Temporal Fusion Transformers for Enhanced Multivariate Time Series Forecasting of Indonesian Stock Prices

Standy Hartanto[1], Alexander Agung Santoso Gunawan[2]
Computer Science Department-Master of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia[1]
Computer Science Department-School of Computer Science, Bina Nusantara University, Jakarta 11480, Indonesia[2]

***Abstract*—The stock market represents the financial pulse of economies and is an important part of the global financial system. It allows people to buy and sell shares in publicly held corporations. It serves as a platform for investors to trade ownership in businesses, enabling companies to raise capital for expansion and operations. However, the stock market can be very risky for any investor because of the fluctuating prices and uncertainties of the market. Integrating deep learning into stock market analysis enables researchers and practitioners to gain a deeper understanding of the trends and variations that will improve investment decisions. Recent advancements in the area of deep learning, more specifically with the invention of transformer-based models, have revolutionized research in stock market prediction. The Temporal Fusion Transformer (TFT) was introduced as a model that uses self-attention mechanisms to capture complex temporal dynamics across multiple time-series sequences. This study investigates feature engineering and technical data integrated into the TFT models to improve short-term stock market prediction. The Variance Inflation Factor (VIF) was used to quantify the severity of multicollinearity in the dataset. Evaluation metrics were used to evaluate TFT models' effectiveness in improving the accuracy of stock market forecasting compared to other transformer models and traditional statistical Naïve models used as baselines. The results prove that TFT models excel in forecasting by effectively identifying multiple patterns, resulting in better predictive accuracy. Furthermore, considering the unique patterns of individual stocks, TFT obtained a remarkable SMAPE of 0.0022.**

Keywords*—Time series forecasting; stock price prediction; capital market; technical analysis; TFT

I. Introduction

Stock market indices show the health of the economy. It allows people to trade in the shares of publicly held corporations. It serves as a platform for investors to trade ownership in businesses, enabling companies to raise capital for expansion and operations. Changes in the equity markets will indicate the economic situation, investors' sentiment, and expectations about future economic performance. This can be very risky for any investor because of the price fluctuations and uncertainties of the market. In addition, stock markets play a vital role in determining the companies' value. Prediction of stock prices is a very complex and highly challenging task due to the intrinsic volatility and multi-dimensionality of financial markets [1]. Indeed, most traditional models are challenged to identify exactly the complex trends and variables that impact stock price movements.

Research on stock market prediction has paid considerable attention to deep learning algorithms in recent years. Some techniques involve training models using large datasets to come up with complex patterns and correlations [2, 3, 4, 5]. These may also combine other data sources, like financial and non-financial data, in a model to be trained for the increase in prediction accuracy [6]. Researchers have just commenced exploring how Transformer-based models [7, 8] apply alongside Reinforcement Learning for forecasting trends in the stock market [9, 10, 11].

TFT is a new transformer-based model for handling multiple time series sequences with complicated temporal dynamics. By merging the LSTM Sequence-to-Sequence framework with the self-attention mechanism of Transformers, TFT adeptly captures temporal dependencies across varying scales while enriching temporal representations with static contextual information about measured entities. In contrast to RNN-based models, Transformers offer expedited processing by simultaneously ingesting all input, thereby bypassing the sequential nature of RNNs. Moreover, it is easier to train Transformers because they have fewer parameters compared to LSTM networks. Transfer learning is also possible with Transformers, which is not the case with LSTM networks. Notably, TFT fits the detailed subtleties within hydrographs, peaks, and transitional phases much more effectively than both LSTM and Transformers.

Generalizing models may not provide valuable insights to analysts and investors in view of the uniqueness of trends and patterns that individual stocks exhibit to make meaningful short-term decisions. It needs more comprehensive indicators that directly impact the market behavior. This involves identifying a few exogenous input features that help the model recognize the patterns of history, evaluate its performance against real market fluctuations so that it can be agile to volatility, and thereby provide valuable insights for short-term stock price predictions.

In this research, five years of stock market data from the Indonesia Stock Exchange (IDX) were used in analyzing with the TFT model, particularly in mining, communications, and industrial sectors. The main objective is to develop a comprehensive analysis of TFT with regards to short-term stock price prediction by using historical technical indicators.

In addition, several feature engineering techniques, model architectures, and training strategies are also evaluated for their effects on the accuracy of prediction. Evaluation metrics were used against baselines composed of Transformer models and Naïve models, comparing the performance of TFT. Lastly, the model was tested under real-market conditions to evaluate its performance in generating accurate predictions on the stock exchange.

II. Related Studies

In this vast domain of time series forecasting, there exist a large number of theories and methodologies that act as the foundation for predictive analysis in different fields. These models provide insight into future trends and events that range from classic statistical approaches to modern deep learning algorithms [12].

Naïve models are very simple, fairly easy to use, and provide a simple starting point in developing or predicting stock prices. They are quick to calculate and inform us about the performance of a more complex method compared to something quite simple. While being relatively interpretable and tolerant of noise, they may miss small details that drive stock prices [13]. In this Naïve approach, each estimate is set based on the last observed value.

$$\hat{y}_{T+h|T} = \text{yT} \quad (1)$$

Transformers have gained prominence in Natural Language Processing (NLP) and computer vision, their application in the realm of time-series data remains relatively unexplored. Our approach addresses this gap through a self-attention mechanism that helps identify complex nonlinear trends and intrinsic dynamics in time series data, which are consolidated under high volatility and nonlinearity. The predictive power of our model includes providing closing price forecasts for the next trading day with insights derived from multiple stock price inputs. Our model is rigorously validated by testing through four different error evaluation metrics. The fact that our model can predict the closing prices with a probability of more than 90% makes this model very useful for fintech [14].

Employing a combination of CNNs, RNNs, LSTMs, and BERT, alongside textual data from social media. It is posited that, by incorporating deep learning models with the state-of-the-art BERT word embedding model, classification performance will be improved. When such deep learning algorithms are combined with such a state-of-the-art natural language processing model, it incurs improvement in performance every time. In predicting stock directional movement, it leads to up to 96.26% accuracy performance [15].

LSTM neural network models are suitable for monitoring trends and capturing seasonality over long forecast periods. A study [16] reveals an increase in model performance with a new approach that uses six variables: High, Low, Open, Volume, HiLo, and OpSe. Give rise to the urge to explore new forecasting strategies with respect to the various scenarios that can be studied. These efforts can provide meaningful insights for investors and analysts who want to understand the working mechanisms of the stock market to better grasp future trends [17].

Recently, studies on the application of Transformer-based models and Reinforcement Learning (RL) models in stock market forecasting have already been initiated. The purpose of the survey is to consolidate the latest developments in methodologies like Transformers and RL with in-depth analysis and discourse on their implications and advancement in this domain [9, 10].

Temporal Fusion Transformer is a model architecture designed for time series forecasting. It intrinsically combines the concepts of transformers very successfully in natural language processing and related sequence data tasks with techniques specifically developed for dealing with temporal data [18, 19, 20]. The primary function of TFT is to enhance learnt temporal representations with static data about measured entities and to capture temporal dependencies at various time scales using a combination of the Transformer's Self Attention mechanism and the LSTM Sequence-to-Sequence [21]. Transformers process all of the input at once, making them faster than RNN-based models [22]. Compared to transformer networks, LSTM networks require longer training due to their significantly larger parameter set. Furthermore, transfer learning is not feasible with LSTM networks. TFT is more effective than LSTM or Transformers at capturing the subtleties of the hydrographs, such as the peaks and limbs.

A study in [21] proposed the TFT model as a solution for multi-horizon time series forecasting, where the goal is to predict multiple future time steps of a sequence simultaneously. The TFT architecture generalizes transformer attention mechanisms and encoder-decoder structures to capture complex temporal patterns in the data while offering interpretability through attention weights. This study encompasses data from four major categories: Electricity, Traffic, Retail, and Volatility. The regional variables in the volatility category are indices of the Americas, Europe, or Asia. There are 31 stock indices in total with open-to-close returns acting as supplementary exogenous inputs, and the time span was from 2000 to 2019. Comparisons against TFT were made with a number of models DeepAR, ConTrans, Seq2Seq, and MQRNN with respective results including 0.050, 0.047, 0.042, 0.042, and 0.039.

Acknowledging the distinct trends and patterns observed in individual stocks, the study aims to evaluate the effectiveness of the TFT model in analyzing short-term trends in Indonesian stock prices, particularly within the mining, communications, and industrial sectors. It seeks to determine whether TFT models can provide valuable insights to analysts and investors for short-term decision-making purposes.

III. Research Methodology

Advanced methodology that drives our research is unveiled. It is imperative to establish a nuanced understanding of what kind of guiding principles and meticulous procedures we put in motion for an in-depth review, starting from careful data collection to rigorous analysis. Fig. 1 depicts the research stages.

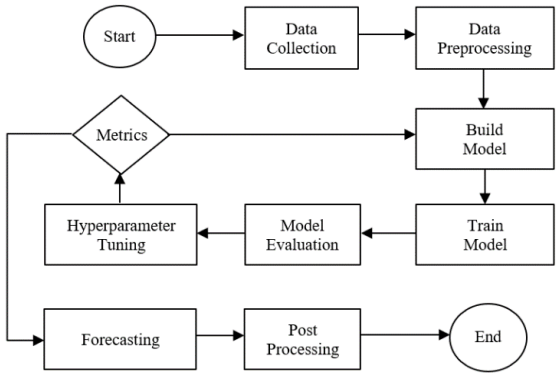

Fig. 1. Research stages.

A. Dataset

The landscape of this research utilized datasets sourced from emerging markets, the Indonesia Stock Exchange (IDX), in order to reflect a growing recognition of the significance of diversifying data sources to attain a more comprehensive understanding of global economic trends. This approach acknowledges the high probability of new trends, market behaviors, and investment patterns that might not be captured or be sufficiently represented within past research datasets.

The dataset is sourced from publicly available data provided by Yahoo!Finance [23]. General market stability and behavior will drive data requirements, in highly volatile markets or in exceptional circumstances such as an economic crisis, longer historical data may be required. However, longer periods of data might increase the risk of overfitting, where the model learns noise in the data rather than true patterns. The investigation contemplated the incorporation of data covering roughly the previous five years.

Each stock exhibits a unique pattern. To acquire comprehensive insights, three different stock analyses were made for Aneka Tambang (ANTM) in the mining sector, XL Axiata (EXCL) in the communication sector, and Astra International (ASII) in the industrial sector. The transaction was specified for the date range from March 27, 2019, to March 27, 2024.

Table I presents a selection of five examples taken from the ANTM stock dataset, showing which main features are detected on each trading day. The main variables regarding the dataset include Close and Volume.

TABLE I. SUMMARY OF STOCK EXTRACTION DATA

Date	Close	Volume
2024-03-15	1490	51840100
2024-03-18	1526	66602000
2024-03-19	1531	49266200
2024-03-20	1531	38425600
2024-03-21	1568	85481500

"Close" represents the price level at which the last trade took place when the market closed for the day. "Volume" shows the degree of activity or liquidity for that stock on the market.

B. Data Preprocessing

In this study, historical stock data was retrieved from Yahoo!Finance. After acquiring the data, some of the columns were removed as they were irrelevant for the analysis. Sorting the dataset in order of date to keep chronological order is a very critical factor in almost all forms of time series analysis. Adding more features to a model is one step toward better capturing the nuanced relationships and dependencies existing within financial markets and hence leads to more accurate and robust predictions of stock prices from the model. Other variables that were included in this dataset to enhance the predictive power of the model were the gap between the opening and closing prices and indicators for working days and months. These new variables provide insight into temporal trends and behavior of the markets, which could not have been done otherwise, and thus facilitate better predictions.

- Gap between opening and closing price: The model will understand days in which prices move highly compared with days when prices remain unchanged. This will help the model understand short-term price trend predictions.
- Working days and months: These variables enabled the model to accommodate known breaks or closures within the markets and seasonal adjustments in demand.

Despite TFT's capability to manage multicollinearity, VIF was utilized in this study to converge the results under the statistical requirements. The VIF was calculated for each predictor variable to measure the degree of multicollinearity. VIF refers to the measure of how much multicollinearity inflates the variance of a regression coefficient. Table II presents the results of VIF.

TABLE II. EVALUATION OF MULTICOLLINEARITY

Variables	Variance Inflation Factor (VIF)		
	ANTM	EXCL	ASII
Close	4.186406	5.989167	6.953210
Volume	2.413983	2.115669	3.727508
Gap_Open_Close	2.784901	2.486082	2.675353
Months	2.641024	3.239180	3.317118
Working_Days	2.482500	2.854557	2.880108

C. Proposed Method

In this section, a discussion and description regarding several deep learning methodologies are presented, followed by careful integration of these methods into the proposed model architecture.

