Advancing Time Series Similarity in Financial Predictions: A Proposal

Gil Biton
Ben-Gurion University
Beer-Sheva, Israel
gilbito@post.bgu.ac.il

Ruben Sasson
Ben-Gurion University
Tel-Aviv, Israel
rubenyaa@post.bgu.ac.il

GitHub Repository: https://github.com/ruben-bgu/time-series-analysis-project.git

1 Abstract

Predicting stock prices is one of the most fundamental challenges in financial research, requiring advanced methodologies to account for the complexities of market dynamics. Traditional statistical models have been widely employed in stock price prediction, but the increasing availability of large financial datasets has led to the adoption of machine learning (ML) techniques for improved forecasting accuracy. As financial markets evolve, so does the need for more sophisticated tools that capture intricate patterns in large-scale data. This paper reviews several approaches to enhancing stock price prediction, focusing on machine learning models such as XGBoost, similarity-based methods, and clustering techniques. We also introduce an extension to existing methodologies, addressing common limitations in ML models, particularly their struggle with rare events and sequential pattern forgetting. Our proposed approach integrates sequential pattern recognition with machine learning predictions to enhance the robustness and accuracy of stock price forecasting, ultimately providing a more reliable and dynamic method for financial prediction.

2 Introduction

Stock price prediction is a critical task in financial research and trading, where accurate forecasts can yield significant profits. The ability to predict future stock prices is not only valuable to individual investors but also to institutional traders, risk managers, and financial analysts who rely on these predictions to make informed decisions. The complexity of financial markets, characterized by volatile and nonlinear price movements, requires advanced computational methods that effectively capture dynamic market behavior. Traditional statistical techniques, such as linear regression and autoregressive models, have been widely used but face limitations when handling high-dimensional, nonlinear datasets. Moreover, these methods often overlook sudden market shifts and do not adapt easily to evolving conditions.

In recent years, machine learning (ML) approaches have gained popularity for stock price forecasting due to their ability to model complex patterns within large datasets. Gradient boosting algorithms, for example, provide robust performance by combining multiple weak learners in a boosting framework. Nonetheless, the quality of ML predictions depends heavily on diverse and representative data. Relying on a single stock's history can limit predictive power, especially for short-horizon forecasts. Integrating information from stocks that exhibit similar market behavior has shown potential to broaden context and improve overall accuracy.

This paper surveys critical research on time series similarity and its impact on stock price prediction, including the usage of similarity-based methods, clustering approaches, and symbolic representations. Additionally, we propose a novel framework that integrates detected sequential patterns from similar stocks into ML-based predictions. By doing so, we address key challenges, such as infrequent but impactful market events and the propensity of certain models to forget relevant sequential patterns. Our method aims to enhance both the robustness and precision of stock price forecasts.

3 Background

3.1 Machine Learning in Stock Price Prediction

Stock price prediction is a complex problem due to the non-stationary and volatile nature of financial markets. Traditional statistical methods, such as autoregressive models and linear regression, have been widely used but often struggle to capture nonlinear dependencies and abrupt market shifts.

Machine learning (ML) techniques offer a promising alternative, allowing models to learn intricate patterns from historical data. Among these, gradient boosting algorithms such as *XGBoost* have gained significant attention. Zhang et al. [11] demonstrated that XGBoost outperforms traditional statistical models by efficiently handling high-dimensional datasets and identifying non-obvious interactions among predictors.

Despite their advantages, ML models rely heavily on the quality and diversity of training data. A common limitation is their dependence on a single stock's historical prices, which restricts their ability to generalize across different market conditions. One way to mitigate this issue is by incorporating additional sources of information, such as price movements of similar stocks. The challenge then becomes how to define and quantify stock similarity in a way that enhances predictive performance.

3.2 Time-Series Similarity in Financial Forecasting

A key assumption in financial modeling is that stocks exhibiting similar past behaviors may share underlying economic drivers. By leveraging data from such stocks, predictive models can gain a broader perspective on market trends and improve their ability to generalize. However, determining similarity between time series is non-trivial due to differences in volatility, scale, and time alignment.

