

# Chapter 1

## Leveraging Motifs for Time Series Forecasting with Matrix Profile

### 1.1 Introduction

Humans have made predictions since ancient times. In ancient societies, making an accurate prediction was crucial to ensure the success of subsistence activities such as hunting, planting, and harvesting. There was a need to predict weather dynamics, such as rainfall and temperature. For example, plant during the season with sufficient rainfall and appropriate temperatures. Our ancestors used divination tools such as turtle shells, wooden blocks (Moon blocks), or bones to make predictions. But the accuracy of those predictions was not guaranteed. More advanced predictions may involve astronomy. They focused on predicting celestial events, such as eclipses, for religious and agricultural purposes. Even in modern societies, prediction is still essential. Predicting traffic-jam patterns only a few hours ahead can save time by enabling the selection of an alternative route. A wealth could be created by forecasting stock market trends. Predicting the future, also known as time series forecasting, is crucial for us.

With advances in hardware technologies, we collect enormous amounts of data from diverse sources, such as smart sensors [14] and social media platforms, continuously, in the form of time series data. A time series is an ordered sequence of measures, represented in real-valued numbers, at discrete equal-interval times-

tamps [4]. These vast data collections have created the era of Big Data, which provides a wealth of datasets for developing and deploying reliable, robust, data-driven forecasting techniques to discover patterns and extract valuable information from the past data [1]. Applications can be found in the financial sector, such as predicting business cycles and stock market movements [9, 8, 15, 5] and the medical field, such as the status of critical patients according to their vital signs [18, 6] and the propagation of diseases such as influenza [5, 19] and COVID-19 [18, 12].

Many methods have been developed for time series forecasting to handle these tasks. Traditional methods include rolling averages, vector auto-regression (VAR) [8, 3], and auto-regressive integrated moving averages (ARIMA) [7, 2, 3]. Because of their statistical properties, they have long been the standard. The shortcomings of ARIMA and its variants include their high computational cost [8]. In contrast, VAR is arguably the most widely used method, particularly in multivariate time series analysis, owing to its simplicity. However, most of these traditional approaches have certain limitations. They perform well if the data meet their specific statistical assumptions, such as stationarity [10], which means the mean and variance of the time series remain consistent throughout time. For example, ARIMA performs well only if the underlying stochastic process of the time series follows ARIMA. It motivates the community to develop machine learning methods, particularly deep learning methods, to address this problem. Many deep learning models have been proposed, including RNN-based models, CNN-based models, GNN-based models, Transformer-based models, and compound models that incorporate different base models mentioned before [5]. The compound models are promising trends. For example, RNNs are well-suited to capturing long-term dependencies, whereas CNNs are well-suited to capturing short-term dependencies. A good way to improve performance is to compound the base models. LSTnet [8] integrates CNN, RNN, and autoregressive [21] techniques to extract both short-term local dependency patterns and long-term patterns. Using the occupancy rate of a freeway as an example [8] to explain these two patterns, the former one refers to the morning peaks against evening peaks, while the latter one refers to the workday and weekend patterns. Clearly, a good forecaster needs to capture both kinds of recurring patterns. Despite the superior performance deep learning methods have achieved,

they tend to be overly complex and incur high computational costs compared to traditional techniques.

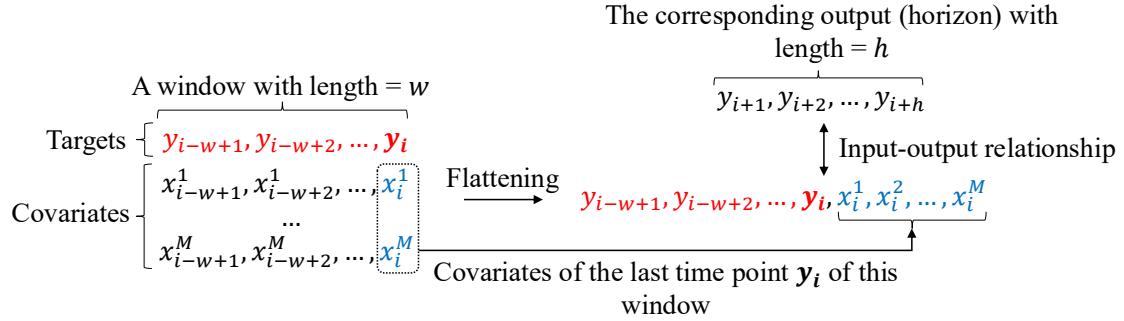


Figure 1.1: Flattening.