Attention mechanisms are key components that allow the model to selectively focus on different segments in the input sequence while processing the temporal data for purposes of forecasting. Attentional mechanisms thus play a very important role in capturing complex temporal patterns, especially temporal interdependencies across a variety of time steps.

Recurrent Neural Network (RNN) stands as a deep learning model designed to process and transform sequential data inputs into specific sequential data outputs. Such sequential data

typically encompasses words, sentences, or time-series data, where sequential elements are interconnected through complex semantic and syntactic rules.

Long Short-Term Memory (LSTM) is one such subtype of RNN, it is applied to sequence data to identify any underlying patterns within it. There may be present sequence data in the form of sensor readings, stock prices, or natural language. All these, while taking the position in the sequence of not only the actual value into account, are obtained during the prediction phase.

The Transformer, a deep learning architecture reliant on attention mechanisms [24], distinguishes itself by necessitating shorter training times compared to preceding recurrent neural architectures like LSTM. More precisely, this model accepts tokenized input tokens and, at each layer, contextualizes each token concurrently with other input tokens through attention mechanisms. Through their self-attention mechanisms, these models adeptly discern patterns spanning extensive sequences, effectively weighing the significance of each time step for accurate predictions. Parallel processing capabilities of Transformers expedite training and inference, useful for long time series. Moreover, by construction, Transformers inherently learn meaningful features from data, avoiding thorough manual feature engineering. By design ready to scale up and adapt, the Transformers are tailored to decode complex relationships in time and positions them as very powerful tools to uncover insights and predict trends within time series data.

The Temporal Fusion Transformer (TFT) represents a transformer-derived model utilizing self-attention mechanisms to grasp the intricate temporal variations across multiple time sequences. It stands as a potent tool for addressing multi-horizon and multivariate time series forecasting scenarios.

TFT uses time-dependent exogenous input features, which are made up of apriori unknown inputs (z) and known inputs (x), as well as static covariates (s), which offer contextual metadata about measured entities that is independent of time, to predict the future. Past target values (y) within a look-back window of length k are used as input. TFT uses quantiles to output prediction intervals rather than just a single value. At time t, every quantile q forecast of τ-step-ahead is expressed as follows:

$$\hat{y}_i(q, t, \tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i) \qquad (2)$$

Where, $_q$: quantile, $y_{i,t-k:t}$: historical target values, $z_{i,t-k:t}$: unknown inputs, $x_{i,t-k:t+\tau}$: known inputs, s_i: static covariates.

The proposed method is visualized in Fig. 2.

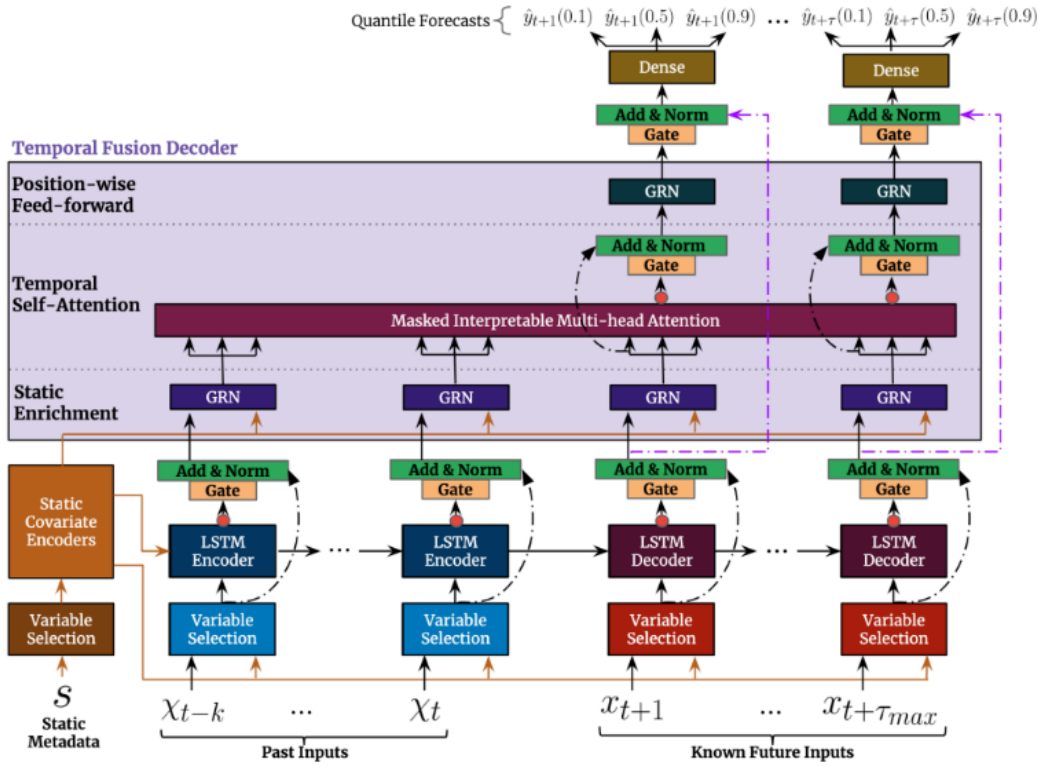

Fig. 2. The TFT Architecture [21].

To improve the flexibility of the TFT architecture, Gated Residual Networks (GRN) are incorporated into several layers of the architecture. They accomplish this by adding skip/residual connections, which transfer a layer's output to higher, non-adjacent levels in the network. As a result, the model has the ability to identify superfluous non-linear processing layers and exclude them. GRN dramatically lowers the number of parameters and processes needed while enhancing the model's generalization capabilities across a variety of application contexts. Fig. 3 illustrates the GRN architecture. ELU stands for Exponential Linear Unit activation function.

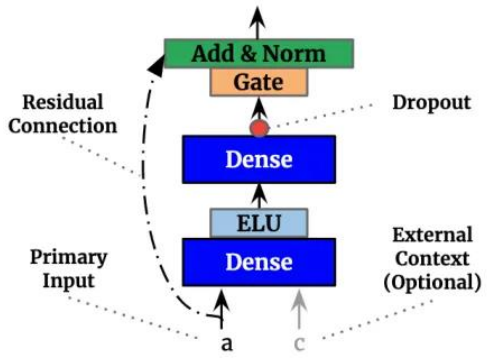

Fig. 3. Gated Residual Network [21].

In order to enable temporal variable selection, local temporal representation processing in the Sequence-to-Sequence layer, and static temporal representation enrichment, static covariate encoders obtain context vectors from static metadata and embed them into various TFT network segments. The conditioning of temporal representation learning with static data is made possible by this integration.

A different variable selection block is constructed for every type of input in the variable selection network, which includes static covariates, past inputs (both known and unknown that vary over time), and known future inputs. By learning to assess the importance of every input feature, these blocks allow the Sequence-to-Sequence layer that follows to handle the reweighted sums of the transformed inputs at each time step. Learned linear transformations of continuous data and entity embeddings of categorical features are examples of transformed inputs. Thus, the variable selection block of static covariates omits the external context vector, which is obtained from the output of the static covariate encoder block. Fig. 4 illustrates the Variable Selection architecture.

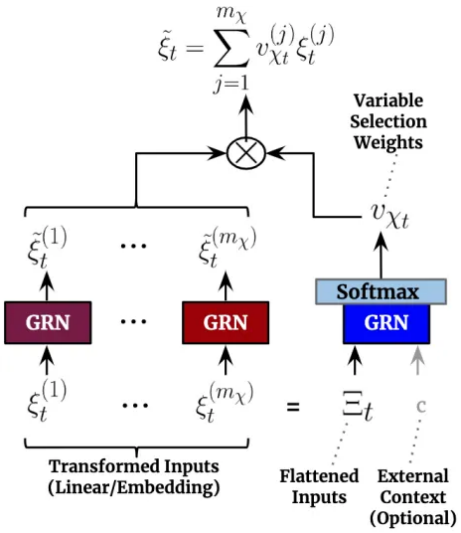

Fig. 4. Variable Selection Network [21].

The TFT network substitutes a Sequence-to-Sequence layer for the positional encoding commonly found in Transformers in the Sequence-to-Sequence component. Due to its ability to capture local temporal trends through recurrent connections, this adaptation is more suited for time series data. This block uses context vectors to initialize the first LSTM unit's cell state and concealed state. Additionally, they add to the static enrichment layer by adding static data to the temporal representation that was learned from the Sequence-to-Sequence layer.

Value relevance is evaluated by the Interpretable Multi-head attention mechanism on the basis of the connections between keys and queries. It works similarly to information retrieval in that it finds the most pertinent documents (values) by comparing a search query (query) to document embeddings (keys) [25]. Fig. 5 shows the adjustments made by the TFT to ensure interpretation. Instead, it shares many head-specific weights for values across all the attention heads.

$$InterpretableMultiHead\ (Q,K,V) = \frac{1}{h}\sum_{i=1}^{h} head_i W_H$$

$$where\ head_i = Attention\left(QW_Q^{(i)}, KW_K^{(i)}, VW_V\right) \quad (3)$$

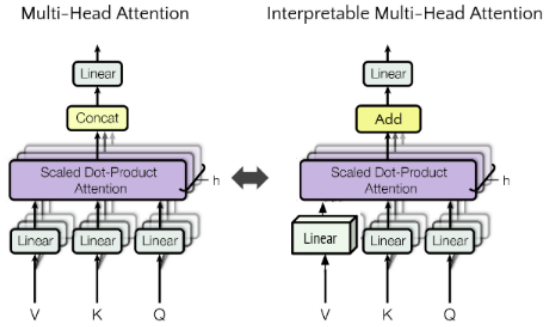

Fig. 5. Interpretable Multi-Head Attention [25].

Using a combination of the Transformer's Self Attention mechanism and the LSTM Sequence-to-Sequence, TFT was utilized to augment learnt temporal representations with static data about measured entities and to capture temporal dependencies at various time scales. In this study, a historical data window size of 12 was employed, with a prediction horizon of 3 for forecasting stock prices.

Table III presents a comprehensive outline of the TFT method utilized for forecasting stock prices.

TABLE III. TFT ALGORITHM

Algorithm 1: TFT	
Input	: Dataset [Close, Volume, GapOpenClose, Month, Day]
Output	: Prediction Result [Closing Price]
1.	**Start**:
2.	Load the dataset
3.	Preprocess the dataset
4.	Split dataset into data(train), data(val)
5.	WS = Initialize window size
6.	H = Initialize horizon
7.	Model ← build_model(TFT)
8.	Model ← train_model(data(train))
9.	Model ← optimize_hyperparameters(data(val))
10.	Model ← evaluate the model's performance
11.	Model ← save the best model
12.	MAE, MAPE, SMAPE ← (Model, data(val))
13.	Prediction ← (Model(WS,H), data(val))
14.	**Return Prediction**

D. Evaluation Metrics

This research incorporates prevalent loss functions for time-series forecasting, MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and SMAPE (Symmetric Mean Absolute Percentage Error). The respective equations for each loss function are computed as follows:

- MAE measures the average absolute difference between the predicted values and the actual values.

$$MAE = \frac{1}{N}\sum_{i}^{N}|y_i - \hat{y}_i| \quad (4)$$

- where, N: number of observation, y_i: the actual value of the i^{th} observation, $\hat{y}_i$: the predicted value of the i^{th} observation.

- MAPE measures the average absolute percentage difference between actual and predicted values [26].

$$MAPE = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right| \quad (5)$$

- where, N is the number of data points, At and F_t denote the actual and forecast values at data point $_t$, respectively.

- SMAPE calculates the percentage error for each data point, but it takes into account the scale of the actual and forecasted values by using their average.

$$SMAPE = \frac{100\%}{n}\sum_{t=1}^{n}\frac{|F_t - A_t|}{(|A_t| + |F_t|)/2} \quad (6)$$

- where, n is the number of data points, F_t is the forecasted value, A_t is the actual value.

E. Training Procedure and Computational Cost

Data splitting was carried out based on window size and horizon. It commences by ordering the dataset based on transaction dates. Subsequently, the dataset undergoes segmentation into training and validation sets. Spanning 5 years, the dataset comprises 1227 rows per individual stock, divided into 90% for training and 10% for validation.

It used a window size of 12 days, refers to the number of prior time steps in consideration while predicting future time steps. The horizon parameter was set to three days, and its meaning was how far the forecasting horizon was to be projected into the future.

Close, Volume, and GapOpenClose were used as exogenous inputs, complemented by working days and months as known categoricals, which are indispensable for predicting closing price as the output target. The training and validation processes were executed on a computer equipped with a 2.3 GHz Intel Core i7 quad-core CPU and 16GB of RAM. It was estimated that each of the individual final models would complete training in less than 30 minutes and use approximately 89% of the CPU's computational resources. Variability of patterns between the different stocks posed a challenge because optimality in hyperparameters configuration had to be identified. Moreover, the extended duration of model training posed a significant obstacle.