One of the simplest similarity measures is *Euclidean distance*, which computes the pointwise difference between two time series. While effective in some cases, this method is sensitive to temporal misalignment, meaning that two stocks with similar trends but different starting points may be incorrectly classified as dissimilar.

To address this limitation, *Dynamic Time Warping (DTW)* has been proposed as a more flexible similarity measure [10]. DTW dynamically aligns two time series by allowing non-linear transformations along the time axis, enabling the comparison of sequences that evolve at different speeds. Unlike Euclidean distance, which requires one-to-one correspondence, DTW finds the optimal alignment path that minimizes the total distance between two sequences.

While DTW improves similarity detection, it comes at a computational cost. The standard DTW algorithm has a complexity of ON^2, making it impractical for large-scale financial applications. Approximate variants such as *FastDTW* have been proposed to improve efficiency, but they often trade off accuracy for speed.

Beyond DTW, another widely used method for similarity detection is *co-integration analysis*, which focuses on long-term dependencies rather than short-term fluctuations. Unlike distance-based methods, co-integration examines whether two stocks maintain a stable linear relationship over time, making it a more robust approach for identifying structurally related assets [3].

3.3 Co-integration in Financial Markets

While DTW and Euclidean distance focus on short-term price similarity, co-integration is a statistical technique designed to identify *long-term relationships* between non-stationary time series. Co-integration is particularly relevant in financial markets, where asset prices often exhibit common stochastic trends due to macroeconomic factors.

Two stocks are considered co-integrated if their price ratio remains stable over time, even if the individual price movements are non-stationary. This suggests a fundamental connection between the assets, which may arise due to economic relationships, sectoral influences, or arbitrage opportunities. Caiado and Crato [5] demonstrated that co-integration-based clustering can outperform traditional distance metrics when analyzing stock volatilities.

Co-integration is widely used in financial modeling for portfolio construction and statistical arbitrage. Alexander and Dimitriu [3] applied co-integration analysis to study market efficiency, while Soon and Lee [8] utilized it for improving stock similarity search. By selecting stocks that exhibit stable long-term dependencies, co-integration offers a principled way to enhance ML models with meaningful external data.

3.4 Symbolic Representations for Time Series

While similarity and clustering techniques provide ways to compare stocks, another challenge in stock price prediction is the high dimensionality of financial time series. *Symbolic Aggregate Approximation (SAX)* addresses this issue by converting continuous time series into discrete symbolic sequences [6].

SAX works by partitioning the time series into equal-sized segments and assigning each segment to a discrete symbol based on predefined thresholds. This transformation reduces the complexity of financial data while preserving its essential trends, making it suitable for pattern recognition and anomaly detection.

Lin et al. [6] introduced SAX as a method for compressing time series data while retaining its fundamental characteristics. However, one limitation of SAX is its inability to capture subtle local variations in stock price movements. To address this, hybrid approaches have been proposed that combine SAX with *technical indicators* such as *Moving Average Convergence Divergence (MACD)* [9] or machine learning models optimized for sequence classification [4].

3.5 Challenges and Enhancements to Existing Models

Despite advances in ML and time-series analysis, many stock prediction models still face challenges in handling *rare events* and *sequential dependencies*. A common issue in ML-based financial models is the tendency to lose track of historical patterns, especially when trained on rolling windows of data.

Aghabozorgi et al. [1] emphasized the importance of time-dependent structures in financial data, advocating for methods that retain sequential information over extended periods. Sidi [7] proposed incorporating co-integrated stocks into ML-based forecasts, demonstrating improvements in accuracy. However, this approach does not fully address the issue of sequence forgetting.

To overcome these challenges, we propose a hybrid methodology that integrates *sequential pattern recognition* with ML-based forecasts. By combining SAX for feature extraction, DTW for short-term sequence alignment, and co-integration for selecting relevant stocks, our approach aims to build a more robust stock prediction model that captures both local fluctuations and long-term dependencies.