A recent study [13] shows that, in time-series classification, nearest-neighbor classifiers (1-NN) and  $k$ -NN with appropriate distance measures, such as Dynamic Time Warping (DTW), despite their simplicity, perform well and are therefore commonly used as benchmarks. In detail, when a new instance is to be classified,  $k$ -NN finds its  $k$  nearest neighbors in the training set and returns their majority labels. It is considered a lazy learner because the training steps involve only memorizing all the instances verbatim; no higher-level concepts have been learned. In addition, in time series forecasting, a recent study demonstrates that a well-known machine learning baseline, Gradient Boosting Regression Tree (GBRT), such as XGBoost, equipped with an appropriate data engineering of the input and output, can achieve competitive or even superior performance than the deep learning method. In detail, they transform the time series forecasting task into a window-based regression problem, as shown in Figure 1.1. For each training window, the target variable's lag values are concatenated with external features and then flattened to form a single input instance for a multi-output regression GBRT model. It provides a more efficient yet accurate method for time series forecasting.

In addition, the non-parametric, instance-based method, known as  $k$ -nearest neighbors, has shown to be a promising method for classification and regression [11]. The forecaster uses lagged values as explanatory variables. These lagged values form a query  $Q$ . We identify the previous similar patterns to  $Q$  and use

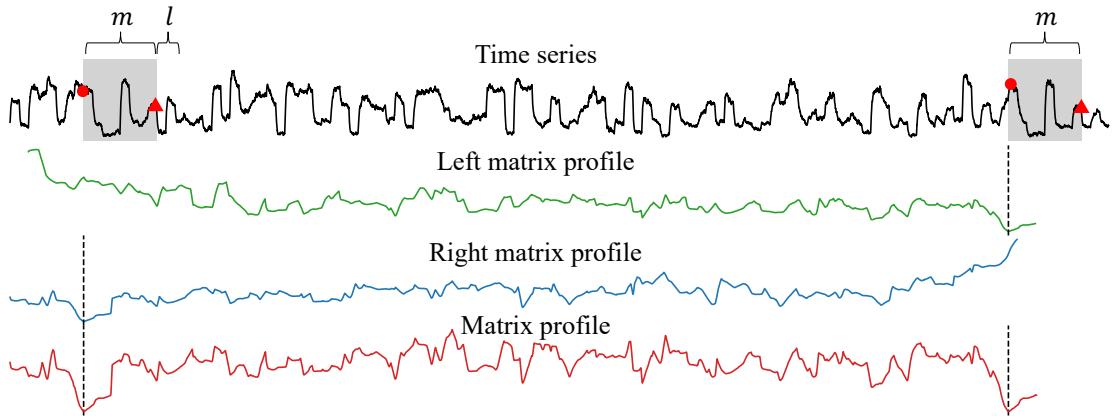


Figure 1.2: The left (right) matrix profile and the matrix profile of a time series. The matrix profile shows the distances between each  $m$ -subsequence and its nearest neighbor. The left (right) matrix profile shows the same information but is limited to its left (right) nearest neighbor. For a particular  $m$ -subsequence shown by the right gray box, its nearest neighbor is the left gray box, as shown in the left matrix profile and the matrix profile. The first (last) point in each box is denoted by a red circle (triangle).  $l$  denotes the length of the subsequent subsequence of the nearest neighbor. Intuitively, this subsequence should be similar to the subsequent (i.e., future) subsequence of the right gray box. To note, the left (right) matrix profile starts (ends) at a later (earlier) index because the corresponding nearest neighbors with length  $m$  do not exist.

their subsequent patterns to predict the future behavior of  $Q$ . The intuition is that history repeats itself. The historical subsequences that are similar to the current state (represented by  $Q$ ) can provide us with a hint about the future. These subsequences are similar, and so are their subsequent subsequences (i.e., their future). This notion has been depicted in Figure 1.2. Observe that the immediate subsequence after the right gray box is similar to that of the left gray box. The left gray box is the nearest neighbor of the right gray in the “past”.

Based on these findings, this study proposes a hybrid method that uses each window’s nearest neighbors’ information in the past to create new covariates. The information includes the similarity between the window and its nearest neighbors, as well as the subsequent points of them. The similarities can be regarded as a measure of confidence in using the information from the corresponding nearest neighbor. The fundamental difference between this study and the previous

approaches [16, 11, 17] is that they directly use the subsequent points for the prediction, in contrast, we utilize the nearest neighbor’s information of each subsequence as covariates. This difference can be explained by Figure 1.2. In the previous approaches, they simply use the information of the nearest neighbors of the last look-back window (i.e., the right gray box in the figure). In contrast, we use the information of the nearest neighbors of *ALL* of the look-back window. It may sound expensive for this computation. We leverage a data primitive, namely Matrix Profile [20, 22]. It is a fast tool that computes meta time series that annotate the original time series, showing the nearest-neighbor information for each subsequence, including the similarity with its nearest neighbor and the location of the nearest neighbor. Note that, given the location information, we can also retrieve their subsequent points, as shown in Figure 1.2.



