engineering proceedings

MDPI

Proceeding Paper

Financial Time Series Models—Comprehensive Review of Deep Learning Approaches and Practical Recommendations †

Mateusz Buczyński [1,2,*], Marcin Chlebus [1], Katarzyna Kopczewska [1] and Marcin Zajenkowski [3]

[1] Faculty of Economic Sciences, University of Warsaw, 00-241 Warsaw, Poland; mchlebus@wne.uw.edu.pl (M.C.); kkopczewska@wne.uw.edu.pl (K.K.)
[2] Interdisciplinary Doctoral School, University of Warsaw, 00-312 Warsaw, Poland
[3] Faculty of Psychology, University of Warsaw, 00-183 Warsaw, Poland; zajenkowski@psych.uw.edu.pl
* Correspondence: mp.buczynski2@uw.edu.pl

† Presented at the 9th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 12–14 July 2023.

Abstract: There have been numerous advances in financial time series forecasting in recent years. Most of them use deep learning techniques. We identified 15 outstanding papers that have been published in the last seven years and have tried to prove the superiority of their approach to forecasting one-dimensional financial time series using deep learning techniques. In order to objectively compare these approaches, we analysed the proposed statistical models and then reviewed and reproduced them. The models were trained to predict, one day in advance, the value of 29 indices and the stock and commodity prices over five different time periods (from 2007 to 2022), with 4 in-sample years and 1 out-of-sample year. Our findings indicated that, first of all, most of these approaches do not beat the naive approach, and only some barely beat it. Most of the researchers did not provide enough data necessary to fully replicate the approach, not to mention the codes. We provide a set of practical recommendations of when to use which models based on the data sample that we provide.

Keywords: financial forecast; deep learning; reproducibility; forecast comparison

check for updates

Citation: Buczyński, M.; Chlebus, M.; Kopczewska, K.; Zajenkowski, M. Financial Time Series Models—Comprehensive Review of Deep Learning Approaches and Practical Recommendations. *Eng. Proc.* **2023**, *39*, 79. https://doi.org/10.3390/engproc2023039079

Academic Editors: Ignacio Rojas, Hector Pomares, Luis Javier Herrera, Fernando Rojas and Olga Valenzuela

Published: 12 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Many researchers have been struggling for decades to understand how the markets behave [1–3]. Some argue that the markets are unpredictable due to the Efficient Market Hypothesis (EMH), stating that, in the short term, financial time series follow a random walk. In contrast, there is a large number of behavioural economists that do not agree with such a statement, believing that investors do not always behave rationally [4,5]. They suggest that the market "can be beaten", because cognitive biases, such as overconfidence and herd behaviour or risk aversion, exist. One is certain and empirically confirmed: investors are winning on the market—mostly because they are ahead of their "brothers in arms".

Generally, there are two main approaches that are used to predict the financial markets: technical and fundamental analysis. Technical analysis approaches focus widely on building the predictions based on the past movements or changes of the stock market [6,7]. On the other hand, fundamental analysis considers the information about the economic status of the company underlying the asset, news, social media, financial reports, etc. Lately, the most emphasis is put on employing machine or deep learning methods to combine these tasks, due to their ability to find and quantify nonlinear correlations very easily [8–12], but researchers are still struggling to provide an objective way to compare the results. There has been a massive progress in artificial intelligence approaches implemented in the financial area, mainly portfolio optimisation, time series prediction, agent-based modelling, etc. A large number of scientists also agree that the origin of successful prediction lies not only in the data related to the predicted object, but also in finding additional data sources [13–15].

Systematic reviews [16,17] show that there have been more than 125 new approaches to time series prediction in the past few years. Many researchers claim to provide better and better-performing models; however, no consensus exists on the best approach yet. The availability of numerous model options in the market without a clear indication of their costs can result in a dilemma known as "choice overload" [18]. This can lead to a situation where one may end up making no choice at all.

When it comes to financial time series, there are also many different areas that can be covered by deep learning. We can see different feature sets being used in the model, either univariate modelling or enriching the data with additional supporting data sources. There is also emerging work using text mining, sentiment analysis, or social media analysis in feature sets. The target variable can also change: it can predict stock prices, indices, commodities, or cryptocurrencies. Some researchers are also looking at volatility and trend forecasting. As far as models are concerned, the horizon is even broader: from simple neural networks, to long-term memory (LSTM) architectures, to sophisticated state-of-the-art approaches such as graph networks or generative adversarial networks. Finally, the prediction horizon is also a point of contention: Are predictions longer than one time step forward of good quality, and what should it be—a regression task or a classification task?

This abundance of different possibilities and options generates a large grid of approaches that cannot be compared with each other solely on the basis of the article provided. There are three main problems when it comes to comparing approaches to predicting financial time series in a deep learning setting:

- Different data, timespans, and metrics used in every experiment;
- Lack of publicly available codes supporting the experiment's execution;
- Lack of a detailed architecture and hyperparameters that are necessary for the experiment's reproduction.

The first problem stems from the lack of a single stock framework, indexes, or any other data samples to objectively test the effectiveness of the models. The usual duo is the S&P 500 and the SSE Composite, according to [17], accounting for 80% of the papers they reviewed. However, a large subset of researchers use single stock quotes or commodity prices. When models are supplemented with additional data, e.g., enriched with text-mining techniques, such datasets are not publicly available (only 10% of the reviewed papers in [17]). In time series problems, it is not only about the data, but also about the time sample to be used: different results will be obtained for models trained in 2019 and 2020, or a different training sample or lookback time horizon: different results will be obtained for models trained in 1 year and 5 years. In terms of metrics, there is some minor consensus there: for regression, the common metric is the mean absolute error; for classification, it is the accuracy. However, this consensus does not mean that every researcher provides a grid of "must have" metrics, but rather selects a few of the most-common ones.

