mathematics

MDPI

Article

Traditional Prediction Techniques and Machine Learning Approaches for Financial Time Series Analysis

Claudia Cappello [1], Antonella Congedi [1], Sandra De Iaco [1,2,*] and Leonardo Mariella [1]

[1] Department of Economic Sciences, University of Salento, 73100 Lecce, Italy; claudia.cappello@unisalento.it (C.C.); antonella.congedi@unisalento.it (A.C.); leonardo.mariella@unisalento.it (L.M.)

[2] National Centre for HPC, Big Data and Quantum Computing, 40033 Bologna, Italy

* Correspondence: sandra.deiaco@unisalento.it

Abstract: Accurate financial time series forecasting is critical for effective decision making in areas such as risk management, portfolio optimization, and trading. Given the complexity and volatility of financial markets, traditional forecasting methods often fail to capture the underlying dynamics. Recent advances in artificial neural network (ANN) forecasting research indicate that ANNs present a valuable alternative to traditional linear methods, such as autoregressive integrated moving average (ARIMA). However, time series are typically influenced by a combination of factors which require to consider both linear and non-linear characteristics. This paper proposes a new hybrid model that integrates ARIMA and ANN models such as long short-term memory and gated recurrent unit neural network to leverage the distinct strengths of both linear and non-linear modeling. Moreover, the goodness of the proposed model is evaluated through a comparative analysis of the ARIMA, ANN and Zhang hybrid model, using three financial datasets (i.e., Unicredit SpA stock price, EUR/USD exchange rate and Bitcoin closing price). Various absolute and relative error metrics, computed to evaluate the performance of models, can support the use of the proposed approach. The Diebold–Mariano (DM) test is also implemented to asses the significance of the obtained differences of the hybrid model with respect to the other competing models.

Keywords: artificial neural network; autoregressive integrated moving average; financial time series prediction; exchange rate; stock prices; comparative analysis; optimal linear combination

MSC: 91B84

Academic Editors: Raydonal Ospina, Victor Leiva and Cecília Castro

Received: 20 December 2024

Revised: 27 January 2025

Accepted: 30 January 2025

Published: 6 February 2025

Citation: Cappello, C.; Congedi, A.; De Iaco, S.; Mariella, L. Traditional Prediction Techniques and Machine Learning Approaches for Financial Time Series Analysis. *Mathematics* **2025**, *13*, 537. https://doi.org/10.3390/math13030537

1. Introduction

In recent years, the rapid advancement of information technology (IT) and artificial intelligence (AI) has brought about a relevant societal transformation. The financial sector, in particular, has been significantly impacted by these advances. Machine learning models are now widely used for various purposes, including risk management, customizing investments to meet individual investor needs, and monitoring regulatory compliance [1–4]. By analyzing vast amounts of data, these models can quickly identify non-compliant behaviors according to regulatory standards, process historical data, and make market forecasts. Moreover, research efforts were also devoted to the development and improvement of models for time series forecasting. Although traditional models, such as the ARIMA [5], are considered effective techniques, stationary and non-stationary time series properly transformed to be stationary are required. Furthermore, predictions of future values are constrained to linear functions of past observations [6,7]. To overcome this limitation, several classes of non-linear models were introduced in the literature, such as the bilinear model [8,9], the threshold autoregressive model (TAR) [10], and the autoregressive conditional heteroscedastic model (ARCH) [11]. These non-linear models demonstrated some improvements; however, their applicability remains constrained, as they are designed for specific types of non-linearity. ANNs were proposed as alternative methods for time series forecasting, primarily due to their data-driven approaches and non-linear modeling capabilities [6,12–14]. A considerable variety of ANN structures were introduced in the literature. In particular, several models incorporate a non-linear function into the autoregressive model, and it is worth recalling ones such as the dynamic architecture neural network [15], the autoregressive neural networks [16], and recurrent and gated recurrent unit networks [17,18]. These approaches involve a non-linear transformation to the AR framework, exemplified by the multilayer perceptron, or integrating with models adept at capturing the linear aspects. Moreover, ANN models have gained significant attention in financial sector for their ability to uncover complex patterns in large datasets, enabling more accurate predictions and informed decision making in areas such as risk assessment, asset pricing, and portfolio optimization [19–22].

Even if ANN models address the limitations of ARIMA models, they have yielded inconsistent results when applied to purely linear time series. Moreover, several constraints related to the proper specification of the process and the lack of progress in broadening the MA model through the integration of ANN, in contrast to the advancements seen in expanding the AR model, were highlighted in [6,13]. Hence, as pointed out by [7,23], neither the pure ARIMA nor the pure ANN are adequate for modeling real-world time series, which typically exhibit linear and non-linear correlation structures.