IV. Result and Discussion

Based on research findings, TFT models have been very effective in problems of time series forecasting, especially in short-term stock price prediction. It has been shown that the model is capable of handling complicated and dynamic temporal patterns in stock price data [21], drawing from information in multiple variables including seasonality. The superiority of TFT is further manifested in its flexibility when trends change. Compared with Transformer models and Naïve models, the TFT models perform better and provide more accurate predictions. TFT is an effective and sophisticated way to increase accuracy and precision in time series forecasting analysis [21].

While TFT has demonstrated significant effectiveness in resolving time series forecasting issues, it is necessary to admit that some element of uncertainty always remains in the stock market. The dynamics of the market may alter due to some unpredictable events, sudden economic changes, or other external events that may remain hidden in historical data alone [27]. Other complementary strategies would be the incorporation of real-time market sentiment analysis, macroeconomic indicators, or geopolitical events into the model in order to increase its performance [28, 29]. This would yield an all-inclusive view of dynamic market conditions to TFT and help in making better decisions due to the uncertainties that characterize changes in stock prices. Table IV presents the comparison metrics used for TFT, Transformer, and Naïve models.

TABLE IV. Performance Evaluation Metrics

Ticker	Model	Evaluation Metrics		
		MAE	***MAPE***	***SMAPE***
ANTM	TFT	3.3324	0.0022	0.0022
	Transformer	43.6452	0.0289	2.8845
	Naïve	35.0000	2.1027	2.0805
EXCL	TFT	9.7546	0.0041	0.0041
	Transformer	57.3425	0.0265	2.6567
	Naïve	115.4754	4.8732	4.7502
ASII	TFT	38.2241	0.0078	0.0078
	Transformer	84.3230	0.0167	1.6806
	Naïve	125.2116	2.5817	2.6285

The TFT models use multivariate data: Close, Volume, GapOpenClose, Month, Day. At the same time, Transformer and Naïve models use univariate data: Close. Based on the above-presented evaluation metrics, TFT models significantly outperform Transformer and Naïve models.

The closing price informs about performance and price trends, the volume conveys relevant information about trading activity. GapOpenClose allows highlighting of days with large price fluctuations, Time_Idx puts information into the context of time, while months and working days serve to enable the model to capture seasonal and daily trends. Based on several effective variables, the TFT model has proved to be very effective in predicting short-term stock prices, as it indeed picked up complex patterns hidden within the stock price time series data.

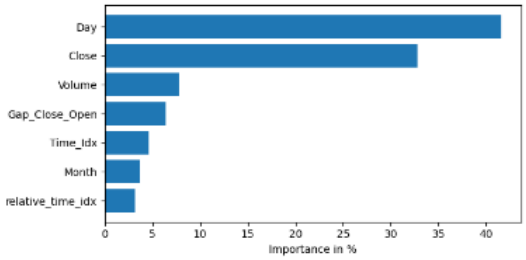

Fig. 6. ANTM encoder variables importance.

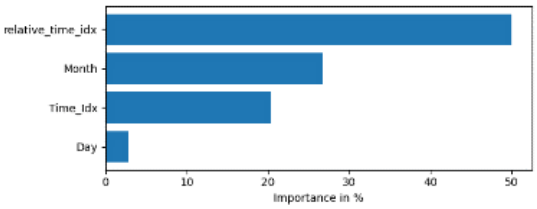

Fig. 7. ANTM decoder variables importance.

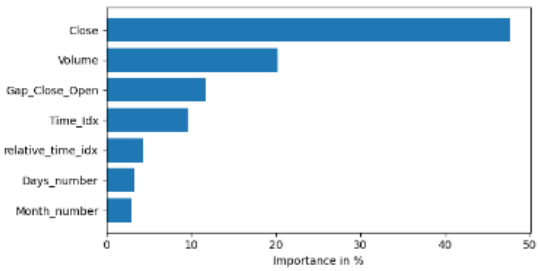

Fig. 8. EXCL encoder variables importance.

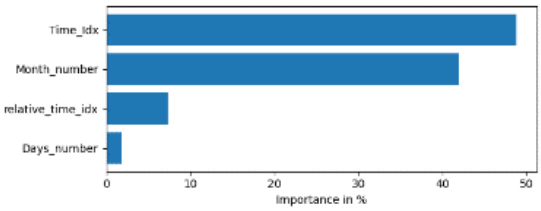

Fig. 9. EXCL decoder variables importance.

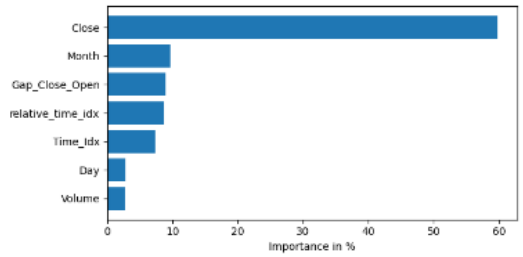

Fig. 10. ASII encoder variables importance.

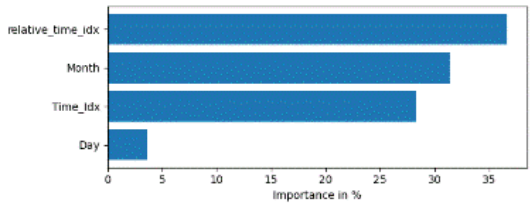

Fig. 11. ASII decoder variables importance.

Fig. 6 to Fig. 11 present evaluations of variables importance used in the encoder and decoder of the TFT models for forecasting ANTM, EXCL and ASII stocks. Encoder variables importance refers to how influential or informative the input variables are in the prediction task. It measures the effect of these variables on how well the model can capture and understand the patterns of data during it's encode phase. Decoder variables importance refers to the relevance of different features used during decoding.

Since every stock trend and patterns are different, the importance of encoder and decoder variables underline different priorities for each individual stock. This underlines the fact that customized approaches must be addressed for each individual stock.

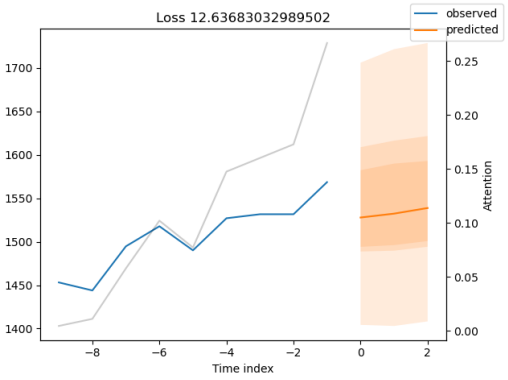

Fig. 12. ANTM prediction results.

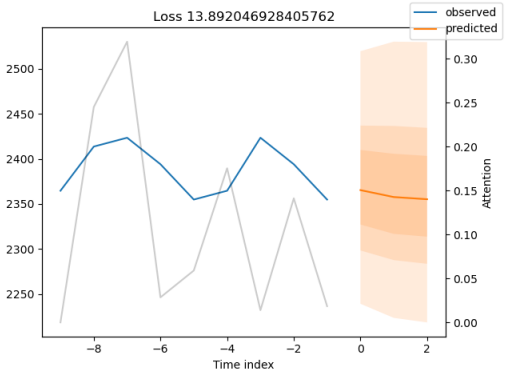

Fig. 13. EXCL prediction results.

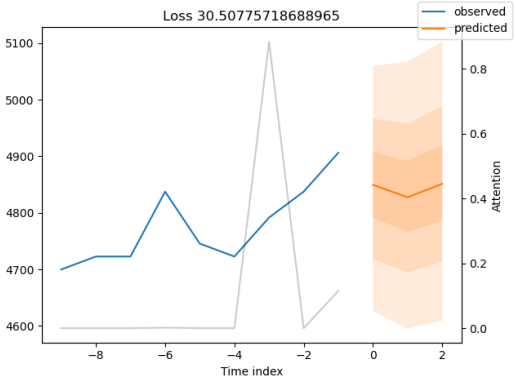

Fig. 14. ASII prediction results.

Fig. 12, 13, and 14 illustrates the prediction results within a real-world market scenario. The aim was to forecast three days ahead without having future data. The TFT model was employed to generate market predictions as of March 27, 2024. These predictions were subsequently compared with actual stock price movements observed at a later date.

The time index represents transaction dates in the dataset, numbers -6 to -1 denote historical dates given to the model, while 0, 1, and 2 are the predicted dates. Specifically, -1 corresponds to March 27, 2024 (the last date in the dataset), 0 corresponds to March 28, 2024, 1 corresponds to April 01, 2024, and 2 corresponds to April 02, 2024.

Grey lines in the plot represent the attention weights to understand the temporal patterns across past time steps. Observed line denotes the amount of attention the model pays to different points in time when making the prediction. Predicted line is an extrapolation, it refers to estimating an unknown value based on extending a known sequence of values or facts. Deviation in the prediction area is calculated using Quantile Loss, with output size=7.

$$\text{QuantileLoss}(\text{pred}, \text{outcome}) =$$

$$\max\{q(\text{pred}-\text{outcome}),(q-1)(\text{pred}-\text{outcome})\} \quad (7)$$

Following a comprehensive evaluation three days later, the TFT model has demonstrated exemplary performance by accurately mirroring real-world stock movements in subsequent days. A key differentiator is the TFT model's ability to adapt to stock volatility, a capability that the Transformer and Naïve models lack.

The approach enables the TFT model to recognize the repeated patterns in closing prices at different time frames, which include trends, cycles, and seasonal variations. This aids in deciding when to make entries and exits for investments. Furthermore, the TFT model uses the opening and closing price differences to identify the patterns that may indicate reversals or continuations in markets. Other contributors to the sentiment analysis include day-of-the-week and month effects. For instance, due to weekend outlooks, the market sentiment is usually optimistic on Fridays, or cautious on the first trading day of the month since there are economic data releases.

This work has emphasized that the TFT model is very effective at capturing temporal patterns. From an application perspective, the TFT model represents one of the more advanced tools available for development in pattern recognition and predictive modeling tools, providing investors and analysts with increased and empowered analytical skills in terms of spanning market dynamics, and making fully informed and reasoned short-term decisions.

V. Conclusion

In this study, we proposed a TFT model for stock price prediction by employing multiple variables to find the influence of each variable on stock price prediction. This approach achieved an outstanding MAPE score of 0.0022. Additionally, the TFT architecture is also applied to detect sudden fluctuations in stock markets, as can be seen from the results. Nevertheless, these fluctuations may not consistently manifest at regular intervals or adhere to identical cycles on each occasion. It is imperative to acknowledge the inherent unpredictability inherent in stock market dynamics. Future research aims to investigate the integration of emerging technologies, such as reinforcement learning, with the objective of augmenting the model's robustness and efficacy in discerning intricate and dynamic patterns.

References

[1] K. Biriukova and A. Bhattacherjee, "Using transformer models for stock market anomaly detection," Creative Commons, vol. 2023, 2023.

[2] T. Muhammad et al., "Transformer-based deep learning model for stock price prediction: A case study on bangladesh stock market," International Journal of Computational Intelligence and Applications, vol. 22, pp. 1-16, 2022. arXiv:2208.08300.

[3] S. Mukherjee, B. Sadhukhan, N. Sarkar, D. Roy, and S. De, "Stock market prediction using deep learning algorithms," CAAI Transactions on Intelligence Technology, vol. 8, pp. 82-94, 2021. doi:10.1049/cit2.12059.

[4] Y. Huang, L. F. Capretz, and D. Ho, "Machine learning for stock prediction based on fundamental analysis," IEEE, 2021. doi:10.1109/SSCI50451.2021.9660134.

[5] S. Lai, M. Wang, S. Zhao, and G. R. Arce, "Predicting high-frequency stock movement with differential transformer neural network," Electronics, vol. 12, pp. 2943, 2023. doi:10.3390/electronics12132943.

[6] K. R. Dahal et al., "A comparative study on effect of news sentiment on stock price prediction with deep learning architecture," PLoS ONE, vol. 18, pp. 1-19, 2023. doi:10.1371/journal.pone.0284695.

[7] M. Paivarinta and L. A. Esteban, "Transformer-based deep learning model for stock return forecasting: Empirical evidence from US markets in 2012–2021," Turun Yliopisto, 2022.

[8] C. Li and G. Qian, "Stock price prediction using a frequency decomposition based GRU transformer neural network," Applied Sciences, vol. 13, pp. 1-18, 2023. doi:10.3390/app13010222.

[9] B. Lim and S. Zohren, "Time series forecasting with deep learning: A survey," Royal Society, vol. 379, 2020. arXiv:2004.13408.

[10] J. Zou et al., "Stock market prediction via deep learning techniques: A survey," Association for Computing Machinery, pp. 1-35, 2023. arXiv:2212.12717.