The second problem stems from the reluctance of researchers to publish the code they used to train the models. Only three of the papers reviewed in this thesis were taken from publicly available sources such as GitHub. The lack of code reduces the usefulness of the work, as the cost of selection is increased by the time it takes to implement. Another disadvantage is the discrepancy between the implemented solution and the one presented in the original work (e.g., different versions of the base packages).

A final problem is the poor description of the approach along with the hyperparameters that are used to train the model. Typically, in time series deep learning, we can expect the following hyperparameters (depending on the approach implemented): number of training epochs, learning rate, optimiser type, batch size, and number of backward steps (number of time series lags). However, in the works mentioned by [16,17], there are huge gaps in the description of the training approach. Most of the approaches lack a concrete specification of the architecture used (number of neurons, number of layers, activation functions, etc.) or lack parameters. Researchers usually stop at explaining that the architecture used is LSTM or NN. Additionally, only one paper mentions the random seed value, which is also necessary to fully reproduce the model weights.

To overcome these problems, we decided to carry out an extensive practical reproduction of fifteen papers that are listed in [16,17]. We rebuilt each approach that was reported in a duplicate article, taking into account the hyperparameters that were reported. Where important parameters were missing (such as the number of neurons, the optimiser, the learning rate, or the number of training epochs), we supplemented with the average of the remaining, non-missing articles. We compared these models with simple statistical approaches—naive forecasting, ARIMA, and exponential smoothing. The result was 18 forecasts over five different time periods, resulting in 90 forecasts, run for 29 different types of financial data, including indices, equities, and commodities. We compared models using the mean absolute percentage error (MAPE), mean-squared error (MSE), mean absolute error (MAE), and mean absolute error compared in the first time step only. We propose a data, time, and model framework to run when performing time series problems with deep learning.

2. Methods

We reviewed 15 different deep learning models that have been mentioned in recent literature reviews on financial time series prediction. We focused on selecting the broadest possible sample of different deep learning models. In this section, we describe them briefly. To limit the size of the article, we refer the reader to the original papers for more details on the individual models:

1. **A hybrid attention-based EMD-LSTM model** [19]:
   The paper proposes a two-stage model for time series prediction, combining empirical mode decomposition (EMD) and attention-based long short-term memory (LSTM-ATTE). EMD was used to decompose the time series into a few inherent mode functions (IMFs), which were then taken as the input to LSTM-ATTE for prediction. They used the SSE Composite index to run the predictions. The attention mechanism was used to extract the input features of the IMF and improve the accuracy of the prediction. Researchers have evaluated the model's predictive quality using linear regression analysis of the stock market index and compared it to other models, showing better prediction accuracy.
2. **Empirical. mode decomposition factorisation neural network (EMD2FNN) model** [20]:
   A simpler approach proposed by [20], includes feeding the IMFs of some time series into a factorisation neural network, concatenating all the IMFs into a single vector. The data used for the experiment were: the SSE Composite, NASDAQ, and S&P 500. The authors performed a thorough comparison between the proposed method and other neural network models, comparing the mean absolute error (MAE) and root-mean-squared error (RMSE).
3. **Neural network ensemble** [21]:
   The paper describes a deep neural network ensemble that aims to predict the SSE Composite and SZSE (Shenzhen) Component. The model consists of a set of neural networks that were trained using open, high, low, close (OHLC) data. Every neural network takes the last few days of such data, flattened to a vector form. Later, bagging is used to combine these networks and reduce the generalisation error.
4. **Wavelet denoising long short-term memory model** [22]:
   The proposed model in this paper is a combination of real-time wavelet denoising and the LSTM neural network. The wavelet denoising was used to separate signals from noise in the stock data and was then taken as the input to the LSTM model. The authors conducted an experiment on several indexes, including the SSE, SZSE, and NIKKEI, using the mean absolute percentage error (MAPE) as a metric.
5. **Dual-stage attention-based recurrent neural network** [23]:
   This paper proposes a two-stage attention-based recurrent neural network (DA-RNN) model for time series prediction. The DA-RNN model uses an input attention mechanism in the first stage to extract the relevant driving series at each time step based on the previous hidden state of the encoder. In the second stage, the temporal attention

mechanism is used to select the relevant hidden encoder states at all time steps. The experiment was conducted on the SML 2010 and NASDAQ datasets and showed that the model outperformed state-of-the-art time series prediction methods. The metrics used were the MAE, MAPE, and RMSE.