In this context, hybrid models that combine ARIMA and ANN models were proposed in many domains of science, engineering and finance to effectively capture the linear and non-linear correlation structures of time series data. Among others, it is worth citing the work of Zhang [7], who first developed an additive hybrid model, where the time series is decomposed into the sum of linear and non-linear components, the ARIMA model is used to fit the linear component, and then the residuals are modeled by the ANN model. Khashei and Bijari [16] proposed a generalization of the additive model in [7], whereas Wang et al. [24] presented a new multiplicative hybrid model. In [23], a prior decomposition and reconstruction of the series through the discrete wavelet approach was proposed, where an ARIMA model was fitted to the reconstructed detailed part and the ANN was fitted to the residual of the previous stage and to the approximate part; then, the final forecast was derived as a combination of the forecasts obtained. Forecasting methods that combine ARIMA models with ANNs have become increasingly popular in financial forecasting. In particular, hybrid ARIMA-ANN models have shown great success in predicting stock prices and stock indices, which often involve both linear trends and non-linear movements, as recently discussed in [25–27]. In particular, in [25], the authors introduced a parallel hybrid model designed to capture pure linear, non-linear and mixed patterns. Four distinct modeling strategies were considered, pure linear, pure non-linear, linear to non-linear and non-linear to linear models, where the linear and the non-linear parts were modeled by using the ARIMA and the multiple-layer perceptron neural network model, respectively. Moreover, in the recent papers of [26], a hybrid model that combines depth learning and the ARIMA model was proposed, where the wavelet-based Daubechies method was combined with the ARIMA. On the other hand, in [27], the authors proposed a hybrid model that integrates the GARCH models, used to catch the volatility clustering effects and the LSTM to identify temporal patterns.

Although various models that integrate traditional models with ANNs were proposed in the literature to improve the forecasting accuracy, none of them is based on a convex combination of the values estimated by ARIMA and by the ANN. In this paper, a new hybrid model, where the linear behavior has been addressed by the ARIMA and the non-linear part by the ANNs, was proposed, and the final forecasts were obtained as a convex combination of the values estimated by ARIMA and by the ANN. The remainder of this paper is organized as follows: the ARIMA and the ANN models, i.e., the long short-term memory (LSTM) and the gated recurrent unit (GRU), are reviewed in Section 2. The details associated with the new hybrid model are introduced in Sections 3 and 4, where the forecast performance tools are also described. The computational aspects related to three financial data sets are discussed in Section 5. A discussion of a comparative analysis using the fitted models is provided in Section 6; finally, concluding remarks are presented in Section 7.

2. Theoretical Framework

In this section, the fundamental characteristics of ARIMA and ANN models will be reviewed.

2.1. ARIMA Approach

In 1970, Box and Jenkins [5] proposed a generalization of the autoregressive moving average (ARMA) class of models, comprising the ARIMA model, to forecast the future value of homogeneous non-stationary time series as a linear function of past observations and random errors.

Given a non-stationary stochastic process $\{X_t,\ t \in T\}$, where the trend could be approximated by a polynomial of degree d, the stationary process $\{W_t,\ t \in T\}$ is obtained by differentiating X_t at order d: $W_t = \nabla^d X_t$. If $W_t \sim ARMA(p,q)$, that is W_t is such that:

$$W_t = \phi_1 W_{t-1} + \cdots + \phi_p W_{t-p} + Z_t + \theta_1 Z_{t-1} + \cdots + \theta_q Z_{t-q}, \tag{1}$$

where Z_t follows a white noise model with zero mean and constant variance; then, the underlying stochastic process $\{X_t, t \in T\}$ is an ARIMA process of order (p,d,q): $X_t \sim ARIMA(p,d,q)$. Equivalently, the ARIMA process is defined as follows:

$$\phi(B)\nabla^d X_t = \theta(B) Z_t, \tag{2}$$

where $\nabla^d = (1-B)^d$ is the differentiating operator of order d, and $\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$ and $\theta(B) = 1 + \sum_{j=1}^{q} \theta_j B^j$ are polynomials of degree p and q, respectively. Note that, if $q = 0$ (or $p = 0$), then (2) reduces to an autoregressive model or order p (or a moving average model or order q).

A critical task in ARIMA model specification is determining the appropriate model order (p,q). According to the Box–Jenkins methodology [5], the autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the sample data can be useful tools to identify the order (p,q) of the ARIMA model.

After the specification of a tentative ARIMA model of order (p,d,q), the unknown parameters ϕ_i, $i = 1,2,\ldots,p$, and θ_j, $j = 1,2,\ldots,q$, are usually estimated through the maximum likelihood estimation. Finally, the adequacy of the model is evaluated, testing for white noise residuals. The ARIMA model was widely used in finance as a valuable tool in forecasting for capturing time-dependent patterns in financial data, such as stock prices, exchange rates, and interest rates, as pointed out in [28–30].

2.2. Artificial Neural Network

Artificial intelligence (AI) can be defined as a branch of computer science focused on enabling computers to perform tasks and functions that typically require human intelligence. During the recent decade, the significance of AI has grown substantially due to technological advancements that have enabled the creation of more sophisticated and powerful AI systems. These systems can solve an increasing variety of complex problems, thereby enhancing efficiency and automation in numerous fields, including finance. AI is now widely used in many areas of finance, such as credit and risk management, fraud detection, and financial asset price prediction [31–33].