4 Methods and Experimental Setup

In this section, we outline the methodology used to predict stock price movements based on past trends and relationships between stocks. Our workflow consists of three main phases:

(1) Establishing a **baseline XGBoost model**, trained exclusively on AAPL stock.
(2) **Enhancing the model** by incorporating stocks with similar long-term behavior, identified via co-integration analysis.
(3) **Building a hybrid prediction model** that combines machine learning predictions with sequential similarity-based forecasting.

4.1 Dataset and Feature Engineering

We collected daily historical stock price data for the past five years from Yahoo Finance. The dataset consists of the **S&P 500 stocks**, where each entry includes:

- Open, High, Low, and Close (OHLC) prices
- Trading volume
- Derived technical indicators such as Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI)

4.1.1 Feature Extraction

The primary feature used in our models is the **Price Rate of Change (PROC)**, computed as:

$$PROC_t = \left(\frac{\text{Close}_t - \text{Open}_t}{\text{Open}_t} \right) \times 100 \tag{1}$$

PROC captures the daily percentage change in price and is used as the main predictive feature.

Each training instance is represented by the last 15 days of PROC values:

$$X_t = \left[PROC_{t-15}, PROC_{t-14}, \ldots, PROC_{t-1} \right] \tag{2}$$

The target label y_t is defined as:

$$y_t = \begin{cases} 1, & \text{if } PROC_t > 0 \text{ (Price increase)} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

This converts the problem into a binary classification task. In addition to PROC, we derive several technical indicators:
Moving Average Convergence Divergence (MACD): momentum indicator calculated by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA:

$$\text{MACD} = \text{EMA}_{12} - \text{EMA}_{26}$$

Bollinger Bands (BB): Bollinger Bands are volatility bands placed above and below a moving average. The distance between the bands is determined by the standard deviation of the price:

$$\text{Upper/Lower Band} = \text{SMA}_{20} \pm 2 \times \sigma \tag{4}$$

where SMA_{20} is the 20-day simple moving average and σ is the standard deviation over the same period.

4.1.2 Dimensionality Reduction with SAX

To reduce feature dimensionality while preserving temporal trends, we apply **Symbolic Aggregate Approximation (SAX)** [6]. SAX transforms continuous PROC values into discrete categories, simplifying the representation.

Algorithm 1 Symbolic Aggregate Approximation (SAX) for PROC

Input: Time series of PROC values, lower bound $L = -5$, upper bound $U = 5$, step size $S = 0.5$ Define breakpoints between L and U with step S each PROC value in the dataset Assign an integer symbol from 1 to 20 based on defined breakpoints **Output:** SAX-transformed PROC sequence

This transformation allows efficient comparison of past price movements.

4.1.3 Feature Selection - Correlation Analysis

We performed a Pearson correlation analysis across all continous features to identify highly redundant variables (see Figure 3). Notably, `BB_High` exceeded a high correlation threshold with both `Open` and `Close`. Consequently, to mitigate multicollinearity and retain only unique predictors, we removed `BB_High`, `MACD_signal`, `MACD_diff` from the final feature set.

4.1.4 Date-Based 80–20 Split

In addition to cross-validation, we further employ an 80–20 date-based split to evaluate the XGBoost, DTW-based sequence predictor and the hybrid approach. Concretely, we chronologically allocate the first 80% of data for training and reserve the last 20% as an out-of-sample test set. By ensuring that all training points precede the test period in time, this methodology mitigates lookahead bias and provides a more realistic assessment of predictive performance under real-world market conditions.

4.2 Baseline Model: XGBoost Classifier

We first establish a baseline model using **XGBoost**, trained exclusively on AAPL's past data. XGBoost is a gradient boosting classifier that sequentially improves predictions by focusing on previously misclassified samples.