6. **Bidirectional LSTM** [24]:
   This paper compared the performance of the bidirectional LSTM (BiLSTM) and unidirectional LSTM models. BiLSTM is able to traverse the input data twice (left to right and right to left) and, thus, has additional training capabilities. The study showed that BiLSTM-based modelling offers better predictions than regular LSTM-based models and outperformed the ARIMA and LSTM models. However, BiLSTM models reach equilibrium much slower than LSTM-based models. The experiment was carried out on several indices and stocks, including the Nikkei and NASDAQ, as well as the daily IBM share price and compared using RMSE.
7. **Multi-scale. recurrent convolutional neural network** [25]:
   The proposed method is a multi-scale temporal dependent recurrent convolutional neural network (MSTD-RCNN). The method utilises convolutional units to extract features on different time scales (daily, weekly, monthly) and a recurrent neural network (RNN) to capture the temporal dependency (TD) and complementarity across different scales of financial time series. The proposed method was evaluated on three financial time series datasets from the Chinese stock market and achieved state-of-the-art performance in trend classification and simulated trading compared to other baseline models.
8. **Time-weighted. LSTM** [26]:
   This paper proposes a novel approach to predicting stock market trends by adding a time attribute to stock market data to improve prediction accuracy. The approach involves assigning weights to the data according to their temporal proximity and using formal stock market trend definitions. The approach also uses a custom long short-term memory (LSTM) network to discover temporal relationships in the data. The results showed that the proposed approach outperformed other models and can be generalised to other stock indices, achieving 83.91% accuracy in a test with the CSI 300 index.
9. **ModAugNet** [27]:
   The paper proposes a data augmentation approach for stock market index forecasting through the ModAugNet framework, which consists of a fitting-prevention LSTM module and a prediction LSTM module. The prediction module is a simple LSTM network that is fit based only on the historical data on the index realised prices. The prevention module builds on that by adding a set of regressors that are other indexes, highly correlated with the predicted one. Using the MSE, MAE, and MAPE on the S&P500 and KOSPI200, the authors proved the validity of their solution.
10. **State frequency memory (SFM)** [28]:
    The state frequency memory (SFM) model is the twin of the LSTM model. The SFM model was inspired by the discrete Fourier transform (DFT) and was designed to capture multi-frequency trading patterns from past market data to make long- and short-term predictions over time. The model decomposes the latent states of memory cells into multiple frequency components, where each component models a specific frequency of the latent trading pattern underlying stock price fluctuations. The model then predicts future share prices by combining these frequency components. The authors tested their solution of 50 different stocks in 10 industries using the MSE.
11. **Convolutional neural-network-enhanced support vector machine** [29]:
    The proposed model in this text is a convolutional neural network (CNN), which is supposed to discover features in the data, which are later passed into the support vector machine (SVM) model. The text then discusses the influence of the model parameters on the prediction results. The model was evaluated empirically on the

Hong Kong Hang Seng Index using the RMSE, and the results showed that both models are feasible and effective.

12. **Generative adversarial network** [30]:
    The generative adversarial network (GAN) in this paper consists of two main components: a discriminator and a generator. The discriminator was designed using a simple feed-forward neural network and is responsible for distinguishing real stock market data from generated data. The generator, on the other hand, was built using an LSTM and is responsible for generating data with the same distribution as the actual stock market data. The model was trained on daily data from the S&P500 index and several other stocks for a wide range of trading days. The LSTM generator learns the distribution of the stock data and generates new data, which are then fed to the MLP discriminator. The discriminator learns to distinguish between the actual stock data and the data generated by the generator. The authors tested their model on several time series, including the S&P 500 and stocks such as IBM or MSFT.
13. **Long short-term memory and gated recurrent unit models** [31]:
    The paper proposes a hybrid model that combines the long short-term memory (LSTM) and gated recurrent unit (GRU) networks. The authors used the S&P 500 historical time series data and evaluated the model using metrics such as the MSE and MAPE on the S&P500.
14. **CNN and bi-directional LSTM model** [32]:
    The paper proposes a model combining multiple pipelines of convolutional neural network (CNN) and bidirectional long short-term memory (LSTM) units. The model improved the prediction performance by 9% compared to a single pipelined deep learning model and by more than six-times compared to a support vector machine regressor model on the S&P 500. The paper also illustrates the improvement in the prediction accuracy while minimising overfitting by presenting several variants of multi- and single-pipelined deep learning models based on different CNN kernel sizes and number of bidirectional LSTM units.
15. **Tim convolution (TC) LSTM model** [33]:
    The authors of this paper propose time convolution long short-term memory (TC-LSTM), employing convolutional neural networks (CNNs) to capture long-term fluctuation features in the stock prices and combining this with LSTM. This combination allows the model to capture both the long-term dependencies of stock prices, as well as the overall change pattern. The authors compared the performance of their TC-LSTM model to three baseline models on 50 stocks from the SSE 50, as well as the index itself. They showed that their model outperformed the others in terms of the mean-squared error.

The proposed architectures were build from scratch in pytorch [34], based on the explanation provided in the article itself. In addition to these models, to provide a more thorough comparison, we also utilised the ARIMA model (tuned, best parameters on the training sample), the naive approach (prediction as: $Y_t = Y_{t-1}$), and exponential smoothing.

The hyperparameters derived based on the text or the publicly available codes are presented in Table A1. All missing parameters were filled in with either the mean of other parameters or the mode.

3. Data and Methodology

We provide a more comprehensive background of the developed models we propose to broaden the range of the time series on which the model is tested. The data used in this study were taken from the following financial types: indexes, currency pairs, stocks, cryptocurrencies, and commodities. The purpose of including a comprehensive range of financial types was to provide a comprehensive comparison of the models' performance.

The following time series were included in this study for analysis:

- Indexes: WIG20 (PL), S&P 500 (US), NASDAQ (US), Dow Jones Industrial (US), FTSE 250 (UK), Nikkei 225 (JP), DJI (USA), KOSPI 50 (KR), SSE Composite (CN), DAX 40 (DE), CAC40 (FR);
- Currency pairs: EURPLN, PLNGBP, USDPLN, EURUSD, EURGBP, USDGBP, CHFGBP, CHFUSD, EURCHF, PLNCHF;
- Stocks: AAPL, META, AMZN, TSLA, GOOG, NFLX;
- Cryptocurrencies: BTCUSD;
- Commodities: XAUUSD.

For each of these time series, we identified five periods in which we made predictions:

- 2016–2020;
- 2013–2017;
- 2007–2011;
- 2009–2013;
- 2018–2022.

