computing Surveys*, 55(7):1–39, 2022.

[16] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[17] Samuel Harford, Fazle Karim, and Houshang Darabi. Generating adversarial samples on multivariate time series using variational autoencoders. *IEEE/CAA Journal of Automatica Sinica*, 8(9):1523–1538, 2021.

[18] Markus Thill, Wolfgang Konen, Hao Wang, and Thomas Bäck. Temporal convolutional autoencoder for unsupervised anomaly detection in time series. *Applied Soft Computing*, 112:107751, 2021.

[19] Qianli Ma, Jiawei Zheng, Sen Li, and Gary W Cottrell. Learning representations for time series clustering. *Advances in neural information processing systems*, 32, 2019.

[20] Yang Guo, Zhenyu Wu, and Yang Ji. A hybrid deep representation learning model for time series classification and prediction. In *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pages 226–231. IEEE, 2017.

[21] Qiuyue Zhang, Chao Qin, Yunfeng Zhang, Fangxun Bao, Caiming Zhang, and Peide Liu. Transformer-based attention network for stock movement prediction. *Expert Systems with Applications*, 202:117239, 2022.

[22] Lingxiao Meng, Wenjun Tan, Jiangang Ma, Ruofei Wang, Xiaoxia Yin, and Yanchun Zhang. Enhancing dynamic ecg heartbeat classification with lightweight transformer model. *Artificial Intelligence in medicine*, 124:102236, 2022.

[23] Gheorghe Grigoras, Florina Scarlatache, and Stefania Galbau. An efficient distribution transformer fleet modernization strategy for a rapid transition toward the active electric networks. In *2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 1–6. IEEE, 2023.

[24] George Zerveas, Srideepika Jayaraman, Dhaval Patel, Anuradha Bhamidipaty, and Carsten Eickhoff. A transformer-based framework for multivariate time series representation learning. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 2114–2124, 2021.

[25] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

[26] Weizheng Yan, Min Zhao, Zening Fu, Godfrey D Pearlson, Jing Sui, and Vince D Calhoun. Mapping relationships among schizophrenia, bipolar and schizoaffective disorders: a deep classification and clustering framework using fmri time series. *Schizophrenia Research*, 245:141–150, 2022.

[27] Taha Belkhouja, Yan Yan, and Janardhan Rao Doppa. Dynamic time warping based adversarial framework for time-series domain. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[28] Weikuan Jia, Meili Sun, Jian Lian, and Sujuan Hou. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, 8(3):2663–2693, 2022.

[29] Paul Boniol and Themis Palpanas. Series2graph: Graph-based subsequence anomaly detection for time series. *arXiv preprint arXiv:2207.12208*, 2022.

[30] Mahbuba Begum, Jannatul Ferdush, and Mohammad Shorif Uddin. A hybrid robust watermarking system based on discrete cosine transform, discrete wavelet transform, and singular value decomposition. *Journal of King Saud University-Computer and Information Sciences*, 34(8):5856–5867, 2022.

[31] Zhijie Zhang, Wenzhong Li, Wangxiang Ding, Linming Zhang, Qingning Lu, Peng Hu, Tong Gui, and Sanglu Lu. Stad-gan: unsupervised anomaly detection on multivariate time series with self-training generative adversarial networks. *ACM Transactions on Knowledge Discovery from Data*, 17(5):1–18, 2023.

[32] Ozge Cagcag Yolcu, Erol Egrioglu, Eren Bas, and Ufuk Yolcu. Multivariate intuitionistic fuzzy inference system for stock market prediction: The cases of istanbul and taiwan. *Applied Soft Computing*, 116:108363, 2022.

[33] Fred Mubang and Lawrence O Hall. Vam: an end-to-end simulator for time series regression and temporal link prediction in social media networks. *IEEE transactions on computational social systems*, 2022.

[34] Mikael van Deurs, Mollie E Brooks, Martin Lindegren, Ole Henriksen, and Anna Rindorf. Biomass limit reference points are sensitive to estimation method, time-series length and stock development. *Fish and Fisheries*, 22(1):18–30, 2021.