[11] J. Sen et al., "Automated stock trading framework using reinforcement learning," 2023. doi:10.13140/RG.2.2.16321.12640/1.

[12] S. Elsayed, D. Thyssens, A. Rashed, H. S. Jomaa, and L. Schmidt-Thieme, "Do we really need deep learning models for time series forecasting?," 2021. arXiv:2101.02118.

[13] B. D. Ripley, "Naive time series forecasting methods," R News, vol. 2, pp. 7-10, 2002.

[14] N. Malibari, I. Katib, and R. Mehmood, "Predicting stock closing prices in emerging markets with transformer neural networks: The Saudi stock exchange case," International Journal of Advanced Computer Science and Applications, vol. 12, pp. 876-886, 2021.

[15] D. Othan, Z. H. Kilimci, and M. Uysal, "Financial sentiment analysis for predicting direction of stocks using bidirectional encoder representations from transformers (BERT) and deep learning models," International Conference on Innovative & Intelligent Technologies, vol. 19, pp. 30-34, 2019. doi:10.17758/URUAE8.UL12191013.

[16] K. Alkhatib, H. Khazaleh, H. A. Alkhazaleh, A. R. Alsoud, and L. Abualigah, "A new stock price forecasting method using active deep learning approach," Elsevier, vol. 8, pp. 96, 2023. doi:10.3390/joitmc8020096.

[17] R. Zhang, "LSTM-based stock prediction modeling and analysis," Atlantis Press, vol. 211, pp. 2537-2542, 2022.

[18] Z. Lin, "Comparative study of LSTM and transformer for a-share stock price prediction," in Proceedings of the 2023 2nd International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2023), pp. 72-82, 2023. doi:10.2991/978-94-6463-222-4_7.

[19] T. S. Mian, "Evaluation of stock closing prices using transformer learning," Engineering, Technology & Applied Science Research, vol. 13, pp. 11635-11642, 2023. doi:10.48084/etasr.6017.

[20] Q. Wang and Y. Yuan, "Stock price forecast: Comparison of LSTM, HMM, and transformer," in Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023), pp. 126-136, 2023. doi:10.2991/978-94-6463-198-2_15.

[21] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," 2020. arXiv:1912.09363.

[22] H. Kaeley, Y. Qiao, and N. Bagherzadeh, "Support for stock trend prediction using transformers and sentiment analysis," IISES, 2023. arXiv:2305.14368.

[23] Yahoo!YF, https://finance.yahoo.com (accessed Mar. 28, 2024).

[24] D. Soydaner, "Attention mechanism in neural networks: Where it comes and where it goes," Neural Computing and Applications, vol. 34, pp. 13371-13385, 2022. arXiv:2204.13154.

[25] A. Vaswani et al., "Attention is all you need," in NIPS, 2017. arXiv:1706.03762.

[26] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent deman forecasts," Elsevier, vol. 32, pp. 669-679, 2016. doi:10.1016/j.ijforecast.2015.12.003.

[27] T. H. H. Aldhyani and A. Alzahrani, "Framework for predicting and modeling stock market prices based on deep learning algorithms," Electronics, vol. 11, pp. 3149, 2022. doi:10.3390/electronics11193149.

[28] A. Lopez-Lira and Y. Tang, "Can chatGPT forecast stock price movements? Return predictability and large language models," SSRN, pp. 1-69, 2024. doi:10.2139/ssrn.4412788.

[29] Y. Li, S. Lv, X. Liu, and Q. Zhang, "Incorporating transformers and attention networks for stock movement prediction," Wiley, vol. 2022, pp. 1-10, 2022. doi:10.1155/2022/7739087.

Computational Economics
https://doi.org/10.1007/s10614-024-10604-6

Portfolio Optimization with Prediction-Based Return Using Long Short-Term Memory Neural Networks: Testing on Upward and Downward European Markets

Xavier Martínez-Barbero[1] · **Roberto Cervelló-Royo**[1] · **Javier Ribal**[1]

Accepted: 10 April 2024

Abstract
In recent years, artificial intelligence has helped to improve processes and performance in many different areas: in the field of portfolio optimization, the inputs play a crucial role, and the use of machine learning algorithms can improve the estimation of the inputs to create robust portfolios able to generate returns consistently. This paper combines classical mean–variance optimization and machine learning techniques, concretely long short-term memory neural networks to provide more accurate predicted returns and generate profitable portfolios for 10 holding periods that present different financial contexts. The proposed algorithm is trained and tested with historical EURO STOXX 50® Index data from January 2015 to December 2020, and from January 2021 to June 2022, respectively. Empirical results show that our LSTM neural networks are able to achieve minor predictive errors since the average of the MSE of the 10 holding periods is 0.00047, the average of the MAE is 0.01634, and predict the direction of returns with an average accuracy over the 10 investment periods of 95.8%. Our prediction-based portfolios consistently beat the EURO STOXX 50® Index, achieving superior positive results even during bear markets.

Keywords Portfolio optimization · Return prediction · Asset allocation · Deep learning · Neural networks

✉ Xavier Martínez-Barbero
xamarbar@doctor.upv.es

Roberto Cervelló-Royo
rocerro@esp.upv.es

Javier Ribal
frarisan@upv.es

[1] Faculty of Business Administration and Management, Universitat Politècnica de València, Camino de Vera S/N, 46022 Valencia, Spain

Published online: 01 May 2024

1 Introduction

The prediction of the financial market behavior and optimal budget allocation to specific stocks is one of the main research topics in the financial field. Various factors including financial or monetary policies, exchange rates, inflation, or interest rates, influence financial markets (Hamdani et al., 2020). The complexity and multitude of factors impacting financial markets have made the selection of assets in a portfolio a challenging problem that has been studied by numerous authors.

Since 1952, when Harry Markowitz presented the mean–variance (MV) portfolio selection model (Markowitz, 1952, 1959), different approaches have been applied by researchers to address the topic of portfolio optimization. Markowitz's approach is the cornerstone of the modern portfolio theory (MPT). Subsequently, numerous other authors, including Tobin, who published his work on the risk aversion liquidity preference theory (Tobin, 1958), or Sharpe who extended Markowitz's ideas (Sharpe, 1963), contributed to the development of the field. Since then, many other academics and practitioners have published their studies related to asset pricing (Fabozzi, 1999; Fama, 1996; Sharpe, 1964).

One limitation of Markowitz's model is its sensitivity to the inputs, where the allocation of the weights for portfolio assets varies based on estimated returns, variance, and covariance (Kolm et al., 2014; Michaud & Michaud, 2008). Consequently, inaccurate estimations of expected returns can present a poor performance out of the sample, missing the ability to generalize with unknown data. This underscores the necessity for new methods that can robustly handle the estimation and provide a more stable performance.

Additionally, some studies have shown that optimized portfolios using the mean–variance model have been outperformed by equally weighted portfolios (Jorion, 1985; Korkie & Jobson, 1981). These sub-optimal weights are often attributed to estimation errors in expected returns (Chopra & Ziemba, 1993), further highlighting the need for approaches that can mitigate the estimation of these errors.

There have been advances in the estimation of the parameters such as the Black-Litterman model (Black & Litterman, 1992), Bayes estimator (Jorion, 1986), and robust estimators (DeMiguel & Nogales, 2009). Besides, artificial intelligence has proven its ability to enhance expected return estimation, employing machine learning techniques for improved accuracy in predicting the expected returns (Ban et al., 2018; Chen et al., 2021; Ma et al., 2021) and covariance (De Prado, 2016).

The main objective of this research is to propose an alternative solution to one of the major limitations of portfolio optimization, which is the estimation of input parameters, by applying machine learning algorithms. Specifically, we develop long short-term memory (LSTM) recurrent neural networks (RNN) to predict the expected returns to perform prediction-based portfolio allocations. Therefore, considering the gaps in the current literature, our contribution and the distinctiveness of our paper with respect to the existing literature can be characterized as threefold.

First, by developing sliding window-based LSTM RNN we improve the prediction of future returns. Consequently, more accurate expected returns would

improve the allocation of weights in the construction of optimal portfolios. In our study, we treat each stock's prediction independently as a univariate time series regression problem, given the index comprises companies from various Eurozone countries with differing trading days.

Second, by combining our predicted future returns and classic mean–variance portfolio optimization, we are able to construct optimal portfolios for several short and medium-term investment periods that consistently beat the main stock index of the Eurozone, based on a free-float market cap, and the equally weighted portfolio over the analyzed periods, demonstrating that active portfolio management based on the output of our algorithm achieves superior returns compared to passive management.

Third, this paper focuses on the European market to construct optimal prediction-based portfolios to obtain superior returns. We evaluate our investment strategies over two very different scenarios. On the one hand, we use 2021, a period in which the market shows an upward trend and consistent growth, showing that our model performs better than the benchmark in favorable market conditions. On the other hand, we also use the first half of 2022 to evaluate our model, a period that presents a downward trend, with prices going down due to the war in Ukraine, growing inflation, interest rates increased by central banks, and recession concerns among others. Thus, testing our machine learning model and investment strategies during this period allows us to analyze the performance during bear market conditions. This aspect sets our work apart from other papers, as machine learning algorithms are typically not evaluated under adverse market conditions. This study demonstrates the ability of our LSTM to predict negative growth and create investment strategies that beat the market in this context.

The remainder of this paper is structured as follows. Section 2 reviews previous studies directly related to this paper, summarizing the different methodologies followed and briefly mentioning the results obtained empirically. Section 3 presents theoretical and practical knowledge about LSTM and portfolio optimization and describes the methodology employed, including the data source, the treatment of the data, the LSTM architecture, and states the portfolio optimization problem. Section 4 provides the experimental results. Section 5 explores the significance of the results of the work and draws a conclusion.

2 Literature Review

The prediction of the inputs used in portfolio optimization represents one of the main challenges in the field of portfolio management. The optimal allocation of the assets that make up the portfolio depends on the estimation of the expected return and the variance–covariance matrix. As an estimation of the future may be uncertain, the returns and the variance–covariance matrix could be inaccurately estimated, giving place to poor out-of-sample performance (Basile & Ferrari, 2016). In addition, the sensitivity of portfolio weights to changes in the means of the assets is considerably high (Best & Grauer, 1991).

Many studies use conventional models to predict the price of stocks like autoregressive integrated moving average (ARIMA) (Adebiyi et al., 2014; Mondal et al., 2014) or generalized autoregressive conditional heteroskedasticity (GARCH) (Herwartz, 2017). However, it has been shown that machine learning and deep learning algorithms, such as neural networks, achieve better accuracy than conventional methods in the prediction of time series. Models like ARIMA or GARCH are able to capture linear relations in the data. Nevertheless, considering the inherent assumption of linearity in these models, they fall short in capturing complex non-linear relations, particularly in longer forecasting horizons (Adebiyi et al., 2014; Ghiassi et al., 2005; Rius et al., 1998). Moreover, one of the significant advantages of using artificial intelligence techniques, such as LSTM networks, in stock price prediction is their ability to model the data without the need to assume the normality of the distribution (Hansen & Nelson, 2002).

With the aim of predicting stock prices, machine learning models have been applied by many researchers. Lin et al. (2006) published their dynamic portfolio selection model, where they simulated the dynamic behavior of securities by using a recurrent neural network (RNN), the Elman network. The results are compared to the vector autoregressive (VAR) model, which was outperformed by the RNN. Freitas et al., (2009) developed a method called autoregressive moving reference neural network to optimize a portfolio based on the predicted values of Brazilian stocks, obtaining better results than the MV model and outperforming the IBOVESPA, Brazilian market index. This conclusion is based on several evaluation metrics presented by the authors, which are Mean Error (ME), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and Hit Rates.

Alizadeh et al. (2011) used an adaptive neuro-fuzzy inference system to predict the return. Portfolio optimization based on predicted returns shows a better performance than Markowitz's model, a multiple regression, a neural network, and the Sugeno-Yasukawa method in terms of minimum RMSE. Huang (2012) developed a hybrid methodology that used support vector regression (SVR) and genetic algorithms (GAs) for stock selection to obtain higher returns than the proposed benchmark. Ticknor (2013) proposed a Bayesian regularized artificial neural network to predict the closing price of stocks on the following day using MAPE as a performance metric. The results obtained by the model are comparable to the fusion model of HMM and the ARIMA model proposed by Hassan et al. (2007).

Patel et al. (2015) applied several machine learning techniques to predict two Indian stock market indices. The authors combined SVR with artificial neural networks (ANN), random forest (RF), and SVR itself. The results are compared to the non-hybrid versions of these algorithms, being the hybrid models the ones achieving better performance in terms of MAPE, MAE (Mean Absolute Error), relative RMSE, and MSE (Mean Squared Error). Wang and Wang (2015) predicted financial time series using principal component analysis and a stochastic time-effective neural network (PCA-STNN). The proposed model outperformed a traditional backpropagation neural network (BPNN), principal component analysis combined with BPNN, and the stochastic time-effective neural network. In order to assess the performance of the models, the authors used MAE, RMSE, and MAPE as metrics.