Each period consisted of 4 years of training and 1 year of day-ahead predictions (ca. 250 testing time steps) without re-training the model. The periods differed significantly between each other due to the different levels of variability between the training and test trials.

The data were preprocessed by performing normalisation on the input features using the MinMaxScaler function from the Scikit-learn library [35]. The normalised data were then split into training and test sets with a ratio of 4 years:1 year, respectively. The model was then trained using only the training sample, and predictions were made for every time step in the testing sample.

The evaluation metrics were used to compare the performance of each model across the different financial types and time series. The best-performing model was selected based on the lowest values of:

- $MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2$;
- $RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(Y_t - \hat{Y}_t)^2}$;
- $MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{Y_t - \hat{Y}_t}{Y_t}\right| \times 100\%$.

Y_t is the actual value at time t; $\hat{Y}_t$ is the predicted value at time t; n is the total number of time periods.

Finally, the results were analysed to identify patterns and insights that could help improve the accuracy of predictions in future studies. We also provide the MAPEs at the first time step of testing (i.e., calculated for $t = 1$) to provide a quality metric for the trained model with the full set of information, i.e., to provide a metric that would allow the quality of the model to be calculated in the short term.

4. Results

The results are presented in Table 1. To be concise, we only report the MAPE in this paper (for the MAE, MSE, or detailed predictions, please contact the authors directly). In bold, we can notice the best (lowest) MAPE for every financial time series.

We can see that the best model was the naive approach. This was mainly due to the fact that the quality of models tends to deteriorate if they are not retrained after a certain period of time; however, we wanted to keep the reproduction of the models as close to the original as possible. Furthermore, the researchers in their original work also did not retrain the model in the test sample (or mention doing so in any other way), nor did they compare to statistical approaches.

When we leave out the statistical approaches, we can see a few approaches that stand out (5, 6, 7, 9, 12, and 13). These models generally have MAPEs lower than five percent. What these models have in common is either simple LSTM/RNN architectures or sophisticated CNN operations, which increase the range of features. The best model that achieved an average MAPE of 1.79% was the multi-scale recurrent convolutional neural network (Model 7). We believe that the surplus in prediction quality came from the additional operations performed on the data (multi-scale CNN), which improved the information processing. The second-best model was ModAugNet (Model 9), which achieved a MAPE of 2.07%. This model relies heavily on the additional data sources that are provided in training for a given time series. On the other hand, the worst model was Model 11 using SVM after preprocessing the data from the SVM—52.24% MAPE.

From another perspective, the worst-performing financial time series was BTCUSD, followed by TSLA, AAPL, and NFLX. These all achieved high returns over the time periods studied, so what we believe is the reason for the deterioration was the inability of all models to correctly identify and predict rapid price increases. We also observed that indices have a roughly similar MAPE (5–6%), as do currency pairs (1–2%). Stocks, on the other hand, have the highest MAPE of all financial time series (>10%).

Since we can clearly see that the models performed worse due to the lack of hypertuning, we propose to run this procedure for each model training. However, such an experiment will be computationally exhaustive (this experiment already consisted of 2610 model trainings), so some restrictions should be introduced. Furthermore, the testing procedure was detached from how these models are used in reality. Time series models should be trained daily to provide the best-possible fit based on the set of information available at the time of prediction. In this case, the information set becomes smaller and smaller with each prediction.

As a proof of this statement, we provide in Table 2 the MAPE results calculated for the first testing time step. This metric allowed us to confirm that, in the short term, these models are correct and better than the naive approach for each stock. Only for five series was the statistical approach (exponential smoothing) better. Model 7 was still the best in 7 of the remaining 24 cases. Several other models came in first (Model 1 and Model 13) proving that the performance of the models deteriorates over time.