[35] Ari Yair Barrera-Animas, Lukumon O Oyedele, Muhammad Bilal, Taofeek Dolapo Akinosho, Juan Manuel Davila Delgado, and Lukman Adewale Akanbi. Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting. *Machine Learning with Applications*, 7:100204, 2022.

[36] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

[37] Yuan Yuan, Lei Lin, Qingshan Liu, Renlong Hang, and Zeng-Guang Zhou. Sits-former: A pre-trained spatio-spectral-temporal representation model for sentinel-2 time series classification. *International Journal of Applied Earth Observation and Geoinformation*, 106:102651, 2022.

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[40] Hongji Xu, Juan Li, Hui Yuan, Qiang Liu, Shidi Fan, Tiankuo Li, and Xiaojie Sun. Human activity recognition based on gramian angular field and deep convolutional neural network. *IEEE Access*, 8:199393–199405, 2020.

[41] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 815–825, 2022.

[42] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[43] Hyukjun Gweon and Hao Yu. A nearest neighbor-based active learning method and its application to time series classification. *Pattern Recognition Letters*, 146:230–236, 2021.

[44] Diyar Qader Zeebaree, Habibollah Haron, Adnan Mohsin Abdulazeez, and Dilovan Asaad Zebari. Trainable model based on new uniform lbp feature to identify the risk of the breast cancer. In *2019 international conference on advanced science and engineering (ICOASE)*, pages 106–111. IEEE, 2019.

[45] Shiqi Wang, Mingfang Jiang, Jiaohua Qin, Hengfu Yang, and Zhichen Gao. A secure rotation invariant lbp feature computation in cloud environment. *CMC-Comput. Mater. Contin*, 68:2979–2993, 2021.

[46] Guoying Zhao and Matti Pietikäinen. Dynamic texture recognition using volume local binary patterns. In *International Worksop on Dynamical Vision*, pages 165–177. Springer, 2005.

[47] Navin Chatlani and John J Soraghan. Local binary patterns for 1-d signal processing. In *2010 18th European signal processing conference*, pages 95–99. IEEE, 2010.

[48] Xing Hu and Guoqiang Li. Temporal tensor local binary pattern: A novel local tensor time series descriptor. *IEEE Transactions on Industrial Informatics*, 16(10):6393–6402, 2019.

[49] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018.

[50] Sanghun Lee and Chulhee Lee. Revisiting spatial dropout for regularizing convolutional neural networks. *Multimedia Tools and Applications*, 79(45-46):34195–34207, 2020.

[51] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[52] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Local binary convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 19–28, 2017.

[53] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *arXiv preprint arXiv:2304.13029*, 2023.

[54] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[55] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, 1(4):1–4, 2015.

[56] Chang Wei Tan, Christoph Bergmeir, François Petitjean, and Geoffrey I Webb. Time series extrinsic regression: Predicting numeric values from time series data. *Data Mining and Knowledge Discovery*, 35:1032–1060, 2021.

[57] Angus Dempster, François Petitjean, and Geoffrey I Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

[58] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[59] Hongyan Wu, Yunpeng Cai, Yongsheng Wu, Ren Zhong, Qi Li, Jing Zheng, Denan Lin, and Ye Li. Time series analysis of weekly influenza-like illness rate using a one-year period of factors in random forest regression. *Bioscience trends*, 11(3):292–296, 2017.

[60] Kuo-Kun Tseng, Jiaqian Li, Yih-Jing Tang, Ching Wen Yang, and Fang-Ying Lin. Healthcare knowledge of relationship between time series electrocardiogram and cigarette smoking using clinical records. *BMC Medical Informatics and Decision Making*, 20:1–11, 2020.

[61] Armin Lawi, Hendra Mesra, and Supri Amir. Implementation of long short-term memory and gated recurrent units on grouped time-series data to predict stock prices accurately. *Journal of Big Data*, 9(1):1–19, 2022.

[62] Jiale Cao, Yanwei Pang, Shengjie Zhao, and Xuelong Li. High-level semantic networks for multi-scale object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3372–3386, 2019.