Baek and Kim (2018) developed a model, ModAugNet, based on data augmentation that using LSTM prevented overfitting and predicted the stock market index. ModAugNet outperformed a model that did not consider overfitting prevention in MSE, MAPE, and MAE. Kim and Won (2018) developed a hybrid model combining LSTM with GARCH models, which performed better than existing models such as GARCH, exponential GARCH, or LSTM. They used MAE, MSE, heteroscedasticity-adjusted MAE, and heteroscedasticity-adjusted MSE to compare the performance of the models. In their comparative study, Lee and Yoo (2020) showed that LSTM predictions present a better result than RNN and gated recurrent unit evaluating the predictive ability of the models by using the Hit Ratio. Rezaei et al. (2021) proposed a hybrid deep learning model to predict the stock price and then optimized the portfolio using prediction-based inputs using the Black-Litterman model. The hybrid model, which consists of a combination of complete ensemble empirical mode decomposition, convolutional neural network, and LSTM performed better than the MV portfolio, the Black-Litterman portfolio, and the equally weighted portfolio, in terms of MSE, MAE, and normalized MSE. Collectively, these studies underscore the potential of LSTM-based models as a superior method in financial forecasting.

Ma et al. (2021) combined several machine learning and deep learning models with mean–variance and omega portfolio optimization for daily trading investment in the China Securities Index 100. The results show that the combination of Random Forest (RF) and mean–variance optimization is the one that performed better based on several metrics such as expected return, standard deviation, information ratio, or turnover rate. Also, considering only stock return prediction, RF presented a lower MSE and MAE than the other models.

Du (2022) predicted the return of CSI 300 and S&P 500 with SVM, random forest, and attention-based LSTM, being the last one, the machine learning technique with the best results compared to the others. Predicted returns were evaluated using MSE, MAE, and Hit Ratios, achieving an accuracy superior to 90% for both analyzed markets. This high level of accuracy underscores the effectiveness of attention-based LSTMs in forecasting financial market movements.

All this literature shows the growing importance of artificial intelligence and machine learning algorithms in financial markets, concretely in the prediction of stock prices and returns. Thus, this paper aims to complete the research on the topic by obtaining more accurate price predictions and combining them with mean–variance optimization creating optimal portfolios that generate superior returns in the European market for different investment horizons, including both favorable and unfavorable market conditions.

3 Material and Methods

3.1 Dataset and Data Treatment

This research has exploited historical closing price data of the components of the EURO STOXX 50® Index from January 1, 2015, to June 30, 2022, on a trading

day basis, covering a total of 1903 trading days. The EURO STOXX 50® Index is composed of the 50 largest companies in the Eurozone based on a free-float market cap. The data is obtained from Yahoo! Finance.

As we adopt the technical approach, we believe that despite the importance of the macroeconomic situation, news, and fundamentals, prices fully reflect all the available information and facts that impact financial markets (Mok et al., 2004). In addition, using daily prices instead of weekly or monthly improves the training process of the neural network, as machine learning algorithms' performance increases exponentially with the increase in the amount of data. Also, other studies use daily information, which makes it easier to compare the results of our research (Chen et al., 2021; Du, 2022; Ma et al., 2021; Weng et al., 2018).

In this study, we approach the task as a univariate time series regression problem, where each stock's prediction is handled independently. This approach is particularly relevant because we are dealing with an index comprising companies from various Eurozone countries, which often have differing trading days. Additionally, not all the companies were included in the index on the same date. The missing values are dropped out of the dataset.

The data is normalized by using Min–Max Scaler before training the model. The estimator scales and transforms the values into a given range, in this case between [0, 1] (Pedregosa et al., 2011). The following Eq. (1) presents the mathematical formulation of the Min–Max scaler:

$$x_{t,i_{scaled}} = \frac{x_{t,i} - \min(x_i)}{\max(x_i) - \min(x_i)} \tag{1}$$

where $x_{t,i_{scaled}}$ is the normalized value of $x_{t,i}$, which is the price of the stock i at a due date t. Being $\min(x_i)$ and $\max(x_i)$ the minimum and maximum of x_i, respectively. x_i represents the vector of prices of the stock i for the considered period.

We use a sliding window to generate overlapping sequences of consecutive trading days with a size of 42, corresponding to approximately two months of trading. Thus, the next consecutive price is predicted based on 42 closing stock prices, creating input–output data that will be used to train our long-short term memory. Table 1 illustrates the autoregressive sequence pattern (Jansen, 2020).

Table 1 Sliding window sequence representation: This table illustrates the sliding window sequence used in predictive modeling. It displays 42 consecutive trading days as the input and the subsequent trading day as the output. The table demonstrates how the prices over these 42 days are used to predict the price for the next day

Input	Output
$\langle x1, x2, \ldots, x41, x42\rangle$	$\langle x43\rangle$
$\langle x2, x3, \ldots, x42, x43\rangle$	$\langle x44\rangle$
$\vdots$	$\vdots$
$\langle xT-42, xT-41, \ldots, xT-2, xT-1\rangle$	$\langle xT\rangle$

We select a sliding window of 42, since, after testing several options (displayed in Table 3), it provided better results.

The scaled dataset is split into two datasets. We use data from 2015 to 2020, both included, to train the model and data corresponding to 2021 and the first half of 2022 to test it. 25% of the training dataset is used to validate the model's performance while tuning the hyperparameters. Usually, between 70 and 80% of the training set is used to train, and the remaining 30–20% is used to validate the model. For instance, Ma et al. (2021) used the first four years to train and the following year to validate, representing an 80–20% approach. Using different data to train, validate and test allows us to evaluate the ability of the model to generalize. A summary of the data split is shown in Table 2:

3.2 Methodology

This study's methodology can be divided into two parts. Firstly, the stock price of all the components of the EURO STOXX 50® Index is predicted by using long short-term memory neural networks after creating overlapping sequences employing rolling windows. Secondly, the prediction-based portfolio optimization uses the outputs of the LSTM to find the optimal portfolio with the highest Sharpe ratio and evaluate whether obtained portfolios outperform the benchmarks for the different investment periods considered.

3.2.1 LSTM Prediction

Recurrent neural networks are a type of artificial neural network that can learn patterns by using sequential information or time-series data as input. RNNs keep a hidden state that acts as internal memory, in this way the output depends on the input and the previous hidden state. However, RNNs present some challenges. When errors are backpropagated many time steps through a large sequence, it is possible to experience vanishing or exploding gradients. In addition, RNNs are difficult to train because when gradients vanish, the influence of short-term dependencies is predominant in the weights of gradients, and they could be inefficient to learn long-term dependencies (Bengio et al., 1994; Hochreiter, 1998; Hochreiter et al., 2001).

Long short-term memory is a variant network architecture of RNNs. LSTM arises in 1997 as a solution or alternative method to solve the problems of traditional RNN.

Table 2 Data subsets split: This table illustrate the division of the dataset into various subsets for model training and evaluation

Date	Dataset	Percentage of data points with respect to the date (%)
Jan. 1st, 2015-Dec. 31st, 2020	Training	75
Jan. 1st, 2015-Dec. 31st, 2020	Validation	25
Jan. 1st, 2021-Jun. 30th, 2022	Test	100

LSTM networks are faster and able to solve complex problems that were not solved by preceding recurrent neural networks (Hochreiter & Schmidhuber, 1997). This type of architecture addresses the problem of long-range dependencies and allows for tracking dependencies between the elements of the sequence. LSTM presents an additional internal state called "cell state" which contains one input gate i_t, one forget gate f_t, and one output gate o_t that controls the new information, manages the information that should be voided from the memory of the LSTM, and controls when the information should be processed, respectively (Gers et al., 2002; Jansen, 2020). The following formulas show the calculations associated with each mentioned gate, the cell state, and the hidden state:

$$i_t = \sigma\left(W_i x_t + Y_i h_{t-1} + b_i\right) \tag{2}$$

$$f_t = \sigma\left(W_f x_t + Y_f h_{t-1} + b_f\right) \tag{3}$$

$$o_t = \sigma\left(W_o x_t + Y_o h_{t-1} + b_o\right) \tag{4}$$

$$c_t = c_{t-1} * f_t + \eta_t * i_t \tag{5}$$

$$h_t = tanh\left(c_t\right) * o_t \tag{6}$$

where W_i, W_f, W_o, Y_i, Y_f and Y_o represent weight matrices, b_i, b_f, b_o are bias vectors, c_t is the cell state at time t, η_t corresponds to the input candidate at time t, which is regulated by the input gate, and h_t is the hidden state at time t and it is updated by using hyperbolic tangent activation. In the calculation of the input, forget, and output gate sigmoid activation is used, represented as σ, and computed as $\sigma(x) = \frac{1}{1+e^{-x}}$.

Table 3 Parameters and values considered during the LSTM's training: This table provides an overview of the hyperparameters and the respective values that were explored during the training of the Long Short-Term Memory model

Parameter	Value	Tested values
Window size	42	5, 7, 14, 21, 42, 63
Layers	2	2, 3, 4
Hidden units	40	10, 15, …,80
Dense units	1	
Activation function	Tanh	ReLU
Recurrent activation function	Sigmoid	
Loss function	MSE	
Optimizer	RMSprop	RMSprop, Adam, SGD
Batch size	50	20, 30, 40, 50
Learning rate	$1 * 10^{-3}$	$1*10^{-2}$, $1*10^{-3}$, $1*10^{-4}$
Epochs	500	
Patience	10	10, 20, 30

The sigmoid function acts as a filter of information, allowing information to enter based on the output value that lies between [0,1] (Baek & Kim, 2018).

The hyperparameters that have been considered in the LSTM and the values used to fine-tune the model are shown in Table 3. After training the model and fine-tuning the different values to find the optimal hyperparameters of the LSTM neural network, based on commonly used values in related literature (Jansen, 2020). The topology incorporates two layers, a long short-term memory and a regular densely connected layer containing 40 and 1 unit or nodes, respectively. We defined several topologies for the neural networks. However, the results did not improve significantly, and the complexity of the model was higher. Thus, we decided on the values based on a trade-off between complexity and performance.

As explained above and represented in Eqs. (2) to (6), the activation and recurrent activation functions are hyperbolic tangent (tanh) and sigmoid, respectively. Both functions are relevant to overcome the problem of vanishing gradients. We also explored rectified linear unit (ReLU) as activation function, but we finally use tanh due to considerations related to the available runtime and performance optimization (Chollet, 2015).

MSE is used as a loss function due to its simplicity and for being the most common loss function for regression problems (Hastie et al., 2009). The model will seek to minimize the MSE during the training. After training the model and comparing the results for RMSprop, Adam, and Stochastic Gradient Descent (SGD), we observe that RMSprop provides better results and helps to avoid vanishing and exploding gradients by using a moving average of squared gradients (Hinton et al., 2012).

The LSTM is trained using early stopping to reduce overfitting during a maximum of 500 epochs to allow the model to iterate as much as needed, using patience of 10. This stops the training if the results do not improve continuously during 10 epochs. We do not use dropout or L1, or L2 regularization since overfitting is already prevented using early stopping. Lastly, the learning rate and the batch size are 50 and 0.001, respectively. This was selected based on a trade-off between the model's performance and the training time. In addition, the third column of Table 3 shows the values by parameter that have been tested to find the model that provides better performance without increasing its complexity in excess.

3.2.2 Portfolio Optimization

Classical portfolio optimization is based on the mean–variance model proposed by Markowitz (1952). Since then, most models have used the mean of historical returns to define the expected returns and the covariance. Based on Markowitz's portfolio selection model, we propose to optimize the portfolio using returns calculated with predicted share prices similar to the work of Du (2022), Ma et al. (2021), or Freitas et al. (2009).

3.2.2.1 Expected Risk and Return of a Stock The expected return of each stock is calculated using predicted stock prices. The outputs of the LSTM correspond to the predicted prices of each stock for every day of the year 2021. The following

formula shows how the return is computed. Being $\widehat{r}_t$ the predicted return at time t, $\widehat{P}_t$ the predicted price at time t and $\widehat{P}_{t0}$ the predicted price at time *t0*, which represent the moment of the sell and buy.

$$\widehat{r}_t = \frac{\widehat{P}_t - \widehat{P}_{t0}}{\widehat{P}_{t0}} * 100 \tag{7}$$

The expected risk of one stock is measured by using the standard deviation. It measures the dispersion of the price with respect to its mean and is represented in the following equation:

$$\widehat{V}_t = \sqrt{\frac{\sum_{i=1}^{t}(\widehat{r}_i - \widehat{r})^2}{t-1}} \tag{8}$$

where the $\widehat{r}_i$ is the predicted return, the $\widehat{r}$ is the average of the predicted returns, and t corresponds to the number of days included in the calculation.