Table 1. An average of the mean absolute percentage error for every model and every financial time series calculated over five different timespans.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**ARIMA**	**naive**	**ExpSmooth**
AAPL	10.33	19.98	63.42	10.77	5.55	5.53	2.48	7.12	1.69	7.32	56.94	4.47	4.32	9.74	8.36	1.7	**1.42**	1.57
AMZN	14.57	19.21	64.15	12.36	6.78	7.01	2.94	12.11	2.82	6.08	31.67	4.98	3.83	14.8	14.73	1.88	**1.61**	1.62
BTCUSD	19.74	41.24	72.07	22.48	17.23	25.6	5.38	24.5	11.3	19.8	1110.76	12.46	17.05	36.32	32.05	3.81	**3.17**	3.37
CHFGBP	3.03	5.16	18.39	1.73	0.99	1.02	0.7	2.26	0.58	4.48	6.44	0.67	0.91	2.82	1.05	0.49	**0.46**	0.46
CHFUSD	1.7	4.69	9.97	2.9	1.42	1.07	0.56	1.3	0.79	1.23	7.74	0.73	0.79	1.96	1.24	0.51	**0.44**	0.45
EURCHF	1.78	2.76	6.66	1.49	1.2	0.84	0.57	1.15	0.39	1.47	4.6	0.55	0.57	3.21	2.2	0.35	**0.31**	0.32
EURGBP	1.28	2.42	11.31	1.02	0.83	0.58	0.52	0.64	0.39	0.6	1.86	0.45	0.68	1.61	1.0	0.42	**0.38**	0.39
EURPLN	0.85	2.01	8.39	1.12	0.65	0.6	0.47	0.85	0.39	1.04	1.74	0.41	0.49	1.41	0.65	0.41	**0.34**	0.36
EURUSD	1.35	2.95	6.55	1.86	1.18	1.03	0.67	1.07	1.24	1.13	2.28	0.68	0.83	2.27	1.82	0.49	**0.43**	0.43
GOOG	10.0	9.53	44.2	7.98	4.64	4.42	2.91	15.73	2.28	5.85	9.43	4.04	6.99	7.53	9.93	2.08	**1.77**	1.97
META	23.0	18.65	34.98	14.39	7.06	6.01	3.19	14.99	2.48	8.14	15.44	5.51	7.69	15.65	17.61	2.98	2.41	**1.90**
NFLX	18.02	29.01	61.32	16.19	8.92	7.86	5.33	28.51	3.46	9.9	33.45	5.55	5.73	21.48	21.07	2.86	**2.26**	2.28
PLNCHF	1.86	5.54	6.31	2.61	1.92	2.0	0.78	2.3	0.56	2.9	6.93	1.0	1.14	3.94	2.73	0.54	**0.51**	0.69
PLNGBP	0.61	2.04	10.86	1.27	1.01	0.73	0.7	1.28	0.55	0.78	2.28	0.61	0.83	2.11	1.38	0.54	**0.51**	0.56
TSLA	17.76	33.73	71.83	26.83	14.56	23.22	4.03	27.29	14.78	21.81	65.77	16.47	13.34	34.94	23.61	3.47	**2.98**	2.48
USDGBP	2.3	3.47	13.73	1.81	0.89	0.88	0.71	1.18	0.47	0.73	2.0	0.56	0.97	1.81	1.04	0.48	**0.45**	0.46
USDPLN	1.82	5.01	16.22	2.96	1.64	1.72	0.77	1.59	0.7	2.09	3.46	0.84	1.26	3.12	1.51	0.73	**0.61**	0.59
XAUUSD	5.76	8.55	28.57	5.53	3.14	3.55	1.42	2.43	1.35	4.67	13.33	1.74	1.74	6.74	3.45	0.94	**0.79**	0.90
CAC	2.41	7.19	26.6	2.51	2.18	1.93	1.52	3.03	1.0	2.11	6.19	1.27	2.08	4.75	3.49	1.21	**0.99**	0.92
DAX	3.66	8.89	34.69	3.77	2.7	2.71	1.62	5.21	1.07	2.14	6.01	1.64	2.5	4.93	4.32	1.2	**0.98**	3.63
DJC	4.64	8.2	38.77	4.07	3.52	2.67	1.28	3.52	1.16	2.27	11.45	2.1	1.92	6.74	5.92	1.0	**0.86**	1.22
DJI	5.85	9.01	38.47	3.09	2.51	2.21	1.48	4.65	1.12	2.42	15.79	2.01	1.79	4.15	5.91	0.97	**0.83**	1.20
FTM	4.55	9.58	33.35	3.47	3.42	3.01	1.38	3.92	1.04	3.38	16.33	2.05	2.04	5.57	3.49	0.98	**0.84**	0.96
KOSPI	3.54	8.65	31.47	4.7	2.06	2.83	1.75	2.85	1.15	2.28	5.59	1.41	2.36	4.28	4.09	1.03	**0.88**	1.11
NDX	7.77	11.68	49.29	7.26	3.99	4.65	2.2	7.07	1.47	4.84	34.68	3.04	3.47	8.89	7.94	1.25	**1.07**	24.63
NKX	3.75	10.5	29.97	4.8	3.05	4.19	2.14	3.69	2.59	3.06	10.39	1.55	2.97	8.44	4.03	1.18	**1.01**	0.58
SHC	1.71	7.52	19.43	2.26	1.76	1.24	1.15	3.39	0.86	1.33	6.3	0.97	1.59	3.38	2.21	0.94	**0.77**	0.78
SPX	4.86	8.05	40.44	4.91	3.44	2.95	1.63	6.66	1.21	4.66	18.57	2.07	2.38	7.43	5.64	1.04	**0.89**	1.21
WIG20	3.32	8.54	25.5	3.67	2.82	1.91	1.52	3.1	1.18	1.77	7.59	1.54	2.16	4.76	4.53	1.33	**1.11**	1.27

GOOG and META calculated for two forecasting horizons; TSLA and BTCUSD calculated for three forecasting horizons. Numbers in bold indicate minimum for a given timeseries.