[63] Yucheng Lu, Qiang Ji, Liang Wang, Tianshu Wu, Hongbo Deng, Jian Xu, and Bo Zheng. Stardom: semantic aware deep hierarchical forecasting model for search traffic prediction. In *Proceedings of the 31st ACM International Conference On Information & Knowledge Management*, pages 3352–3360, 2022.

[64] Guozhong Li, Byron Choi, Jianliang Xu, Sourav S Bhowmick, Kwok-Pan Chun, and Grace Lai-Hung Wong. Shapenet: A shapelet-neural network approach for multivariate time series classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 8375–8383, 2021.

[65] Matthew Middlehurst, William Vickers, and Anthony Bagnall. Scalable dictionary classifiers for time series classification. In *Intelligent Data Engineering and Automated Learning–IDEAL 2019: 20th International Con-*

ference, Manchester, UK, November 14–16, 2019, Proceedings, Part I 20, pages 11–19. Springer, 2019.

[66] Wei Liu, Shengcai Liao, Weiqiang Ren, Weidong Hu, and Yinan Yu. High-level semantic feature detection: A new perspective for pedestrian detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5187–5196, 2019.

[67] David Hallac, Peter Nystrup, and Stephen Boyd. Greedy gaussian segmentation of multivariate time series. *Advances in Data Analysis and Classification*, 13(3):727–751, 2019.

[68] Yie-Tarng Chen, Wen-Hsien Fang, Shi-Ting Dai, and Choa-Chuan Lu. Skeleton moving pose-based human fall detection with sparse coding and temporal pyramid pooling. In *2021 7th International Conference on Applied System Innovation (ICASI)*, pages 91–96. IEEE, 2021.

[69] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10076–10085, 2020.

[70] Amine Hadji and Botond Szabó. Can we trust bayesian uncertainty quantification from gaussian process priors with squared exponential covariance kernel? *SIAM/ASA Journal on Uncertainty Quantification*, 9(1):185–230, 2021.

[71] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4834–4843, 2018.

[72] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905, 2021.

[73] Cuong Ha, Van-Dang Tran, Linh Ngo Van, and Khoat Than. Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout. *International Journal of Approximate Reasoning*, 112:85–104, 2019.

[74] Yanping Chen, Bing Hu, Eamonn Keogh, and Gustavo EAPA Batista. Dtw-d: time series semi-supervised learning from a single example. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 383–391, 2013.

[75] Ting Wang, Sheng-Uei Guan, Ka Lok Man, TO Ting, et al. Eeg eye state identification using incremental attribute learning with time-series classification. *Mathematical Problems in Engineering*, 2014, 2014.

[76] Erick Stattner and Martine Collard. Modèles et l'analyse des réseaux: approches mathématiques et informatiques (marami). In *Conférence sur les modèles et l'analyse des réseaux: Approches mathématiques et informatiques (MARAMI)*, volume 4, pages pp–40, 2013.

[77] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1183–1192, 2016.

[78] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi, Yanhua Yang, Cheng Deng, and Heng Huang. Unsupervised deep generative adversarial hashing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3664–3673, 2018.

[79] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.

[80] Monica Arul and Ahsan Kareem. Applications of shapelet transform to time series classification of earthquake, wind and wave data. *Engineering Structures*, 228:111564, 2021.

[81] Shima Imani and Eamonn Keogh. Matrix profile xix: time series semantic motifs: a new primitive for finding higher-level structure in time series. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 329–338. IEEE, 2019.

[82] Chengyang Ye and Qiang Ma. Gp-hls: Gaussian process-based unsupervised high-level semantics representation learning of multivariate time se-

ries. In *International Conference on Database Systems for Advanced Applications*, pages 221–236. Springer, 2023.

[83] Eric P Lehman, Rahul G Krishnan, Xiaopeng Zhao, Roger G Mark, and H Lehman Li-Wei. Representation learning approaches to detect false arrhythmia alarms from ecg dynamics. In *Machine learning for healthcare conference*, pages 571–586. PMLR, 2018.