3.2.2.2 Expected Risk and Return of a Portfolio The portfolio is made up of N stocks selected by the investor. The expected return is the weighted average of the predicted return of each portfolio. The expected return of the portfolio $\widehat{r}_p$ is shown in the following equation:

$$\widehat{r}_p = \sum_{i=1}^{N} \widehat{r}_i \times W_i \tag{9}$$

where $\widehat{r}_i$ is the predicted return of stock i and the weight is the proportion of the budget allocated to every stock, being $\sum_{i=1}^{N} W_i = 1$. In the current optimization problem, we do consider the possibility of short selling, as some asset types cannot be sold short (Pfaff, 2016). Therefore, for simplicity, we do not allow short selling, and the weights are always positive ($0 \leq W_i \leq 1$). This non-negativity condition is included as a constraint in the optimization of the portfolio.

On the other hand, the risk of the portfolio is measured using the standard deviation, which is the square root of the variance, and it is calculated as follows:

$$\widehat{V}_p = \sqrt{\sum_{i=1}^{N} \sum_{j=1}^{N} W_i W_j \widehat{\gamma}_{ij}} \tag{10}$$

where W_i and W_j represent the weights allocated in the stocks and $\widehat{\gamma}_{ij}$ is the covariance, which serves as a measure of how the stocks vary in relation to each other. This model assumes a fixed covariance structure for each of the holding periods and does not account for time-varying covariances within the same holding period. The calculation is shown in the following equation, where $\widehat{r}_{i,t}$ and $\widehat{r}_{j,t}$ are the predicted return of the given stocks i and j, while $\widehat{r}_i$ and $\widehat{r}_j$ represent their means, respectively and N stands the sample size:

$$\widehat{\gamma_{ij}} = \frac{\sum_{t=1}^{N} \left(\widehat{r}_{i,t} - \bar{\widehat{r}}_i\right) * \left(\widehat{r}_{i,t} - \bar{\widehat{r}}_j\right)}{N - 1} \tag{11}$$

The variance and covariance are calculated considering all the predicted prices throughout the entire investment period considered, from the initial point of purchase to the final point of sale. These measures incorporate the predicted prices as various dates, reflecting the changing value of the assets over the duration of the investment. In contrast, the returns are calculated with the price at the beginning and the end of the investment period, essentially the purchase and sale prices.

3.2.2.3 Portfolio Optimization—Mean–Variance with Forecasting (MVF) Model The portfolio optimization model is built based on the previously defined measures. There are many different approaches, such as minimizing the volatility for a certain level of return or maximizing the return for a given target risk or volatility. In this case, we aim to maximize the Sharpe Ratio (Sharpe, 1994), which reflects the reward to volatility. It is represented in the following formula:

$$SR = \frac{r_p - r_f}{\sigma_p} \tag{12}$$

where r_p is the return of the portfolio, r_f is the Risk-free rate which is assumed to be 0.01 based on the value of the 3-month US Treasury bill according to the Federal Reserve Bank of St. Louis at the end of May 2022, and σ_p is the standard deviation of the portfolio.

The proposed model for portfolio optimization can be formulated as

$$\text{Maximize } \widehat{S} = \frac{\widehat{r}_p - r_f}{\widehat{V}_p} \tag{13}$$

$$\text{Subject to} \sum_{i=0}^{N} \widehat{r}_i \times W_i \geq r_f \tag{14}$$

$$\sum_{i=1}^{N} W_i = 1 \tag{15}$$

$$W_i \geq 0, i = 1, 2, \cdots, N \tag{16}$$

Equation (13) is the objective function that we attempt to maximize. As mentioned before, is the prediction-based Sharpe ratio; Eq. (14) is an inequality constraint function to ensure that the portfolio's returns are higher than the risk-free rate. Otherwise, it would make sense to select a risk-free investment. Equation (15) is an equality constraint function that ensures that all the resources are allocated, whereas Eq. (16) is an inequality constraint function that guarantees non-negative weights in the portfolio.

Maximizing the Sharpe ratio, it is possible to get the optimal portfolio based on risk-adjusted return, showing the expected return in excess of the risk-free rate achieved by the portfolio per unit of risk. To solve the problem is necessary to analyze the set of efficient portfolios, that are the ones that belong to the efficient frontier. These are the portfolios with the highest expected return for each level of risk or the lowest risk for each level of expected return. The selection of one or another portfolio will depend on the risk aversion of the investor.

4 Experimental Results

4.1 Stock Price Prediction

The prediction of stock prices is the cornerstone of the current paper. The predicted price is crucial to obtain the predicted return and volatility of the portfolios. It directly affects the optimal weights and the performance of the optimal portfolio. In the following subsections, we present the evaluation metrics used to assess the robustness of the predictions and a concise interpretation of the results obtained as the output of the proposed models.

4.1.1 Evaluation Metrics

The selected evaluation metrics used to evaluate the performance of the LSTM in forecasting the price of stocks are based, among others, on Freitas et al. (2009), Ma et al. (2021), and Du (2022). Specifically, we used MSE, MAE, and the classification metrics to understand the ability to predict the direction of the return that the model has. These metrics can be defined as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(r_t - \hat{r}_t)^2 \tag{17}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|(r_t - \hat{r}_t)\right| \tag{18}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{19}$$

$$precision = \frac{TP}{TP + FP} \tag{20}$$

$$recall = \frac{TP}{TP + FN} \tag{21}$$

$$f1 = 2 * \frac{precision * recall}{precision + recall} \tag{22}$$

where n is the number of predicted prices or trading days, and r_t and $\hat{r}_t$ are the realized and predicted returns at time t, respectively. *TP* refers to true positive values, *TN* to true negative, *FP* to false positive, and *FN* to false negative.

Although the current study is formulated as a regression problem, calculating classification metrics provides an alternative perspective on model performance. It serves as both a reference for assessing the model's classification-like behavior within the regression context and a potential starting point for future work.

Despite that we calculate the MSE and MAE for every analyzed stock, we use the average MSE and average MAE as global measures of overall prediction

Table 4 Predictive performance for the year 2021 and the first half of 2022: This table offers a comprehensive summary of the performance metrics used to evaluate the predictive capabilities for EURO STOXX 50 stocks across different holding periods

Holding period (days)		20	63	125	191	255
Year 2021						
MSE	Mean	0.000282	0.000315	0.000567	0.000506	0.000607
	Std	0.000356	0.000974	0.001320	0.000804	0.002265
MAE	Mean	0.014035	0.011952	0.016857	0.017717	0.016256
	Std	0.009309	0.013271	0.017002	0.013989	0.018709
Accuracy (%)	Total	92	96	94	96	98
Precision (%)	Up	79	98	98	95	97
	Down	97	86	71	100	100
Recall (%)	Up	92	98	95	100	100
	Down	92	86	83	82	92
F1 (%)	Up	85	98	97	97	99
	Down	95	86	77	90	96
Holding period (days)		25	50	75	100	127
First half of 2022						
MSE	Mean	0.000733	0.000274	0.000296	0.000539	0.000541
	Std	0.000818	0.000349	0.000484	0.000944	0.000632
MAE	Mean	0.023118	0.013162	0.013619	0.018037	0.018692
	Std	0.014228	0.010157	0.010605	0.014749	0.013994
Accuracy (%)	Total	90	100	98	94	100
Precision (%)	Up	79	100	100	85	100
	Down	100	100	97	97	100
Recall (%)	Up	100	100	92	92	100
	Down	84	100	100	95	100
F1 (%)	Up	88	100	96	88	100
	Down	91	100	99	96	100

performance. These measures are compared to other studies (Du, 2022; Ma et al., 2021; Sadaei et al., 2016; Wang et al., 2020; Weng et al., 2018).

4.1.2 Prediction Results

The results obtained by the LSTM are presented in Table 4. They summarize the performance of the recurrent neural network across the 50 components of the EURO STOXX 50® Index by showing the mean and the standard deviation (std) for the two scenarios considered. Table 4 presents the results for 2021, a year with continued growth, and the results for the first half of 2022, during which the market experienced a decline. The results presented correspond to the model that performed best after fine-tuning the hyperparameters for several holding periods. This allows us to evaluate the robustness of the model across different holding days and to be able to consider several investment strategies in terms of the forecast horizon.

Each calculation of the evaluation metrics considers that investors buy on the first day of the year and sell on the day of the selected time horizon. Therefore, the returns that investors will obtain are predicted and analyzed for different holding periods, considering investment strategies from 20 days to 1 year in 2021, which correspond to 1, 3, 6, 9, and 12 months. For the first half of 2022, since the total amount of trading days is 127, we consider 5 investment periods of 25 days, except the last one, which is 27 days. This allows us to have 5 holding periods for each evaluated year.

The results show that the model predicts future returns with minor predictive errors since the average of the MSE of the 10 holding periods is 0.00047, and the average of the MAE is 0.01634.

In 2021, for all the holding periods, the results show small MSE and MAE. Generally, both MSE and MAE increase with time. Also, in order to evaluate how the model predicts the direction of the returns of the different stocks, we employ several classification metrics. For all the analyzed holding horizons, the accuracy is above 90%. Besides, the model can accurately predict both upward and downward movements.

For the period that covers the first half of 2022, the model performs better for the holding periods of 50 and 75 trading days, which present similar results in terms of predictive errors. It shows higher errors for the shortest investment period considered. Similar to the year 2021, the classification metrics show that the model can predict the direction of returns, achieving an accuracy of 100% for two of the analyzed periods.

A comparison between predicted and real returns is shown for every considered holding period in Fig. 1. It is observable that predicted returns are close to real returns and the direction of the returns is well predicted in the vast majority of cases, as can be observed in the mentioned figure, and as it was shown in the classification metrics in Table 4.

Our results in terms of return prediction are comparable to the current literature. We obtain similar or superior results to other studies such as Du (2022), Ma et al. (2021), Sadaei et al. (2016), Wang et al. (2020), or Weng et al. (2018). Generally, our results present predictive errors with smaller mean errors. Nevertheless, it

Fig. 1 Comparison of predicted and real return per holding period: It illustrates a comprehensive comparison of predicted and actual returns of EURO STOXX 50 stocks, evaluating their performance across multiple distinct holding periods

should be considered that we aim to predict returns in different markets, in different contexts, and considering different holding periods. For instance, Ma et al. (2021) focused on the components of the China Securities Index 100, testing the model with data spanning from 2012 to 2015. Meanwhile, Du (2022) encompassed the China Securities Index 300 and the S&P 500, evaluating the model's performance using data covering the period from 2018 to 2020. The consistency of our results with the existing literature, despite considering different time periods and markets,

underscores the ability of LSTM models to adapt and accurately predict across different financial environments.

4.1.3 Prediction Benchmark Validation

In order to validate the superiority and effectiveness of our proposed model, we have conducted an extensive comparative analysis of our LSTM-based stock price prediction approach. We comprehensively evaluate the results of the LSTM by comparing them to other established machine learning models. Our selection of benchmark models encompasses different machine learning techniques, including decision trees, random forest, artificial neural networks, and support vector machine (SVM).

We used the evaluation metrics from Table 4 (detailed in Sect. 4.1.1) for both regression and classification. Our assessment considers the average performance of these metrics across the different holding periods, providing us with a comprehensive overview of the algorithm's performance. Importantly, we conduct this analysis separately for different years, enabling us to gain insights into how our model performs under both bullish and bearish market conditions. This approach allows for a more comprehensive and robust validation of our model. It is essential to emphasize that our primary objective is not to delve into algorithmic details, but rather to validate the superior performance of the LSTM methodology.

The results of the comparison are displayed in Table 5. It provides a comprehensive comparison of the LSTM model with various benchmark models across both the specified scenarios with the explained evaluation metrics for each algorithm. The results show that the LSTM model consistently outperforms the selected benchmark models across both the specified scenarios, namely, the year 2021 and the first

Table 5 Comparative Predictive Performance of LSTM and Benchmark Models for the Year 2021 and the First Half of 2022: This table provides a comprehensive comparison of the LSTM model with various benchmark models across both the specified scenarios

Metric	LSTM	Decision tree	Random forest	SVM	ANN
Year 2021					
MSE	0.000455	0.027642	0.0266164	0.0836882	0.0032534
MAE	0.015363	0.100684	0.0909884	0.179945	0.0410784
Accuracy (%)	95.2	83.2	86.6	61	91.4
Precision (%)	92.1	71.2	75.2	62	90
Recall (%)	92	83.4	89.2	72.6	91
F1 (%)	92	73.8	78.8	55.4	87.8
First half of 2022					
MSE	0.000477	0.015379	0.0143448	0.0424244	0.0045154
MAE	0.017326	0.091939	0.0842554	0.1500896	0.0548612
Accuracy (%)	96.4	64	64.4	48.8	88.8
Precision (%)	95.8	64	60.4	49.4	83.2
Recall (%)	96.3	72.4	67	49.2	91.8
F1 (%)	95.8	58.8	57.6	43.8	84.4

half of 2022. This observation holds true across various evaluation metrics, including Mean Squared Error, Mean Absolute Error, Accuracy, Precision, Recall, and F1-score. These results are consistent with the existent literature (Wang et al., 2020). This demonstrates the LSTM's proficiency in handling sequential data while successfully identifying and capturing long-range dependencies and non-linear patterns.