Among the most important machine learning algorithms, artificial neural networks (ANNs) are specialized computer mathematical models designed to mimic the structure and function of the human brain.

In the literature, particular neural networks, known as recurrent neural networks, are widely used to model time series, since their complex and dynamic structure includes "recurrences" or "loops". These loops allow each network node to receive not only the current input data but also the output from the node itself or from previous nodes; this capability enables recurrent neural networks to remember past information. For predicting financial time series, some advanced recurrent neural networks can be employed, such as the long-short term memory (LSTM) and the gated recurrent unit (GRU) neural networks [34–36].

2.2.1. Long Short-Term Memory Networks

Long short-term memory (LSTM) neural networks are a specific type of recurrent neural network designed to select and retain relevant information over long periods. This selective long-term memory is achieved through the use of specialized cells included in the network's hidden layers, known as LSTM cells, which are advanced extensions of the traditional recurrent units. They typically consist of three "gates": the input gate, the forget gate, and the output gate [37,38]. The input gate determines how much input information (current input $x(t)$ and output of the previous cell $h(t-1)$) should be added to the memory; on the other hand, the forget gate determines how much of the current memory should be discarded. In Figure 1, the LSTM architecture is provided.

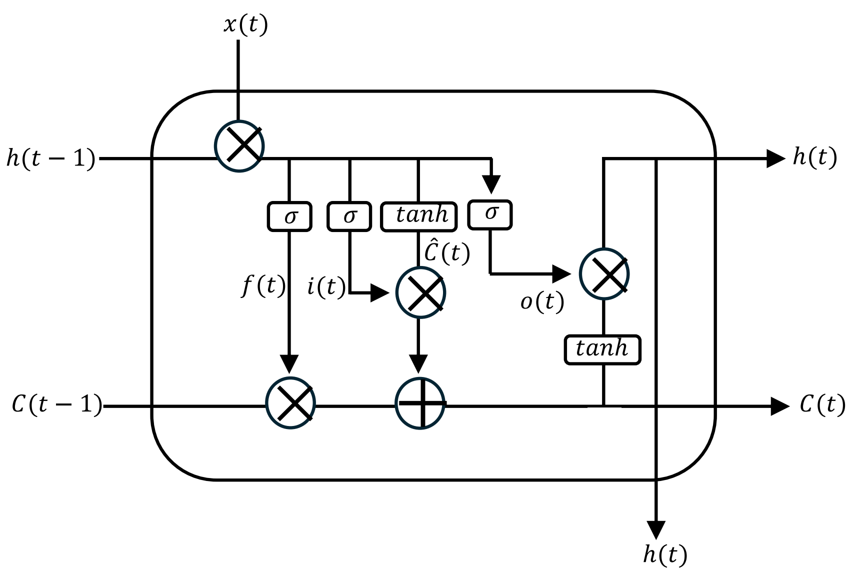


Figure 1. LSTM cell architecture.

Formally, the state of the cell $C(t)$, which represents the long-term memory of the LSTM network, is defined as follows:

$$C(t) = f(t) \otimes C(t-1) + i(t) \otimes \widehat{C}(t), \quad (3)$$

where $\otimes$ is the Hadamard product, $f(t)$ represents the output of the forget gate, $C(t-1)$ is the state of the cell at the previous time instant, $i(t)$ is the output of the input gate, and $\widehat{C}(t)$ is the proposed cell state update based on the input information. The output of forget gate $f(t)$ is obtained as follows:

$$f(t) = \sigma(w_{if} * x(t) + w_{hf} * h(t-1)), \quad (4)$$

where σ represents the sigmoid activation function, w_{if} is the matrix of weights associated with the current input, and w_{hf} is the matrix of weights associated with the previous output. The proposed cell state update is processed by the input gate according to the following equation:

$$\widehat{C}(t) = tanh(w_{ic} * x(t) + w_{hc} * h(t-1)), \quad (5)$$

where $tanh$ is the hyperbolic tangent activation function, w_{ic} is the weight matrix associated with the input $x(t)$, and w_{hc} is the weight matrix associated with the previous output. As mentioned earlier, not all of the input information will be retained, but only a portion of it, defined by the learning rate $i(t)$:

$$i(t) = \sigma(w_{ii} * x(t) + w_{hi} * h(t-1)), \quad (6)$$

where σ is the sigmoid activation function, w_{ii} is the matrix of weights associated with the current input, and w_{hi} is the matrix of weights associated with the previous output. The output gate, on the other hand, determines how much of the current memory is to be transmitted as the cell output. The output of the output gate is determined as follows:

$$o(t) = \sigma(w_{io} * x(t) + w_{ho} * h(t-1)), \quad (7)$$

where w_{io} is the matrix of weights associated with the current input, and w_{ho} is the matrix of weights associated with the cell output at the previous instant. Finally, the output of the cell at the current instant is given by

$$h(t) = o(t) \otimes tanh(C(t)). \quad (8)$$

2.2.2. Gated Recurrent Unit Networks

Gated recurrent unit (GRU) neural networks are advanced recurrent neural networks, designed to mitigate the short-term memory problem found in traditional recurrent neural networks [39,40]. Compared to long short-term memory (LSTM) networks, GRU networks allow us to reduce time processing, since their simpler architecture is designed for faster training. As in LSTM networks, specialized cells called "GRU cells" characterize hidden layers of the GRU network. These cells use a gating mechanism to manage the information within each cell, allowing the network to retain relevant information and discard the rest through two gates of the structure: the reset gate and the update gate.