Table 2. An average of the mean absolute percentage error at the first predicted time step for every model and every financial time series calculated over five different timespans.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**ARIMA**	**naive**	**ExpSmooth**
AAPL	6.64	16.68	62.55	3.97	3.31	2.91	**1.65**	2.96	2.04	1.69	10.48	2.99	1.98	6.37	5.80	2.16	2.03	1.90
AMZN	5.23	11.70	63.84	1.86	1.88	1.89	**1.54**	7.67	2.01	2.03	7.86	2.24	1.89	3.56	1.73	2.15	2.02	1.81
BTCUSD	15.14	26.29	73.84	8.68	9.68	11.37	4.49	10.38	3.34	**2.91**	9.83	4.65	7.26	15.16	10.13	4.11	3.29	3.21
CHFGBP	0.88	3.50	16.07	0.78	0.71	0.80	**0.36**	1.88	0.44	1.04	3.31	0.58	0.75	2.37	1.40	0.43	0.46	0.48
CHFUSD	0.80	4.36	9.62	2.53	0.37	0.51	0.44	0.79	0.47	0.58	3.42	0.53	**0.28**	0.91	0.33	0.40	0.41	0.42
EURCHF	0.57	2.76	4.44	0.51	0.54	0.25	0.28	0.58	**0.12**	0.26	3.48	0.20	0.18	1.92	1.61	0.16	0.14	0.17
EURGBP	0.85	2.61	9.11	0.96	0.79	**0.21**	0.26	0.62	0.38	0.42	2.27	0.24	0.47	0.98	0.50	0.47	0.34	0.34
EURPLN	0.44	1.61	6.60	0.50	0.41	0.31	0.56	0.64	**0.19**	0.34	1.29	0.33	0.41	0.71	0.54	0.31	0.27	0.25
EURUSD	**0.27**	2.23	4.90	1.01	0.53	0.57	0.49	1.02	0.40	0.27	2.10	0.40	0.82	0.96	1.37	0.31	0.40	0.38
GOOG	3.11	7.19	47.74	1.07	**0.56**	1.83	1.55	19.93	1.24	2.76	9.02	1.57	6.62	1.65	2.13	1.20	1.25	1.38
META	2.03	12.28	47.44	2.60	2.07	2.14	**0.60**	11.48	1.34	1.28	3.07	2.00	1.16	3.68	3.84	1.49	1.40	1.39
NFLX	9.50	13.23	55.41	3.70	1.59	1.39	2.59	19.01	1.66	4.36	9.67	1.26	4.00	4.35	3.36	1.66	1.56	**1.26**
PLNCHF	0.32	4.85	3.52	0.86	0.71	1.19	0.51	0.96	0.27	0.33	4.12	0.46	**0.25**	1.45	2.17	0.38	0.39	0.37
PLNGBP	0.22	1.20	10.07	0.59	0.74	0.28	0.23	1.20	0.41	0.32	2.83	0.23	**0.22**	0.97	0.53	0.39	0.38	0.36
TSLA	11.77	21.97	65.74	11.01	9.36	7.85	4.86	9.54	5.78	5.78	**4.09**	4.94	10.58	11.63	6.78	4.68	5.40	5.56
USDGBP	1.30	3.17	12.62	0.40	0.52	0.37	0.31	1.75	0.51	0.51	2.04	**0.19**	1.02	1.53	1.02	0.53	0.51	0.47
USDPLN	0.76	3.28	15.68	1.35	0.83	0.72	0.33	0.75	0.40	0.34	1.98	0.53	1.15	1.95	1.15	0.31	0.33	**0.30**
XAUUSD	1.20	6.98	26.94	1.21	1.50	0.88	**0.53**	1.27	0.76	1.00	5.12	0.75	1.17	1.15	1.09	1.06	0.80	0.79
CAC	**0.79**	7.72	30.38	1.52	1.57	1.54	1.08	2.04	1.63	1.30	4.87	1.42	1.49	3.59	2.66	1.61	1.46	1.46
DAX	1.63	8.79	36.95	2.12	2.28	1.64	**0.76**	4.24	1.27	1.15	4.66	1.22	1.61	3.17	3.66	1.61	1.22	1.28
DJC	2.72	6.95	38.85	1.31	1.88	1.34	1.07	**0.59**	0.77	1.73	6.72	1.21	0.70	3.51	2.61	1.03	0.80	0.77
DJI	2.61	7.17	38.40	1.68	1.10	1.39	1.01	1.99	**1.01**	1.11	7.44	1.35	1.73	1.88	3.38	1.23	1.10	1.08
FTM	1.62	9.71	37.59	1.83	3.22	2.25	**0.93**	2.69	1.09	1.05	5.82	1.69	1.24	3.77	3.02	1.45	1.28	1.64
KOSPI	1.26	6.42	31.89	2.01	1.20	1.33	1.30	2.18	0.81	**0.52**	3.18	1.29	2.62	1.18	1.25	1.21	0.80	0.77
NDX	2.96	8.35	48.43	3.06	2.31	2.57	1.66	3.18	**1.56**	1.59	6.95	1.86	2.21	3.56	3.59	1.68	1.67	1.60
NKX	**1.74**	6.67	30.44	2.75	2.46	1.96	2.32	3.71	1.87	2.18	5.78	2.19	2.81	2.55	2.90	2.68	2.10	2.04
SHC	1.24	4.93	22.28	1.77	1.98	1.70	1.22	2.91	0.93	1.29	5.30	1.39	**0.79**	2.60	2.32	1.52	0.86	1.00
SPX	2.10	7.42	40.64	2.86	1.43	2.20	1.31	5.93	1.11	3.78	7.87	1.48	1.86	4.81	2.50	1.30	1.18	**1.10**
WIG20	**0.56**	5.00	30.25	3.92	2.92	1.53	1.17	3.51	1.25	1.82	7.55	1.37	2.26	3.80	3.44	1.71	1.32	1.34

GOOG and META calculated for two forecasting horizons; TSLA and BTCUSD calculated for three forecasting horizons. Numbers in bold indicate minimum for a given timeseries.

5. Conclusions

The experiment presented in this paper aimed to compare the predictive performance of various deep learning models on different financial time series. The experiment was conducted using the data of daily prices for 29 financial time series, including stocks, indexes, and currency pairs, over a period of 15 years (2007–2022).

The models used in the experiment included classical statistical approaches such as exponential smoothing and ARIMA, as well as deep learning models such as NN, LSTM, CNN, or GAN. The models were trained using a sliding window approach and evaluated using the mean absolute percentage error, mean squared error, and mean absolute error, as well as the mean absolute percentage error at the first time step.

The results of the experiment showed that the best model was the naive approach, but when disregarding the statistical approaches, several deep learning models showed promising results. In particular, the multi-scale recurrent convolutional neural network (Model 7) achieved the best MAPE of 1.79% on average, while ModAugNet (Model 9) achieved a MAPE of 2.07%. The worst-performing model was Model 11, which utilised SVM after data preprocessing with the CNN.