The LSTM is followed by the non-LSTM artificial neural network which ranks as the second-best performer in predicting outcomes for both scenarios. Subsequently, the random forest model secures the third position, trailed by the decision tree model, with SVM presenting the least effective predictive performance among the algorithms considered.

In conclusion, while the primary objective of this section is not an exhaustive examination of the factors contributing to one algorithm's superior performance over another, the aim is to validate the LSTM model concerning other employed models, encompassing both bullish and bearish market conditions. The results consistently indicate that the LSTM stands as the most proficient algorithm for the given task.

4.2 Prediction-Based Portfolio Optimization

The main reason for predicting the returns is to construct optimized prediction-based portfolios to solve the main drawback of classical portfolio optimization, the sensitivity to estimated inputs. In this section, the experimental results of the portfolio optimization are presented, and the different analyzed scenarios are compared. In addition, the results of the portfolios are benchmarked with the performance of the index. This is crucial, as it will show if the returns of the portfolios outperform the benchmark and, therefore, if it is worth actively managing the portfolio. Otherwise, passive management would be a better option.

4.2.1 Portfolio Construction

We construct portfolios for the same holding periods described in the previous section. The expected return (ER), the volatility (vol), and the Sharpe Ratio (SR) of each portfolio are the metrics employed to evaluate the different portfolios and to compare the results to other studies (Du, 2022; Ma et al., 2021; Sadaei et al., 2016; Wang et al., 2020 or Weng et al., 2018). These results are shown in Table 6.

First, it is observable for the year 2021 that the ER for the combination of the LSTM and the MVF model increases with time, showing that the longer the holding period, the higher the expected return. This is consistent with the increase of the ER of the index since the return of investing in the EURO STOXX 50® Index increases for the analyzed holding periods, as summarized in Table 7. The reason is that financial markets show an upward trend during this period, growing consistently, which is accurately predicted by our model. We do not annualize the expected returns since we calculate the return investors would obtain for that specific investment horizon. Thus, selecting one holding period or another would depend on investor preferences and needs. We do not intend to compare the different holding periods.

Table 6 Portfolio performance for the year 2021 and the first half of 2022: It presents a detailed examination of portfolio performance using predicted data for both years, considering a range of holding periods. It provides a comparative analysis of two strategies: LSTM + MVF and LSTM + 1/N, including their respective performance metrics

Holding period (days)		20	63	125	191	255
Year 2021						
LSTM + MVF	ER (%)	8.4	23.4	30.0	42.8	53.8
	Vol (%)	6.9	7.6	7.5	9.1	9.8
	SR	1.06	2.95	3.85	4.6	5.37
LSTM + 1/N	ER (%)	− 2.7	11.1	17.6	17.7	21.6
	Vol (%)	3.1	5.3	7.19	9.4	11.6
	SR	− 1.21	1.89	2.3	1.78	1.76
Holding period (days)		25	50	75	100	127
First half of 2022						
LSTM + MVF	ER (%)	16.4	14.3	43.2	36.5	23.0
	Vol (%)	15.6	24.6	21.8	20.8	18.4
	SR	0.99	0.54	1.93	1.71	1.19
LSTM + 1/N	ER (%)	− 0.86	− 13.3	− 9.33	− 10.8	− 15.1
	Vol (%)	5.5	11.6	12.9	13.9	15.1
	SR	− 0.35	− 1.24	− 0.81	− 0.86	− 1.07

Table 7 Portfolio and index return for the year 2021 and the first half of 2022. It provides the returns of the portfolios using real returns and compares them to the index proposed as benchmark across various holding periods

Holding period (days)		20	63	125	191	255
Year 2021						
Portfolio Return	(%)	9.02	24.19	28.21	40.6	54.34
Benchmark	(%)	− 0.94	10.15	14.03	13.57	20.81
Holding period (days)		25	50	75	100	127
First half of 2022						
Portfolio Return	(%)	14.85	17.81	41.32	34.91	20.38
Benchmark	(%)	− 4.88	− 13.64	− 10.04	− 15.11	− 20.24

Parallelly, as expected, volatility levels enlarge with the increase in the return, except for the holding period of 63 trading days. This represents 3 months and covers January, February, and March. The higher level of expected volatility could be explained by the previous months of February and March 2020. These months are used to train the LSTM, and from mid-February until the end of March, the EURO STOXX 50® Volatility (VSTOXX®) recorded its highest increase and level since 2008 due to COVID-19.

In the first half of 2022, overall markets decreased, being the market's worst first half in 50 years. As it is observable in Table 7, the EURO STOXX 50® Index represented as the benchmark, which is based on a free-float market cap, presents negative returns for all the analyzed periods, showing a decrease of more than 20% at the end of the sixth month. Despite that, our model achieves positive returns based on predicted data for the five holding periods considered in 2022.

Upon conducting a more comprehensive comparative analysis of our algorithm's performance under distinct market conditions, a notable distinction becomes evident between the two markets under examination: growth and bear markets. This differentiation primarily pertains to the composition of the portfolio with optimized weights. In both scenarios, the portfolios are constructed by selecting a subset of the 50 components belonging to the EURO STOXX 50® Index. These components are selected through the optimization method detailed in the preceding section, with the primary objective being to achieve the highest attainable Sharpe ratio based on predictions generated by the LSTM neural network. Thus, the optimized portfolios hold the components of the EURO STOXX 50® Index but with optimized weights that maximize the Sharpe ratio, exhibiting weightings that deviate from those solely based on free-float market capitalization.

During the period of market growth, the number of stocks with predicted positive returns tends to be higher compared to the first half of 2022, a period marked by substantial market declines. As a result, in the first half of 2022, the number of stocks with predicted positive returns is reduced, leading to a smaller set of stocks comprising the portfolio compared to the previous year.

Furthermore, the portfolios exhibit a superior relative performance in terms of the Sharpe ratio for the year 2021 compared to 2022. The higher Sharpe ratios in 2021 can be attributed to a larger number of companies yielding positive results, as illustrated in Fig. 1. Conversely, in 2022, a reduced number of companies with positive results implies lower diversification, increased risk, and consequently, a smaller Sharpe ratio.

Second, the performance of the LSTM + MVF model is compared to the LSTM + 1/N (see Table 6). The LSTM + 1/N corresponds to the equally weighted portfolio based on the predicted returns. In this case, the weight of each component of the index is 1/50, i.e. 2%. The results show that the LSTM + MVF model outperforms equally weighted portfolios for both years, as it obtains higher ER and SR for all the analyzed holding periods. Even in cases when the equally weighted portfolio shows a negative Sharpe ratio, the combination of our algorithm and the mean–variance model achieves high levels of predicted returns.

Third, the performance of LSTM + MVF portfolios is tested using historical data and compared to the return of the index. For the first comparison, we use the optimized weights of the LSTM + MVF for each holding period with the real return of the 50 components of the index. This way, we see what would have been the real return of the portfolios and we analyze if the proposed investment strategies are profitable or not in reality. By looking at the returns, which are shown in Table 7, it is possible to observe that in the year 2021, having held the investment for 1 month (20 trading days) would have generated a real return of 9.02%. However, if the holding period is higher than 3 months, the real returns oscillate between 24.19 and

54.34%. In 2022, despite the negative results of the index, our portfolios are profitable, and our investment strategies achieve a return of 14.85% in the first month and from 17.81 to 41.32% for longer holding periods.

Moreover, when comparing the relative performance of both analyzed years, the difference between the portfolio returns and the benchmark is more pronounced during the bear market. This difference can be explained by the differences in portfolio composition during growth and bear market periods. During the market growth period, portfolios consist of a larger number of stocks, which is the reason that there are more profitable stocks among the 50 components available in the EURO STOXX 50® Index, which are the ones our model can select. In this scenario, the algorithm has more options to choose from, increasing the likelihood of a greater similarity between the portfolio components and the benchmark. In contrast, during the bear market, the level of diversification is lower due to the lack of stocks with positive returns. Additionally, during this period, stocks with higher market capitalization tend to have lower performance and are therefore less likely to be selected by the algorithm. These factors contribute to a greater difference between the benchmark and the optimized portfolios during the bear market. It is worth noting that the unequal length of both years results from data unavailability during the research period.

Lastly, as it is represented in Fig. 2, the real return obtained by using the weights of optimized portfolios based on predicted returns outperforms the benchmark. In the year 2021, the EURO STOXX 50® Index return is 20.8%. This implies a difference of more than 30% compared to the 54.34% obtained at the end of the year by the portfolio managed by the LSTM + MVF. Even during the first month, in which the EURO STOXX 50® Index went down, the optimized portfolio was able to obtain positive results. In the period that corresponds to 2022, the index went down more than 20% and is consistently outperformed by optimal portfolios. It is important to clarify that the returns presented in this study do not account for transaction costs or other expenses incurred during due to the trading activity. Our research is primarily oriented towards forecasting and planning investment strategies, considering only one-time buy and sell transactions. We do not delve into the creation of automated trading bots, high-frequency strategies, or continuous portfolio rebalancing. However, there are different ways to treat transaction costs. For instance, Ledoit

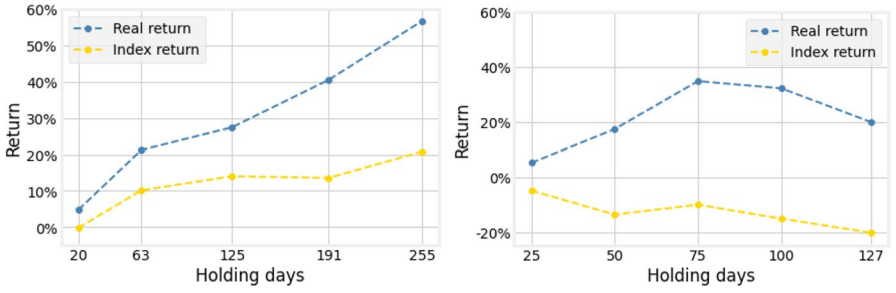

Fig. 2 Comparison of real returns of portfolio and index per holding period for the years 2021 (left) and 2022 (right). It illustrates the performance of the portfolios in terms of returns and compares them to the returns obtained by the EURO STOXX 50® Index

and Wolf (2022) propose a method to integrate transaction costs into the portfolio selection phase in a realistic way, and when they are properly considered, enhances the Sharpe ratio.

If we compare the performance of our created portfolios to the existing literature, our results are in line with it (Du, 2022; Ma et al., 2021; Wang et al., 2020; or Freitas et al., 2009). Our Sharpe ratio is higher or lower depending on the holding period analyzed. Since the research of other authors covers different markets and investment strategies is difficult to compare one to one. However, if we look at the returns, our portfolios usually outperform the expected return achieved by the other studies, being the lower Sharpe ratio driven by volatility.

5 Discussion and Conclusion

5.1 Discussion of key Findings

This paper extends the existing literature by creating profitable investment strategies that clearly and consistently beat the market over two very different scenarios, one in which the market shows consistent growth and another in which is considered a bear market. Our LSTM neural networks can accurately predict the price of European stocks used to create portfolios that achieve superior returns for both mentioned scenarios. Our deep learning algorithms are trained with data from January 2, 2015, to December 30, 2020, and tested by predicting prices for 2021 and the first half of 2022, considering several investment horizons. Our research focuses on the EURO STOXX 50® Index, for which we calculate the return of the 50 components based on predicted prices. Then, we combine calculated returns with mean–variance optimization to generate optimized portfolios that generate returns. By using this approach, we reduce or fully eliminate the subjective human factor that affects the selection of stocks and trading actions.

First, this study presents how we are able to overcome one of the main drawbacks or limitations of portfolio optimization since our rolling-window-based LSTM networks generate predicted prices to calculate returns with minor predictive errors that are used as inputs for the optimization of portfolios. We apply six different metrics that allow us to understand the performance of the model's prediction from a regression problem point of view and consider the prediction a binary classification problem. With these two approaches, we can see how accurate our predicted returns are and whether our model correctly predicts the direction of stock returns. These evaluation metrics fully reflect the performance of the recurrent neural network. The results are compared to the existing literature, showing similar or improved performance. This confirms that our LSTM can address the problem of long-range dependencies and allows us to track dependencies between the elements of the sequence. Adding some economic context, financial markets plummeted in March 2020 due to Covid-19, and the value of the EURO STOXX 50® Index went down from 3.840 on February 14, 2020, to 2.548 on March 20, 2020. Despite that and considering the uncertainty around the global political and economic situation, with many countries applying several measures due to COVID-19. Also, as aforementioned, during 2022

we have experienced the worst market's first half of the past 50 years, and we are experiencing the highest inflation levels since 1981. Despite the adverse economic context, our model is able to overcome this uncertain environment and generate accurate predictions.