# Bibliography

- [1] Omar Y. Al-Jarrah, Paul D. Yoo, Sami Muhaidat, George K. Karagiannidis, and Kamal Taha. Efficient machine learning for big data: A review. *Big Data Research*, 2(3):87–93, September 2015.
- [2] G. E. P. Box and David A. Pierce. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332):1509–1526, 1970.
- [3] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, May 2015.
- [4] Chris Chatfield. *The Analysis of Time Series: An Introduction, Sixth Edition*. Chapman and Hall/CRC, New York, 6 edition, July 2003.
- [5] Zonglei Chen, Minbo Ma, Tianrui Li, Hongjun Wang, and Chongshou Li. Long sequence time-series forecasting with deep learning: A survey. *Information Fusion*, 97:101819, September 2023.
- [6] Denise Bandeira da Silva, Diogo Schmidt, Cristiano André da Costa, Rodrigo da Rosa Righi, and Björn Eskofier. Deepsigns: A predictive model based on deep learning for the early detection of patient health deterioration. *Expert Systems with Applications*, 165:113905, March 2021.
- [7] Shereen Elsayed, Daniela Thyssens, Ahmed Rashed, Hadi Samer Jomaa, and Lars Schmidt-Thieme. Do we really need deep learning models for time series forecasting?, October 2021.

- [8] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, SIGIR '18, pages 95–104, New York, NY, USA, June 2018. Association for Computing Machinery.
- [9] Wenxiang Li and K. L. Eddie Law. Deep learning models for time series forecasting: A review. *IEEE Access*, 12:92306–92327, 2024.
- [10] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 379(2194):20200209, February 2021.
- [11] Francisco Martínez, María Pilar Frías, María Dolores Pérez, and Antonio Jesús Rivera. A methodology for applying k-nearest neighbor to time series forecasting. *Artificial Intelligence Review*, 52(3):2019–2037, October 2019.
- [12] F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, D. Gutiérrez-Avilés, L. Melgar-García, R. Pérez-Chacón, C. Rubio-Escudero, J. C. Riquelme, and A. Troncoso. Coronavirus optimization algorithm: A bioinspired metaheuristic based on the covid-19 propagation model. *Big Data*, 8(4):308–322, August 2020.
- [13] Matthew Middlehurst, Patrick Schäfer, and Anthony Bagnall. Bake off redux: A review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1958–2031, July 2024.
- [14] Andreas P. Plageras, Kostas E. Psannis, Christos Stergiou, Haoxiang Wang, and B. B. Gupta. Efficient iot-based sensor big data collection–processing and analysis in smart buildings. *Future Generation Computer Systems*, 82:349–357, May 2018.
- [15] Nicholas I. Sapankevych and Ravi Sankar. Time series prediction using support vector machines: A survey. *IEEE Computational Intelligence Magazine*, 4(2):24–38, May 2009.

- [16] Samya Tajmouati, Bouazza E. L. Wahbi, Adel Bedoui, Abdallah Abarda, and Mohamed Dakkon. Applying k-nearest neighbors to time series forecasting: Two new approaches. *Journal of Forecasting*, 43(5):1559–1574, 2024.
- [17] Dung Tran Thi, Nha Tran Phong, and Ngoc Dung Bui. Forecast timeseries based on matrix profile. *Journal of Informatics and Innovative Technologies (JIIT)*, 2021.
- [18] José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21, February 2021.
- [19] Neo Wu, Bradley Green, Xue Ben, and Shawn O’Banion. Deep transformer models for time series forecasting: The influenza prevalence case, January 2020.
- [20] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. Matrix profile i: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 1317–1322, Barcelona, Spain, December 2016. IEEE.
- [21] G. Udny Yule. On a method of investigating periodicities in disturbed series, with special reference to wolfer’s sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226:267–298, 1927.
- [22] Yan Zhu, Makoto Imamura, Daniel Nikovski, and Eamonn Keogh. Matrix profile vii: Time series chains: A new primitive for time series data mining. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 695–704, November 2017.