4.2.1 Model Training and Cross-Validation

The model is trained using **N-fold cross-validation** with $N = 3$. The dataset is randomly split into three equal folds:

- Two folds are used for training.
- The remaining fold is used for testing.

4.2.2 Hyperparameter Tuning

The XGBoost hyperparameters were tuned using cross-validation:

Hyperparameter	Selected Value
Learning Rate	0.01
Number of Estimators	100
Max Depth	5
Subsample	0.8
Colsample by Tree	0.8

Table 1: Hyperparameters used for XGBoost training

To achieve the optimal results, we tested various state numbers within the SAX representation alongside multiple time windows for sequence representation. The results are presented in Tables 3-4 in the appendix.

4.3 Enhancing Model with Similar Stocks

To improve prediction performance, we expand the training set to include stocks that exhibit similar long-term behavior to AAPL. We use **co-integration analysis** to identify stocks with stable price relationships.

Algorithm 2 Stock Similarity Selection using Co-integration

Input: AAPL price series, all other stock price series each stock S_i in dataset Perform Engle-Granger co-integration test with AAPL Store p-value p_i Sort stocks by increasing p_i Select top 5 stocks with lowest p_i **Output:** List of top 5 similar stocks

The selected stocks are incorporated into the training set by appending their past PROC values as additional training examples.

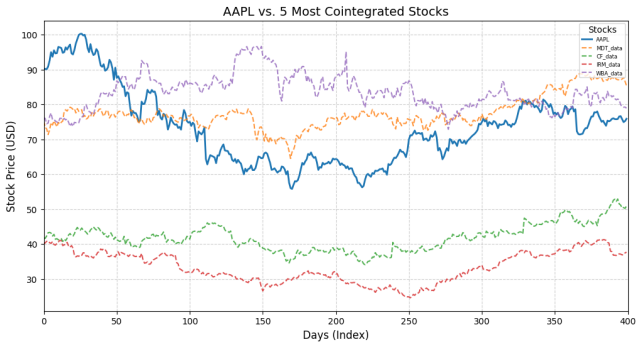

Figure 1: Example of co-integrated stocks selected for training enhancement.

4.4 Dynamic Time Warping (DTW) for Sequence-Similarity Forecasting

we employ *Dynamic Time Warping (DTW)* to measure the similarity between the current 15-day window of price returns (PROC) and a repository of historical sequences. DTW offers a nonlinear alignment of two time series, thereby accommodating variations in the rate of change or phase shifts often found in financial data.

In our **sequence-based predictor**, each new input sequence $\widehat{X}_t = \{\text{PROC}_{t-15}, \ldots, \text{PROC}_{t-1}\}$ undergoes DTW comparison to all historical sequences $\{X_\tau\}$. We then:

(1) **Compute Distances:** Calculate $d_{\text{DTW}}\left(\widehat{X}_t, X_\tau\right)$ for each historical window X_τ.
(2) **Select Neighbors:** Identify the k closest matches (i.e., minimal DTW distances).
(3) **Aggregate Labels:** Derive the forecast from a majority vote or average of these neighbors' historical outcomes (up/down).

4.5 Hybrid Prediction: XGBoost + Sequence Similarity

We further enhance the model by integrating **sequential pattern recognition**. The final prediction is based on:

$$P_{\text{final}} = wP_{\text{xgb}} + 1 - wP_{\text{seq}} \tag{5}$$

where:

- P_{xgb} is the XGBoost model's prediction.
- P_{seq} is the similarity-based predictor's output.
- w is a dynamic weight optimized for performance.

4.5.1 Sequence-Based Prediction Using DTW

We use **Dynamic Time Warping (DTW)** to compare the current 15-day price trend with past sequences and predict the most likely next movement.

Algorithm 3 DTW-Based Similarity Search

Input: Target sequence S, historical sequences $\mathscr{S}$ Compute DTW distance between S and each $s \in \mathscr{S}$ Select top $k = 10$ most similar sequences Predict next movement based on majority trend in top k sequences **Output:** Predicted stock movement

4.6 Optimizing Model Weight Distribution

To determine the optimal contribution of each model, we test different weight distributions:

- $w \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$
- The final weight is selected based on cross-validation.