Second, we combine our predicted future returns and MV portfolio optimization, defining several holding periods during 2021 and the first half of 2022. Our empirical results show that the created investment strategies consistently beat the EURO STOXX 50® Index, proposed as the benchmark, and the equally weighted portfolio for all the investment horizons considered. We take advantage of the accurate predicted returns to improve the allocation of weights in the construction of optimal portfolios. The portfolios not only beat the benchmarks but also generate positive returns even when the index and overall markets plummet under the conditions mentioned before. In addition, we validate our selected portfolios by calculating the real return by combining historical data and the weights allocated to each stock that makes up the optimal portfolio for each period. The results show that, in reality, our portfolios beat the index for every investment horizon by far.

5.2 Theoretical Implications

This paper enriches the theoretical research on prediction-based portfolio optimization and portfolio management. First, the proposed LSTM neural networks predict future returns with minor predictive errors and overcome the problem of long-range dependencies. Second, using MV optimization, the selection of the portfolios is more precise, due to more accurate predicted returns. This allows us to define several investment strategies that outperform the European market tested using real data for two periods with very different economic and social contexts and are able to consistently generate remarkable returns for investors, which shows the robustness and reliability of our approach.

5.3 Practical Implications

From a practical point of view, this study proposes the application of deep learning techniques to improve the selection of portfolios. For asset and portfolio managers, it can help to make investment decisions, create investment strategies, or complement their current market research and investment processes. For individual investors, it can help to invest without having specific knowledge of the companies and investments. For both, it can reduce the time necessary to study or deep dive into company details, automate their investments and fully isolate emotions that affect the selection of stocks.

5.4 Limitations and Future Work

Despite the results achieved, this research also has limitations. The prediction is based only on historical data; therefore, we do not consider news, economic indicators, technical or fundamental indicators. We adopt a purist technical perspective,

considering that prices fully reflect the available information. However, further research can try to include other inputs to complement the historical data. Also, there can be other ways to estimate the risk, such as the accuracy or the errors of the prediction, which in this case would consider the certainty of the predictions of the model, creating the portfolio based on the trade-off between predicted return and the confidence level of the model in that predicted return. In addition, considering a multivariate LSTM model and comparing it to the univariate approach could be interesting to assess its effectiveness in capturing potential dependencies and correlations among the different stocks. Lastly, mean–variance optimization has been used for many years and portfolios could be optimized using other technics like deep reinforcement learning or quantum-inspired algorithms.

Author Contributions Martínez-Barbero, X: Conceptualization, Methodology, Software, Writing—Original Draft, Writing—Review & Editing. Cervelló-Royo, R: Methodology, Writing—Original Draft, Writing—Review & Editing, Supervision. Ribal, J: Conceptualization, Writing—Original Draft, Writing—Review & Editing, Supervision.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*. https://doi.org/10.1155/2014/614342

Alizadeh, M., Rada, R., Jolai, F., & Fotoohi, E. (2011). An adaptive neuro-fuzzy system for stock portfolio analysis. *International Journal of Intelligent Systems, 26*(2), 99–114. https://doi.org/10.1002/int.20456

Baek, Y., & Kim, H. (2018). ModAugNet: A new forecasting framework for stock market index value with an overfitting prevention LSTM module and a prediction LSTM module. *Expert Systems with Applications, 113*, 457–480. https://doi.org/10.1016/j.eswa.2018.07.019

Ban, G.-Y., El Karoui, N., & Lim, A. E. (2018). Machine learning and portfolio optimization. *Management Science, 64*(3), 1136–1154. https://doi.org/10.1287/mnsc.2016.2644

Basile, I., & Ferrari, P. (2016). *Asset management and institutional investors*. Springer. https://doi.org/10.1007/978-3-319-32796-9

Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks, 5*(2), 157–166. https://doi.org/10.1109/72.279181

Best, M. J., & Grauer, R. R. (1991). On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results. *Review of Financial Studies, 4*(2), 315–342. https://doi.org/10.1093/rfs/4.2.315

Black, F., & Litterman, R. (1992). Global portfolio optimization. *Financial Analysts' Journal, 48*(5), 28–43. https://doi.org/10.2469/faj.v48.n5.28

Chen, W., Zhang, H., Mehlawat, M. K., & Jia, L. (2021). Mean–variance portfolio optimization using machine learning-based stock price prediction. *Applied Soft Computing, 100*, 106943. https://doi.org/10.1016/j.asoc.2020.106943

Chollet, F (2015). Keras. https://keras.io/api/layers/recurrent_layers/lstm/

Chopra, V. K., & Ziemba, W. T. (1993). The effect of errors in means, variances, and covariances on optimal portfolio choice. *Journal of Portfolio Management, 19*(2), 6–12. https://doi.org/10.1142/9789814417358_0021

De Prado, M. L. (2016). Building diversified portfolios that outperform out of sample. *Journal of Portfolio Management, 42*(4), 59–69. https://doi.org/10.3905/jpm.2016.42.4.059

DeMiguel, V., & Nogales, F. J. (2009). Portfolio selection with robust estimation. *Operations Research, 57*(3), 560–577. https://doi.org/10.1287/opre.1080.0566

Du, J. (2022). Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks. *Expert Systems with Applications, 201*, 117005. https://doi.org/10.1016/j.eswa.2022.117005

Fabozzi, F. J. (1999). *Investment management.* New Jersey: Prentice Hall (2nd ed.).

Fama, E. F. (1996). Multifactor portfolio efficiency and multifactor asset pricing. *Journal of Financial and Quantitative Analysis, 31*(4), 441–465. https://doi.org/10.2307/2331355

Freitas, F. D., De Souza, A. F., & De Almeida, A. R. (2009). Prediction-based portfolio optimization model using neural networks. *Neurocomputing, 72*(10–12), 2155–2170. https://doi.org/10.1016/j.neucom.2008.08.019

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research, 3*, 115–143.

Ghiassi, M., Saidane, H., & Zimbra, D. K. (2005). A dynamic artificial neural network model for forecasting time series events. *International Journal of Forecasting, 21*(2), 341–362. https://doi.org/10.1016/j.ijforecast.2004.10.008

Hamdani, Z., Dif, A., Zairi, B., Benziane, A., Hamdani, M. (2020). Factors Affecting the Behavior of Financial Markets in the Light of Financial Risks. *Advances in Human Factors, Business Management and Leadership.* AHFE 2020. *Advances in Intelligent Systems and Computing*, vol 1209. Springer, Cham. https://doi.org/10.1007/978-3-030-50791-6_24

Hansen, J. V., & Nelson, R. D. (2002). Data mining of time series using stacked generalizers. *Neurocomputing, 43*(1–4), 173–184. https://doi.org/10.1016/S0925-2312(00)00364-7

Hassan, R., Nath, B., & Kirley, M. (2007). A fusion model of HMM, ANN, and GA for stock market forecasting. *Expert Systems with Applications, 33*(1), 171–180. https://doi.org/10.1016/j.eswa.2006.04.007

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1–758). New York: Springer.

Herwartz, H. (2017). Stock return prediction under GARCH—An empirical assessment. *International Journal of Forecasting, 33*(3), 569–580. https://doi.org/10.1016/j.ijforecast.2017.01.002

Hinton, G., Srivastava, N., & Swersky, K. (2012). *Neural networks for machine learning lecture 6a overview of mini-batch gradient descent.*

Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. *A field guide to dynamical recurrent neural networks*. https://www.researchgate.net/publication/2839938

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 6*(02), 107–116. https://doi.org/10.1142/S0218488598000094

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang, C. F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing, 12*(2), 807–818. https://doi.org/10.1016/j.asoc.2011.10.009

Jansen, S. (2020). RNNs for Multivariate Time Series and Sentiment Analysis. In S. Jansen, *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python* (pp. 591–624). Packt Publishing Ltd.

Jorion, P. (1985). International portfolio diversification with estimation risk. *The Journal of Business, 58*(3), 259–278.

Jorion, P. (1986). Bayes-Stein Estimation for Portfolio Analysis. *Journal of Financial and Quantitative Analysis, 21*(3), 279–292. https://doi.org/10.2307/2331042

Kim, H. Y., & Won, C. H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications, 103*, 25–37. https://doi.org/10.1016/j.eswa.2018.03.002

Kolm, P. N., Tütüncü, R., & Fabozzi, F. (2014). 60 Years of portfolio optimization: Practical challenges and current trends. *European Journal of Operational Research, 234*(2), 356–371. https://doi.org/10.1016/j.ejor.2013.10.060

Korkie, R. M., & Jobson, J. (1981). Putting Markowitz theory to work. *The Journal of Portfolio Management, 7*(4), 70–74. https://doi.org/10.3905/jpm.1981.408816

Ledoit, O., & Wolf, M. (2022). Markowitz portfolios under transaction costs. *Working paper series/Department of Economics*, (420). https://ideas.repec.org/p/zur/econwp/420.html

Lee, S. I., & Yoo, S. J. (2020). Threshold-based portfolio: The role of the threshold and its applications. *The Journal of Supercomputing, 76*(10), 8040–8057. https://doi.org/10.1007/s11227-018-2577-1

Lin, C.-M., Huang, J. J., Gen, M., & Tzeng, G.-H. (2006). Recurrent neural network for dynamic portfolio selection. *Applied Mathematics and Computation, 175*(2), 1139–1146. https://doi.org/10.1016/j.amc.2005.08.031

Ma, Y., Han, R., & Wang, W. (2021). Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications, 165*, 113973. https://doi.org/10.1016/j.eswa.2020.113973

Markowitz, H. M. (1959). *Portfolio Selection: Efficient Diversification of Investments*. Yale University Press.

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance, 7*(1), 77–91. https://doi.org/10.2307/2975974

Michaud, R. O., & Michaud, R. O. (2008). *Efficient asset management: A practical guide to stock portfolio optimization and asset allocation*. Oxford University Press.

Mok, P., Lam, K., & Ng, H. (2004). An ica design of intraday stock prediction models with automatic variable selection. *2004 IEEE international joint conference on Neural Networks* (pp. 2135–2140). https://doi.org/10.1109/IJCNN.2004.1380947

Mondal, P., Shit, L., & Goswam, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*. https://doi.org/10.5121/ijcsea.2014.4202

Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock market index using fusion of machine learning techniques. *Expert Systems with Applications, 42*(4), 2162–2172. https://doi.org/10.1016/j.eswa.2014.10.031

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & Vanderplas, J. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pfaff, B. (2016). *Financial risk modelling and portfolio optimization with R*. John Wiley & Sons.

Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. *Expert Systems with Applications, 169*, 114332. https://doi.org/10.1016/j.eswa.2020.114332

Rius, A., Ruisanchez, I., Callao, M. P., & Rius, F. X. (1998). Reliability of analytical systems: Use of control charts, time series models and recurrent neural networks (RNN). *Chemometrics and Intelligent Laboratory Systems, 40*(1), 1–18. https://doi.org/10.1016/S0169-7439(97)00085-3

Sadaei, H. J., Enayatifar, R., Lee, M. H., & Mahmud, M. (2016). A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting. *Applied Soft Computing, 40*, 132–149. https://doi.org/10.1016/j.asoc.2015.11.026

Sharpe, W. F. (1994). The sharpe ratio. *The Journal of Portfolio Management, 21*(1), 49–58. https://doi.org/10.3905/jpm.1994.409501

Sharpe, W. F. (1963). A simplified model for portfolio analysis. *Management Science, 9*(2), 277–293. https://doi.org/10.1287/mnsc.9.2.277

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance, 19*(3), 425–442. https://doi.org/10.1111/j.1540-6261.1964.tb02865.x

Ticknor, J. L. (2013). A Bayesian regularized artificial neural network for stock market forecasting. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2013.04.013

Tobin, J. (1958). Liquidity preference as behavior towards risk. *The Review of Economic Studies, 25*(2), 65–86. https://doi.org/10.2307/2296205

Wang, J., & Wang, J. (2015). Forecasting stock market indexes using principle component analysis and stochastic time effective neural networks. *Neurocomputing, 156*, 68–78. https://doi.org/10.1016/j.neucom.2014.12.084

Wang, W., Li, W., Zhang, N., & Liu, K. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications, 143*, 113042. https://doi.org/10.1016/j.eswa.2019.113042

Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications, 112*, 258–273. https://doi.org/10.1016/j.eswa.2018.06.016