Figure 2 provides a graphical representation of the GRU cell structure. At time t, upon receiving the current input $x(t)$ and the hidden state from the previous instant $h(t-1)$, the reset gate determines the past information to be forgotten, while the update gate decides how much of the new information should be propagated through the current state. The current hidden state is determined by a linear combination of the previous hidden state $h(t-1)$ and the candidate hidden state $\tilde{h}(t)$ as shown in the following:

$$h(t) = (1 - z(t)) \otimes h(t-1) + z(t) \otimes \tilde{h}(t), \tag{9}$$

where $\otimes$ is the Hadamard product, and $z(t)$ is the output of the update gate and it is determined as

$$z(t) = \sigma(w_{iz} * x(t) + w_{hz} * h(t-1)), \tag{10}$$

with w_{iz} being the matrix of input weights and w_{hz} being the matrix of weights relative to the previous hidden state. The candidate state $\tilde{h}(t)$ is given by the following analytical expression:

$$\tilde{h}(t) = tanh(w_{ih} * x(t) + (r(t) \otimes (w_{hh} * h(t-1))), \tag{11}$$

where w_{ih} is the matrix of input weights, w_{hh} is the matrix of weights related to the previous hidden state, and $r(t)$ is the output of the reset gate, as follows:

$$r(t) = \sigma(w_{ir} * x(t) + w_{hr} * h(t-1)) \tag{12}$$

with w_{ir} and w_{hr} being the matrices of weights associated with the current input and the previous hidden state, respectively, for the reset gate.

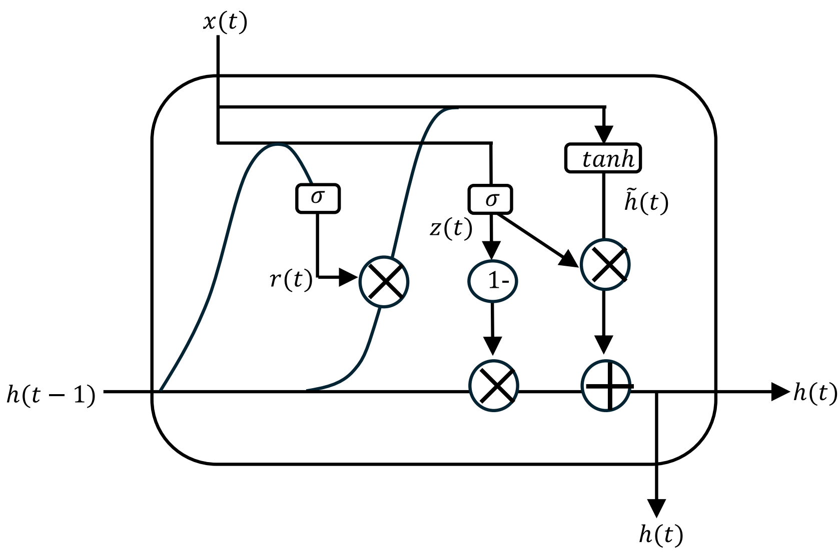


Figure 2. GRU cell architecture.

2.2.3. ANNs Hyperparameter Tuning

As detailed in the previous paragraph, ANNs are defined as a mathematical model characterized by several parameters that need to be learned from the data during the training process. However, another class of parameters known as hyperparameters affects the learning process of ANNs and cannot be directly learned from the data; it is worth recalling the speed of learning, the number of neurons or hidden layers, the batch size and the number of epochs. Hyperparameter tuning involves choosing the best settings to achieve optimal performance. In [41–43], a review of the most common strategies for identifying these values was proposed, such as empirical testing, which compares the combinations used in several trials by evaluating error metrics to identify the best performance; the random search, which considers all possible combinations after creating a grid of potential values of discrete hyperparameters; or randomized search, which selects random values and returns the combination that has provided the best result after several iterations. As detailed in Section 5, empirical testing was used in the proposed case studies by evaluating the performance of models with varying number of neurons, layers and batch values. Moreover, the early stopping criteria were considered when choosing the number of epochs in the learning process, stopping iterations when the optimal number of epochs has minimized the loss function [44,45].

3. A Hybrid Modeling Approach

ARIMA and ANN models are widely applied in finance for their linear and non-linear forecasting capabilities, respectively. However, taken individually, none of them is suitable for all types of time series, since real-world data exhibit both linear and non-linear correlation structures among the observations. To overcome this limitation and assuming that a time series could be decomposed into the sum of linear and non-linear components, different authors [7,16,23,24] proposed hybrid models, where the ARIMA model was used to fit the linear component. Moreover, residuals, which contain only the non-linear component, were modeled through an ANN model.