Based on the results presented in this study, it can be concluded that simple time series models, even naive approach, can perform relatively well against more-complex deep learning models in forecasting financial time series, notably in the long run. However, deep learning models, in particular those using LSTM/RNN architectures or complex CNN functions, have the potential to outperform statistical models in the short term, provided they are regularly retrained and properly tuned.

It has also been observed that the quality of models tends to deteriorate if they are not retrained after a certain period of time. This highlights the importance of regular retraining of time series models to ensure the best-possible fit based on all the information available at the time of forecasting. In addition, it was noted that stocks tend to have a higher MAPE than indices or currency pairs, which may be due to their higher volatility and the need for more sophisticated modelling techniques.

Author Contributions: Conceptualization: M.B. and M.C.; methodology and software: M.B.; writing—original draft preparation: M.B.; validation: M.C.; formal analysis: all authors; writing—review and editing: M.C., K.K. and M.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Program of Integrated Activities for the Development of the University of Warsaw (ZIP Program), co-financed by the European Social Fund under the Knowledge Education Development Operational Program 2014-2020, Path 3.5.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analysed in this study. These data can be found here: https://stooq.com/ (accessed on 12 June 2023).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ARIMA	Autoregressive integrated moving average
BiLSTM	Bidirectional LSTM
CNN	Convolutional neural network
DFT	Discrete Fourier transform
EMD	Empirical mode decomposition
EMH	Efficient market hypothesis
GAN	Generative adversarial network
GRU	Gated recurrent unit
IMF	Inherent mode function

LSTM	Long short-term memory
MAE	Mean absolute error
MAPE	Mean absolute percentage error
MSE	Mean-squared error
NN	Neural network
OHLC	Open, high, low, close
RNN	Recurrent neural network
SFM	State frequency memory

Appendix A. Hyperparameters Used for Training

Table A1. List of hyperparameters used for training.

Model No.	NN Architecture	Epochs	Learning Rate	Optimiser	Batch Size	Steps Back	Data Sample	Performance Metrics
1	-	-	-	Adam	-	20	3407 (Jan 2004–Jan 2018)	MAE, RMSE, MAPE, R^2
2	-	-	-	SGD	-	3,4,5	Jan 2012–Dec 2016/Jan 2007–Dec2011	RMSE, MAE, MAPE
3	randomly selected number of layers (1–6) ensembled 10 times	200,000	0.0001	Adam	-	20	-	relative error
4	2 layer LSTM, with 1/2 and 1/3 input neurons	-	-	-	-	2, 4, 8, 16, 32, 64, 128, 256, 512	Jan 2010–Dec 2016 (testing last year)	MAPE
5	grid search over layer sizes (16, 32, 64, 128, 256)	-	0.001 (decreasing)	Adam	128	3, 5, 10, 15, 25	Jul 2016 - Dec 2016 minutely data	RMSE, MAE, MAPE
6	one layer, 4 neurons	1 or 2	-	Adam	-	-	Jan 1985-Aug 2018	RMSE
7	differently (3) scaled time series -> CNN (16 filters)-> GRU (16 × 3)	100	0.0005	Adam	32	30	Jan 2016–Dec 2016	accuracy
8	320 neurons LSTM x3	4500	0.0024	-	-	20	Jan 2002–Dec 2017	accuracy
9	1 LSTM 2 layers 5 and 3 neurons, 2LSTM: 4 and 2	200	0.00005	Adam	32	20	Jan 2000–July 2017	MSE MAE MAPE
10	-	4000	0.01	RMSProp	-	3, 5, 10, 15, 20	2007–2014	MSE MAE MAPE
11	G: LSTM -> 7 neuron FC D: FC NN with 3 layers (72, 100, 10 neurons)	-	-	-	-	5	last 20 years	MAE MSE MAPE
12	2–4 CNN layers	-	-	-	-	30, 40, 50, 60	1990–2014	MSE
13	-	20	0.001	Adam	-	-	1950-2016	MAE MSE MAPE
14	CNN -> MaxPooling -> LSTM -> Dense	-	-	AdaDelta	-	50	2008–2018	MSE
15	-	-	-	-	-	100	2008–2017	MSE