[84] Zheng Yang, Binbin Xu, Wei Luo, and Fei Chen. Autoencoder-based representation learning and its application in intelligent fault diagnosis: A review. *Measurement*, 189:110460, 2022.

[85] Youqiang Sun, Jiuyong Li, Jixue Liu, Bingyu Sun, and Christopher Chow. An improvement of symbolic aggregate approximation distance measure for time series. *Neurocomputing*, 138:189–198, 2014.

[86] Antigoni Mezari and Ilias Maglogiannis. Gesture recognition using symbolic aggregate approximation and dynamic time warping on motion data. In *Proceedings of the 11th EAI International Conference on Pervasive Computing Technologies for Healthcare*, pages 342–347, 2017.

[87] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020.

[88] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

[89] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.

[90] Aoqian Zhang, Shaoxu Song, Jianmin Wang, and Philip S Yu. Time series data cleaning: From anomaly detection to anomaly repairing. *Proceedings of the VLDB Endowment*, 10(10):1046–1057, 2017.

[91] Matthew D Zeiler and Rob Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013.

[92] Albert Bifet, Silviu Maniu, Jianfeng Qian, Guangjian Tian, Cheng He, and Wei Fan. Streamdm: Advanced data mining in spark streaming. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1608–1611. IEEE, 2015.

[93] Izaskun Oregi, Aritz Pérez, Javier Del Ser, and Jose A Lozano. An active adaptation strategy for streaming time series classification based on elastic similarity measures. *Neural Computing and Applications*, 34(16):13237–13252, 2022.

[94] Izaskun Oregi, Aritz Pérez, Javier Del Ser, and José A Lozano. On-line dynamic time warping for streaming time series. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II 10*, pages 591–605. Springer, 2017.

[95] Yiming Ding, Wei Luo, Yufei Zhao, Zhen Li, Peng Zhan, and Xueqing Li. A novel similarity search approach for streaming time series. In *Journal of Physics: Conference Series*, volume 1302, page 022084. IOP Publishing, 2019.

[96] Xiang Lian, Lei Chen, Jeffrey Xu Yu, Guoren Wang, and Ge Yu. Similarity match over high speed time-series streams. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 1086–1095. IEEE, 2006.

[97] Wei Luo, Yongqi Li, Fubin Yao, Shaokun Wang, Zhen Li, Peng Zhan, and Xueqing Li. Multi-resolution representation for streaming time series retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 35(06):2150019, 2021.

[98] Yuan-Yuan Fan, Chu Chu, Yun-Ting Zhang, Kun Zhao, Li-Xia Liang, Jing-Wen Huang, Jia-Xin Zhou, Li-Hao Guo, Lu-Yin Wu, Li-Zi Lin, et al. Environmental pollutant pre-and polyfluoroalkyl substances are associated with electrocardiogram parameters disorder in adults. *Journal of Hazardous Materials*, page 131832, 2023.

[99] Shahzad Zaheer, Nadeem Anjum, Saddam Hussain, Abeer D Algarni, Jawaid Iqbal, Sami Bourouis, and Syed Sajid Ullah. A multi parameter

forecasting for stock time series data using lstm and deep learning model. *Mathematics*, 11(3):590, 2023.

[100] Arshi Parvaiz, Muhammad Anwaar Khalid, Rukhsana Zafar, Huma Ameer, Muhammad Ali, and Muhammad Moazam Fraz. Vision transformers in medical computer vision—a contemplative retrospection. *Engineering Applications of Artificial Intelligence*, 122:106126, 2023.

[101] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419*, 2023.

[102] Jason Lines, Sarah Taylor, and Anthony Bagnall. Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(5):1–35, 2018.

[103] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I Webb. Ts-chief: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery*, 34(3):742–775, 2020.

[104] Marco S Reis. Multiscale and multi-granularity process analytics: A review. *Processes*, 7(2):61, 2019.

[105] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. Time2vec: Learning a vector representation of time. *arXiv preprint arXiv:1907.05321*, 2019.

[106] Kenneth Ward Church. Word2vec. *Natural Language Engineering*, 23(1):155–162, 2017.