Unlike this common approach, the new advanced model takes into account the model errors of both ARIMA and ANN as well as the superiority of the ANN model with respect to the ARIMA model. To the best of our knowledge, the proposed hybrid model has not been presented in the literature and is an innovative approach that combines the benefits of both ARIMA and ANN models.

The predicted values ($\hat{x}_t$, $t = 1, \ldots, n$,) are calculated using a convex combination of the values estimated by ARIMA and the values estimated by the ANN at the current and preceding time points, as shown below:

$$\hat{x}_t = \alpha\left[\beta_t \hat{x}_{(t-1,ARIMA)} + (1-\beta_t)\hat{x}_{(t-1,ANN)}\right] + (1-\alpha)\left[\gamma \hat{x}_{(t,ARIMA)} + (1-\gamma)\hat{x}_{(t,ANN)}\right], \quad t = 1, \ldots, n, \tag{13}$$

where $\hat{x}_{(\cdot,ARIMA)}$ and $\hat{x}_{(\cdot,ANN)}$ are the forecasting values fitting the pure ARIMA and pure ANN models, respectively; α, β_t and γ are the weights estimated using testing data. In particular,

- α is a smoothing parameter, such that $0 < \alpha < 1$, obtained by minimizing the sum of the squares of the differences between the observed and the predicted values $\hat{x}_t$,
- $\beta_t \in [0,1]$, $t = 1, \ldots, n$ comprises the weight that takes into account the errors between the true values and the forecasting ones; specifically,

$$\beta_t = \begin{cases} 1 & \text{if } sign(e_{(t,ARIMA)}) = sign(e_{(t,ANN)}) \text{ and } |e_{(t,ARIMA)}| < |e_{(t,ANN)}| \\ 0 & \text{if } sign(e_{(t,ARIMA)}) = sign(e_{(t,ANN)}) \text{ and } |e_{(t,ARIMA)}| > |e_{(t,ANN)}| \\ 1 - \dfrac{|e_{(t,ARIMA)}|}{|e_{(t,ARIMA)}| + |e_{(t,ANN)}|} & \text{otherwise,} \end{cases} \tag{14}$$

where $e_{(t,ARIMA)} = \hat{x}_{(t,ARIMA)} - x_t$ and $e_{(t,ANN)} = \hat{x}_{(t,ANN)} - x_t$.

- $\gamma \in [0,1]$ is the weight that reflects the advantage of the ANN model over the ARIMA model, estimated with various training datasets by counting how often the absolute difference between the ANN predictions and actual values exceeds that between ARIMA predictions and actual values.

A schematic representation of the proposed hybrid model is shown in Figure 3, which describes the procedure steps to obtain the new combined results. In the newly proposed hybrid model, the data set has to be divided into training, validation, and testing subsets. At first, the training data are used to fit ARIMA and ANN models (Steps 1 and 2). After the validation of these models, ARIMA and ANN forecasts are obtained. Then the testing data are used to estimate the hybrid model's parameters (Step 3), and the final hybrid forecasts are obtained as detailed in (13).

The proposed hybrid model that combines the linear (ARIMA) and non-linear (ANN) models, through a weighted average, allows for us to obtain an improvement in terms of the predictive accuracy, even though this implies a computational cost implication. The convenience of using the new method depends on the impact of the linear and non-linear characteristics on the behavior of the variable under study.

It is worth pointing out that the assumption regarding linear versus non-linear decomposition included in the proposed approach inherits advantages and disadvantages of the solely linear models or solely non-linear models. It is well known that linear models are often preferred in applications since they are computationally less expensive than non-linear ones. For the latter, the fitting process often requires the use of numeric methodologies, which can generate convergence problems in searching for the optimal solutions. If a solution is found, it is sometimes not easily interpretable. On the other hand, linear models might provide a simplified representation which does not match the effective dynamics of the variable.

In the next section, the proposed novel model will be used to forecast three financial time series.