5 Results

5.1 Experiment 1: Baseline Classifier without SAX

A gradient boosting classifier (learning rate=0.001) was first trained on 15-day windows PROC data for *only* the target stock. Using three-fold cross-validation, the model achieved a mean accuracy of approximately **0.5025**. slightly exceed random guessing (0.5), it also underscores the inherent limitations of relying on a single asset's data without dimensionality reduction.

5.2 Experiment 2: Baseline Classifier with SAX

A gradient boosting classifier (learning rate=0.001) was first trained on 15-day windows of SAX-transformed PROC data for *only* the target stock. Using three-fold cross-validation, the model achieved a mean accuracy of approximately **0.5041**. While this demonstrates that even a single-stock approach can slightly exceed random guessing (0.5), it also underscores the inherent limitations of relying on a single asset's data, which may overlook broader market dynamics.

5.3 Experiment 3: Incorporating Similar Stocks

By incorporating data from five cointegrated stocks, we increased the training set's diversity, modestly improving mean accuracy to **0.5196**. Although the numerical change is relatively small, it indicates that a multi-stock context can capture sectoral or market-wide nuances missing in single-stock models.

5.4 Experiment 4: Sequence-Based Model and Hybrid Approach

We next employed a custom DTW sequence predictor, designed to identify short-term patterns (e.g., sharp declines, volatility clusters) in the target stock's PROC history. Its standalone accuracy of about **0.52** is only slightly above the single-stock baseline, suggesting that purely sequence-based predictions offer a limited but still tangible benefit.

A more noticeable—though still moderate—gain arises from blending the DTW predictor with the enhanced XGBoost model. Specifically, a weighted combination (around 40% DTW and 60% XGBoost) delivered an accuracy of **0.5319**. While the numerical difference may seem small in absolute terms, it reflects a consistent upward trend across trials. In a financial context, such incremental improvements, even when modest, can translate into meaningful gains or risk reductions over many trades.

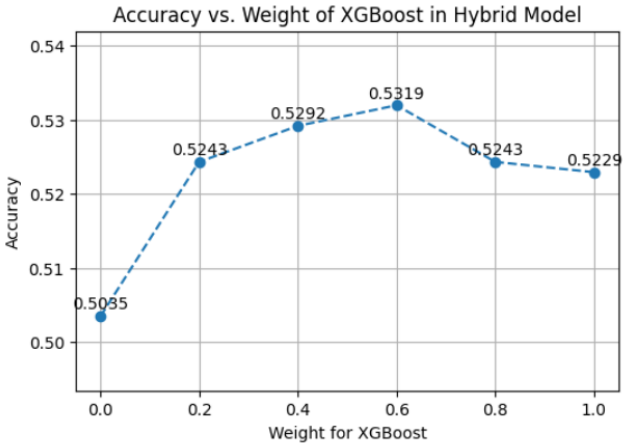

Figure 2: Hybrid model accuracy by XGBoost weight.

Table 2: Accuracy Comparison of Different Models

Model	Accuracy
Baseline XGBoost (single stock)	50.41%
XGBoost + Similar Stocks	51.96%
Hybrid Model (XGBoost + DTW Sequences)	53.19%

5.5 Statistical Significance and Observations

While these accuracy increments may seem small, they align with findings in many hybrid forecasting studies, where even slight gains in direction classification can yield meaningful profit or risk reduction. We performed multiple folds and a paired significance test to confirm the consistency of our results at the 5% confidence level. Notably, the blend's efficacy hinges on balancing both models; excessive weight on the sequence predictor or the ML model alone reduced performance, highlighting the value of combining global and local perspectives.