References

1. Ang, A.; Bekaert, G. Stock Return Predictability: Is It There? *Rev. Financ. Stud.* **2007**, *20*, 651–707. [CrossRef]
2. Campbell, J.Y.; Hamao, Y. Predictable Stock Returns in the United States and Japan: A Study of Long-Term Capital Market Integration. *J. Financ.* **1992**, *47*, 43–69. [CrossRef]
3. Granger, C.W.J.; Morgenstern, O. *Predictability of Stock Market Prices*, 1st ed.; Heath Lexington Books: Lexington, MA, USA, 1970.
4. Bollerslev, T.; Marrone, J.; Xu, L.; Zhou, H. Stock Return Predictability and Variance Risk Premia: Statistical Inference and International Evidence. *J. Financ. Quant. Anal.* **2014**, *49*, 633–661. [CrossRef]
5. Phan, D.H.B.; Sharma, S.S.; Narayan, P.K. Stock Return Forecasting: Some New Evidence. *Int. Rev. Financ. Anal.* **2015**, *40*, 38–51. [CrossRef]
6. Campbell, J.Y.; Thompson, S.B. Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? *Rev. Financ. Stud.* **2008**, *21*, 1509–1531. [CrossRef]
7. Agrawal, J.; Chourasia, V.; Mittra, A. State-of-the-Art in Stock Prediction Techniques. *Int. J. Adv. Res. Electr. Electron. Instrum. Energy* **2013**, *2*, 1360–1366.
8. Yim, J. A Comparison of Neural Networks with Time Series Models for Forecasting Returns on a Stock Market Index. In *Developments in Applied Artificial Intelligence*; Lecture Notes in Computer Science; Hendtlass, T., Ali, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 25–35. [CrossRef]
9. Bao, W.; Yue, J.; Rao, Y. A Deep Learning Framework for Financial Time Series Using Stacked Autoencoders and Long-Short Term Memory. *PLoS ONE* **2017**, *12*, e0180944. [CrossRef]
10. Lahmiri, S.; Bekiros, S. Cryptocurrency Forecasting with Deep Learning Chaotic Neural Networks. *Chaos Solitons Fractals* **2019**, *118*, 35–40. [CrossRef]
11. Long, W.; Lu, Z.; Cui, L. Deep Learning-Based Feature Engineering for Stock Price Movement Prediction. *Knowl.-Based Syst.* **2018**, *164*, 163–173. [CrossRef]
12. Chong, E.; Han, C.; Park, F.C. Deep Learning Networks for Stock Market Analysis and Prediction: Methodology, Data Representations, and Case Studies. *Expert Syst. Appl.* **2017**, *83*, 187–205. [CrossRef]
13. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
14. Oreshkin, B.; Carpo, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting. *arXiv* **2019**, arXiv:1905.10437.
15. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The M4 Competition: 100,000 Time Series and 61 Forecasting Methods. *Int. J. Forecast.* **2020**, *36*, 54–74. [CrossRef]
16. Sezer, O.; Gudelek, U.; Ozbayoglu, M. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [CrossRef]
17. Jiang, W. Applications of Deep Learning in Stock Market Prediction: Recent Progress. *Expert Syst. Appl.* **2021**, *184*, 115537. [CrossRef]
18. Reutskaja, E.; Lindner, A.; Nagel, R.; Andersen, R.A.; Camerer, C.F. Choice Overload Reduces Neural Signatures of Choice Set Value in Dorsal Striatum and Anterior Cingulate Cortex. *Nat. Hum. Behav.* **2018**, *2*, 925–935. [CrossRef]
19. Chen, L.; Chi, Y.; Guan, Y.; Fan, J. A Hybrid Attention-Based EMD-LSTM Model for Financial Time Series Prediction. In Proceedings of the 2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 25–28 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 113–118. [CrossRef]
20. Zhou, F.; Zhou, H.; Yang, Z.; Yang, L. EMD2FNN: A Strategy Combining Empirical Mode Decomposition and Factorization Machine Based Neural Network for Stock Market Trend Prediction. *Expert Syst. Appl.* **2018**, *115*, 136–151. [CrossRef]
21. Yang, B.; Gong, Z.J.; Yang, W. Stock Market Index Prediction Using Deep Neural Network Ensemble. In Proceedings of the 2017 36th Chinese Control Conference (CCC), Dalian, China, 11 September 2017; pp. 3882–3887. [CrossRef]
22. Li, Z.; Tam, V. Combining the Real-Time Wavelet Denoising and Long-Short-Term-Memory Neural Network for Predicting Stock Indexes. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8. [CrossRef]
23. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G. A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction. *arXiv* **2017**, arXiv:1704.02971.
24. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. A Comparative Analysis of Forecasting Financial Time Series Using ARIMA, LSTM, and BiLSTM. *arXiv* **2019**, arXiv:1911.09512.
25. Guang, L.; Xiaojie, W.; Ruifan, L. Multi-Scale RCNN Model for Financial Time-series Classification. *arXiv* **2019**, arXiv:1911.09359.
26. Zhao, Z.; Rao, R.; Tu, S.; Shi, J. Time-Weighted LSTM Model with Redefined Labeling for Stock Trend Prediction. In Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA, USA, 6–8 November 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 1210–1217. [CrossRef]
27. Baek, Y.; Kim, H.Y. ModAugNet: A New Forecasting Framework for Stock Market Index Value with an Overfitting Prevention LSTM Module and a Prediction LSTM Module. *Expert Syst. Appl.* **2018**, *113*, 457–480. [CrossRef]
28. Zhang, L.; Aggarwal, C.; Qi, G.J. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; ACM: New York, NY, USA, 2017; pp. 2141–2149. [CrossRef]

29. Cao, J.; Wang, J. Stock Price Forecasting Model Based on Modified Convolution Neural Network and Financial Time Series Analysis. *Int. J. Commun. Syst.* **2019**, *32*, e3987. [CrossRef]
30. Zhang, K.; Zhong, G.; Dong, J.; Wang, S.; Wang, Y. Stock Market Prediction Based on Generative Adversarial Network. *Procedia Comput. Sci.* **2019**, *147*, 400–406. [CrossRef]
31. Hossain, M.A.; Karim, R.; Thulasiram, R.; Bruce, N.D.B.; Wang, Y. Hybrid Deep Learning Model for Stock Price Prediction. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 1837–1844. [CrossRef]
32. Eapen, J.; Bein, D.; Verma, A. Novel Deep Learning Model with CNN and Bi-Directional LSTM for Improved Stock Market Index Prediction. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 264–270. [CrossRef]
33. Zhan, X.; Li, Y.; Li, R.; Gu, X.; Habimana, O.; Wang, H. Stock Price Prediction Using Time Convolution Long Short-Term Memory Network. In *Knowledge Science, Engineering and Management*; Lecture Notes in Computer Science; Liu, W., Giunchiglia, F., Yang, B., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 461–468. [CrossRef]
34. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
35. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.