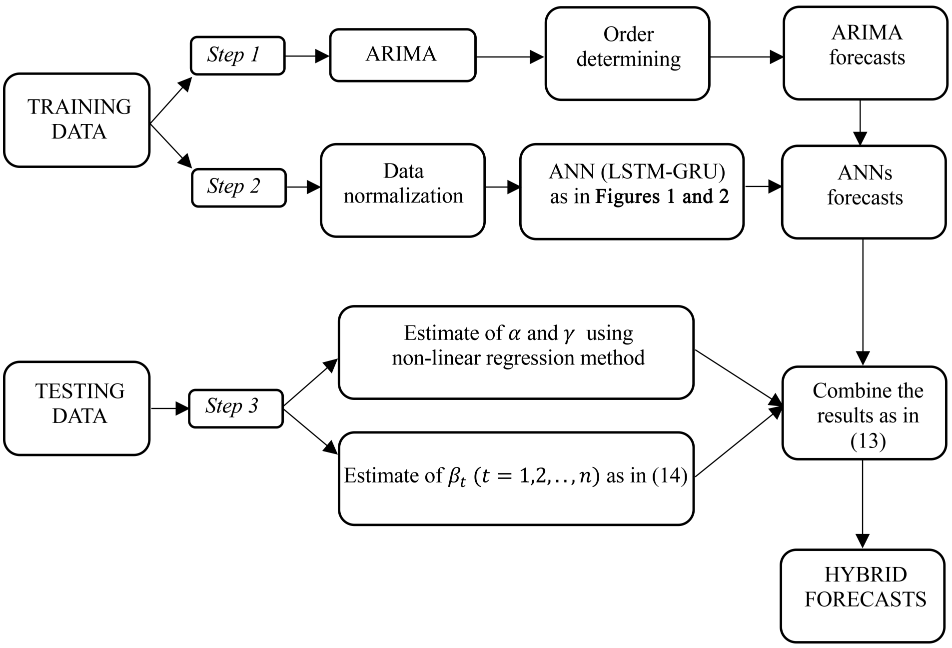


Figure 3. Schematic diagram of the new hybrid model structure.

Remark 1. *The implementation of a hybrid neural network model involves careful consideration of computational costs, data availability and training time. Practitioners need to ensure that the hardware used is sufficient to handle the complexity of the chosen ANN models. Regarding data availability, a model performs best when trained on high quality. Training times can vary according to the size of the model, the number of data and the chosen hyperparameters.*

4. Forecast Performance Assessment

In order to evaluate the predictive capabilities of the models' loss function, various metrics and the Diebold–Mariano test [46] were used in the case studies. In particular, among the loss metrics available in the literature, the mean absolute error (MAE), the root mean square error (RMSE), the mean absolute percentage error (MAPE) and the root mean squared error percentage (RMSEP) were considered:

$$MAE = \frac{1}{T}\sum_{t=1}^{T}|x_t - \hat{x}_t| \tag{15}$$

$$RMSE = \left[\frac{1}{T}\sum_{t=1}^{T}(x_t - \hat{x}_t)^2\right]^{0.5} \tag{16}$$

$$MAPE = \frac{1}{T}\sum_{t=1}^{T}\left|\frac{x_t - \hat{x}_t}{x_t}\right| \cdot 100 \tag{17}$$

$$RMSEP = \left[\frac{1}{T}\sum_{t=1}^{T}\left(\frac{x_t - \hat{x}_t}{x_t}\right)^2\right]^{0.5} \cdot 100 \tag{18}$$

where x_t and $\hat{x}_t$, $t = 1, \ldots, T$ are the observed value and the forecasted one, respectively. In addition, the Diebold–Mariano test [46] was applied to compare the predictive precision of the proposed hybrid model with respect to the pure ARIMA, the pure ANN and the Zhang hybrid model.

Given the forecast errors $e_{it} = \hat{x}_{it} - x_t$, $i = 1, 2$, $g(e_{it})$ is the loss associated with the i-th forecast, which is usually the square error loss or the absolute error loss:

$$g(e_{it}) = e_{it}^2 \quad \text{or} \quad g(e_{it}) = |e_{it}|.$$

The loss differential is defined as $d_t = g(e_{1t}) - g(e_{2t})$, and the hypothesis to be tested is

$$H_0 : E(d_t) \leq 0 \qquad H_1 : E(d_t) > 0. \tag{19}$$

The test statistic is

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi\hat{f}_d(0)}{T}}} \rightarrow N(0,1), \tag{20}$$

where

- $\bar{d} = \frac{1}{T}\sum_{t=1}^{T} d_t$,
- $\hat{f}_d(0)$ is a consistent estimator of the spectral density function of the loss differential $f_d(0) = \frac{1}{2\pi}\sum_{k=-\infty}^{\infty}\gamma_d(k)$, and $\gamma_d(k)$ is the autocovariance function.

As detailed in [46], $2\pi\hat{f}_d(0) = \sum_{\tau=-(T-1)}^{(T-1)} I\left(\frac{\tau}{h-1}\right)\hat{\gamma}_d(\tau)$, $\hat{\gamma}_d(\tau) = \frac{1}{T}\sum_{t=|\tau|+1}^{T}(d_t - \bar{d})$ $(d_{t-|\tau|} - \bar{d})$ is the sample autocovariance, and $I\left(\frac{\tau}{h-1}\right) = 1$ for $\left|\frac{\tau}{h-1}\right| \leq 1$ or 0 otherwise.

5. Case Studies

In this paper, three financial data sets extracted from Yahoo Finance (https://finance.yahoo.com, 20 January 2025), i.e., the Unicredit SpA stock, the Bitcoin prices and the nominal EUR/USD exchange rates, were analyzed to exhibit the effectiveness of the proposed hybrid model, with respect to the pure ARIMA, pure ANNs and Zhang hybrid model [7], obtained by combining ARIMA and LSTM as well as ARIMA and GRU. In addition, one-step-ahead forecasts were proposed. Gretl and Python software packages were used for data analysis. In particular, the Python Keras package was used to implement ANNs. The corresponding code is available upon request from the corresponding author.