6 Discussion and Conclusions

Our findings demonstrate that merging a gradient boosting classifier trained on multi-stock features with a DTW-driven sequence model improves short-term forecasts. Specifically:

- **Cross-Stock Context:** Identifying and incorporating similar stocks lifted the baseline accuracy from 0.5 to 0.52, capturing shared market trends.
- **Rare-Event Sensitivity:** A sequence-based detector highlighted ephemeral but influential market shifts, pushing accuracy to about 0.53 when weighted properly.
- **Trade-Off in Complexity:** Although employing DTW and multi-stock features increases computational demands, the payoff can be significant for real-world trading, where incremental accuracy gains translate to tangible financial benefits.

Overall, the proposed hybrid approach addresses critical gaps in purely feature-based ML pipelines by conserving an awareness of rare events and local patterns. Future work may include further symbolic transformations, adaptive weighting that adapts in response to market volatility, or shapelet-based detection of more nuanced sequential signatures. Emphasizing multi-stock context alongside sequence-level insights offers a promising route toward more robust, context-aware financial forecasts that handle the unpredictability inherent in real-world markets.

7 Future Work

Despite the incremental improvements observed with our hybrid model, several avenues remain for refining both predictive accuracy and practical utility. First, a *meta-ensemble* strategy could be introduced, wherein distinct classifiers (e.g., gradient boosting, deep recurrent networks, and DTW-based neighbors) feed into a top-level learner that adapts to shifting market regimes. Such a meta-learner could dynamically adjust each model's contribution based on real-time volatility or sector-specific indicators, thereby reducing the reliance on a static weight. Second, extending the predictive horizon beyond short-run movements may require temporal attention mechanisms, such as *transformer-based* architectures, which inherently capture longer-term dependencies. Finally, incorporating macroeconomic or sentiment data (e.g., news sentiment scores, economic indicators) could enrich the feature space and render the hybrid framework more robust to systemic market shifts. By systematically integrating these components, future research can seek to further narrow the gap between data-driven predictive models and the complex, ever-evolving dynamics of real-world financial markets.