[107] Yufeng Yu, Yuelong Zhu, Dingsheng Wan, Huan Liu, and Qun Zhao. A novel symbolic aggregate approximation for time series. In *Proceedings of the 13th International Conference on Ubiquitous Information Management and Communication (IMCOM) 2019 13*, pages 805–822. Springer, 2019.

[108] Jin Shieh and Eamonn Keogh. i sax: disk-aware mining and indexing of massive time series datasets. *Data Mining and Knowledge Discovery*, 19:24–57, 2009.

[109] Manuel Anacleto, Susana Vinga, and Alexandra M Carvalho. Msax: Multivariate symbolic aggregate approximation for time series classification. In *Computational Intelligence Methods for Bioinformatics and Biostatistics: 16th International Meeting, CIBB 2019, Bergamo, Italy, September 4–6, 2019, Revised Selected Papers 16*, pages 90–97. Springer, 2020.

[110] Min Hou, Chang Xu, Yang Liu, Weiqing Liu, Jiang Bian, Le Wu, Zhi Li, Enhong Chen, and Tie-Yan Liu. Stock trend prediction with multi-granularity data: A contrastive learning approach with adaptive fusion. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 700–709, 2021.

[111] Xin Yang, Metoh Adler Loua, Meijun Wu, Li Huang, and Qiang Gao. Multi-granularity stock prediction with sequential three-way decisions. *Information Sciences*, 621:524–544, 2023.

[112] Thapana Boonchoo. Ms-sralat: Multi-granularity substructure-aware representation learning algorithm for time-series. In *2022 19th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 1–6. IEEE, 2022.

[113] Feng Nie, Yunbo Cao, Jinpeng Wang, Chin-Yew Lin, and Rong Pan. Mention and entity description co-attention for entity disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[114] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

[115] Joost CF De Winter, Samuel D Gosling, and Jeff Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273, 2016.

[116] Mete Kemertas, Leila Pishdad, Konstantinos G Derpanis, and Afsaneh Fazly. Rankmi: A mutual information maximizing ranking loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14362–14371, 2020.

[117] Tyler Sypherd, Mario Diaz, Lalitha Sankar, and Peter Kairouz. A tunable loss function for binary classification. In *2019 IEEE international symposium on information theory (ISIT)*, pages 2479–2483. IEEE, 2019.

[118] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020.

[119] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. Lstm fully convolutional networks for time series classification. *IEEE access*, 6:1662–1669, 2017.

Selected List of Publications

- **Journals**

  [1] Chengyang Ye and Qiang Ma. LBP4MTS: Local Binary Pattern-Based Unsupervised Representation Learning of Multivariate Time Series. *IEEE Access*. 2023, 11: 118595-118605.

  [2] Chengyang Ye and Qiang Ma. Semantic Relationship-Based Unsupervised Representation Learning of Multivariate Time Series. *IEICE Transactions on Information and System*.

- **International Conferences**

  [3] Chengyang Ye and Qiang Ma. TS2V: A Transformer-Based Siamese Network for Representation Learning of Univariate Time-Series Data. In *2022 IEEE 25th International Conference on Computer Supported Cooperative Work in Design (CSCWD 2022)*, pages 1245–1250, Hangzhou, China, May 2022.

  [4] Chengyang Ye and Qiang Ma. GP-HLS: Gaussian Process-Based Unsupervised High-Level Semantics Representation Learning of Multivariate Time Series In *The 28th International Conference on Database Systems for Advanced Applications (DASFAA 2023)*, pages 221–236, Tianjin, China, April 2023.

  [5] Chengyang Ye and Qiang Ma. Unsupervised Representation Learning with Semantic of Streaming Time Series In *24th International Conference on Web Information Systems Engineering (WISE 2023)*, Melbourne, Australia, October 2023.

[6] Chengyang Ye and Qiang Ma. Multi-Granularity Framework for Unsupervised Representation Learning of Time Series In arXiv.

- **Domestic Conferences**

[7] Chengyang Ye and Qiang Ma. Representation Learning of Time Series Data with High-Level Semantic Features. In *The 20th Forum on Data Engineering and information Management (DEIM)*, H23-4, 2022.