5.1. Unicredit SpA Stock Price

The Unicredit data set refers to the daily dividend-adjusted closing price, in euros, of the Unicredit SpA stock, collected from 2 January 2019, to 26 September 2023, providing a total of 1207 temporal observations. The time series plot (Figure 4) suggests that there is an increasing trend, starting from the second part of 2022; hence, the series exhibits non-stationary. The data range between 5.25 and 23.67, presenting a mean value equal to EUR 10.48 and a mean square deviation of 4.10, indicating that prices typically deviate by about EUR 4.10 from the average; with respect to the shape of the data distribution, the skewness coefficient of 1.56 indicates a strong positive skewness (Figure 4b). The time series presents two deep drops in March 2020 and March 2022, caused by the volatility of the markets, which were influenced by extraordinary events such as the pandemic crisis and the Russo-Ukrainian war.

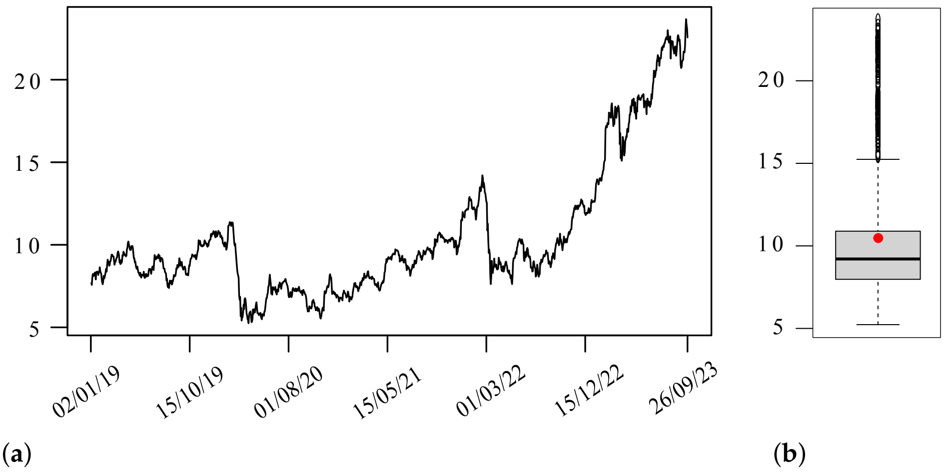


Figure 4. (**a**) Daily temporal evolution and (**b**) box plot of the dividend-adjusted closing price of the Unicredit SpA stock.

In the following, the ARIMA model and the ANNs (i.e., the LSTM and GRU) will be fitted to the Unicredit SpA time series and validated; then, the new proposed hybrid model will be defined according to the previous fitted models.

To identify the order of the ARIMA model, the time series plot (Figure 4a) and the ACF (Figure 5a) were evaluated. Looking at Figure 4a), a trend was detected, Moreover, the presence of a non-stationary trend was confirmed from the ACF (Figure 5a), which decays very slowly to zero for high lags. Consequently, the difference method of transformation was adopted, and second differences ($d = 2$) of the series were made. Moreover, from the ACF and PACF of the differenciated series (Figure 5b,c), an ARIMA of order $(1, 2, 1)$ was identified as follows:

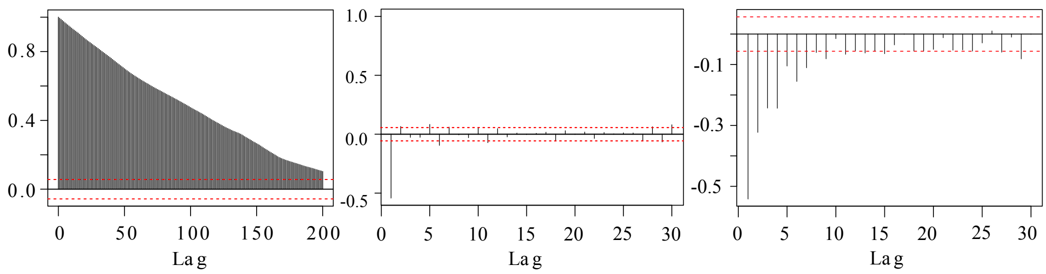
$$\phi(B)\nabla^2 X_t = \theta(B)Z_t. \tag{21}$$

where $\phi(B) = (1 - \phi_1 B)$, $\theta(B) = (1 + \theta_1 B)$, and Z_t is a white noise random process with variance ${\sigma_Z}^2$. Note that during the model validation process, both the ACF and the PACF of the residuals have shown a statistically significant peak at the fifth lag, which was considered in the model. Then, the model's parameters were estimated using the maximum likelihood estimation technique and the fitted model as follows:

$$W_t = 0.0881462W_{t-5} + Z_t - 0.997509Z_{t-1}, \tag{22}$$

where $W_t = \nabla^2 X_t$.