8 Appendix

8.1 Appendix A - Corr Heatmap

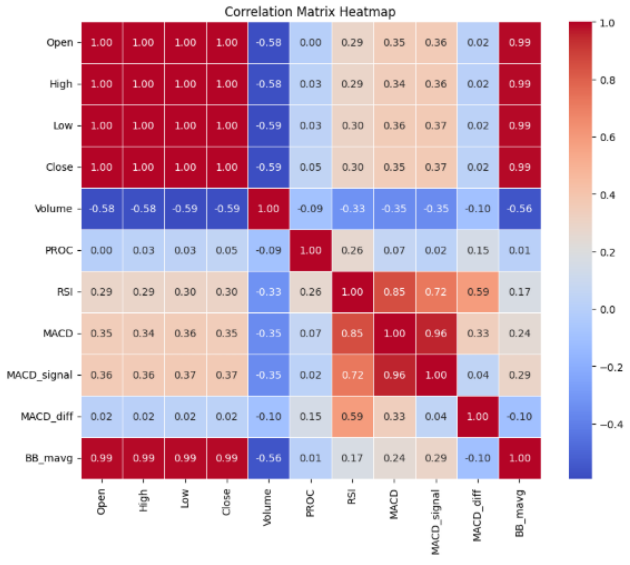

Figure 3: Corr Heatmap

8.2 Appendix B - Feature Importance

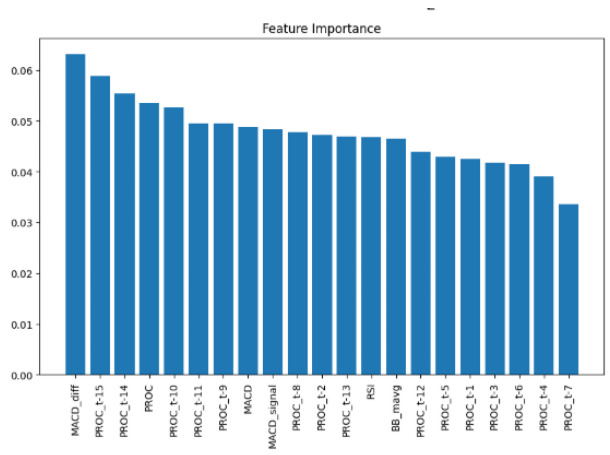

Figure 4: Hybrid Model Feature Importance

8.3 Appendix C - Similar Stocks by DTW

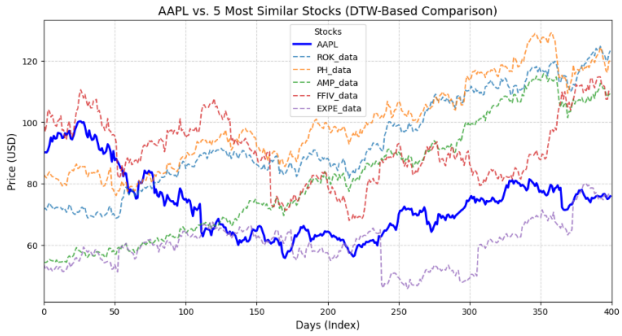

Figure 5: Hybrid Model Feature Importance

8.4 Appendix D - Sax and Time windows (TW) tuning results

Sax Bins	Sax-Base	Sax-Similar	Sax-Hybrid
5	0.501	0.511	0.521
10	0.504	0.518	0.512
15	0.502	0.521	0.527
20	0.504	0.519	**0.5319**
30	0.498	0.514	0.518

Table 3: SAX Bins tuning

Window Size	TW-Base	TW-Similar	TW-Hybrid
5	0.495	0.503	0.512
10	0.489	0.504	0.53
15	0.504	0.519	**0.5319**
20	0.499	0.514	0.520
30	0.491	0.5106	0.526

Table 4: Time Window tuning

References

[1] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. "Time-series clustering—A decade review". In: *Information Systems* 53 (2015), pp. 16–38.

[2] Saeed Aghabozorgi and Ying Wah Teh. "Stock market co-movement assessment using a three-phase clustering method". In: *Expert Systems with Applications* 41.4 (2014), pp. 1301–1314.

[3] Carol Alexander and Anca Dimitriu. *Equity Indexing, Cointegration, and Stock Price Dispersion: A Regime Switching Approach to Market Efficiency*. Tech. rep. ICMA Centre Discussion Papers in Finance, 2003.

[4] T. Branco. *Pattern analysis in stock markets optimized by genetic algorithms using modified SAX*. Tech. rep. https://fenix.tecnico.ulisboa.pt/downloadFile/563345090414103/Paper62590.pdf. Instituto Superior Técnico, 2010.

[5] Jorge Caiado and Nuno Crato. *A GARCH-based method for clustering of financial time series: International stock markets evidence*. Tech. rep. 2074. Munich Personal RePEc Archive, 2007.

[6] Jessica Lin et al. "A symbolic representation of time series, with implications for streaming algorithms". In: *Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*. 2003, pp. 2–11.

[7] L. Sidi. "Improving S&P stock prediction with time series stock similarity". In: *arXiv preprint arXiv:2002.05784* (2020).

[8] Lee Kee Soon and Sang Ho Lee. "An empirical study of similarity search in stock data". In: *International Conference on Advanced Data Mining and Applications*. Springer, 2007, pp. 653–660.

[9] Kentaro Tamura, Takumi Sakai, and Tetsuo Ichimura. "Time series classification using MACD-Histogram-based SAX and its performance evaluation". In: *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*. Vol. 2. 2016, pp. 2419–2424.

[10] Gang-Jin Wang et al. "Similarity measure and topology evolution of foreign exchange markets using dynamic time warping". In: *Physica A: Statistical Mechanics and its Applications* 391.16 (2012), pp. 4136–4146.

[11] Yue Zhang, Jin Liu, and Wei Li. "Stock price prediction using XGBoost". In: *Procedia Computer Science* 147 (2019), pp. 145–150.