The parameter estimates are $\hat{\phi}_5 = 0.0881462$ and $\hat{\theta}_1 = -0.997509$, with corresponding p-values being smaller than 0.01; the 95% confidence intervals for these coefficients are $[0.0311586; 0.145134]$ and $[-1.00294; -0.992075]$, respectively.



(**a**) (**b**) (**c**)

Figure 5. ACF of the dividend-adjusted closing price of the Unicredit SpA stock (**a**), ACF (**b**) and PACF (**c**) of second-order differences.

With regard to ANNs, the LSTM and GRU models were considered. Note that, before defining the ANNs structure, the data were scaled between 0 and 1 in order to prevent the saturation of the networks' activation functions, through the min–max normalization. For training the models, the dataset was divided into three parts: a training set (80% of the sequences), a validation set (10% of the sequences), and a test set (10% of the sequences). Multiple trials were carried out in order to choose the appropriate values of parameters for the ANNs building. As shown in Figure 6, the prediction accuracy of models with a varying number of neurons and layers was evaluated on the test set. In particular, starting from a minimum number of hidden layers equal to 2 and a minimum size of the input layer equal to 32 neurons, an increase in the number of neurons from 32 to 64 or 128 slightly improves the accuracy of the prediction. The number of neurons was fixed and equal to 64, representing the optimal balance between model complexity and computing time. In addition, a comparison between 6 and 8 hidden layers was conducted; however, increasing the number of layers caused prediction instability; thus, the optimal number of hidden layers was set equal to 2.

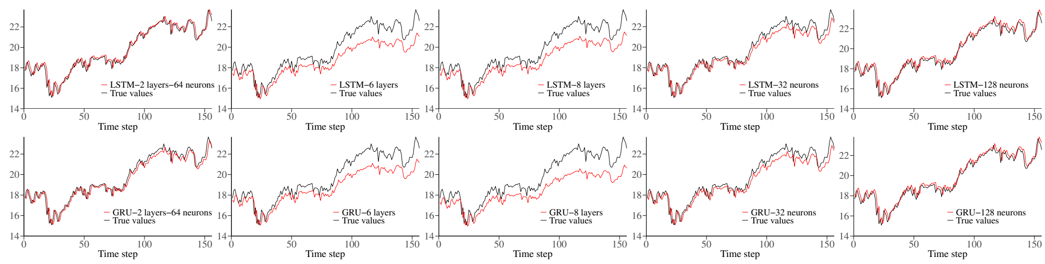


Figure 6. Dividend-adjusted closing price of Unicredit SpA stock prediction accuracy for LSTM and GRU with varying number of hidden layers and neurons.

In Table 1, the sequential configuration evaluated as an appropriate trade-off between accuracy and ANNs complexity was summarized. In particular, one linear input layer of 64 neurons and two hidden dense layers of 32 and 16 neurons, respectively, with the hyperbolic tangent activation function and an output layer of length equal to 1, were set. The networks were compiled by adopting the mean square error as the loss function and the adaptive moment estimation (*Adam*) as the optimization algorithm, which uses a default learning rate value equal to 0.001. The number of epochs was decided by using the early stopping function of the Keras Python package. Starting from a large number of epochs equal to 200, the loss function wasn minimized at the 107th epoch; finally, the batch size was tuned, starting with a small value that was gradually increased while monitoring the model's performance, until a balance between speed of convergence and stability was reached, with a batch size equal to 64. Then, the reliability of the fitted models mentioned above was evaluated by comparing true values and predicted ones; in particular, error metrics in ((15)–(18)) were computed and are summarized in Table 2. The results obtained to validate the ARIMA and ANN models show that the predicted values are strictly close to the test values: absolute errors MAEs and RMSEs are relatively low (0.1–0.5) when considering that the data range from 15.09 to 23.67; moreover, the relative errors, MAPEs and RMSEPs, are very close to zero (1–3%). These results confirmed the adequacy of the ARIMA model in (22) and ANNs for describing the temporal correlation; hence, they were used to perform the one-step ahead forecasts for the dividend-adjusted closing price of the Unicredit SpA stock for the 22 days out-of-sample period, i.e., from the 27th of September to the 26th of October.

Furthermore, the new hybrid model in (13) has been defined by considering the ARIMA and ANN validated models, and the estimated parameters are reported in Table 3.

Table 1. ANNs' hyperparameter settings in the three case studies.

	Unicredit	Bitcoin	EUR/USD
Number of units in LSTM/GRU cells	64	64	64
Number of dense layers	2	3	3
Activation function in a dense layer	tanh	tanh	tanh
Number of epochs	107	128	102
Batch size	64	64	32
Learning rate	0.001	0.001	0.001

Table 2. Error metrics for model validation.

	Unicredit		Bitcoin		EUR/USD	
	MAE (MAPE%)	RMSE (RMSEP%)	MAE (MAPE%)	RMSE (RMSEP%)	MAE (MAPE%)	RMSE (RMSEP%)
ARIMA	0.186	0.270	514.884	895.407	0.004	0.005
	(1.875)	(2.696)	(2.321)	(3.665)	(0.350)	(0.468)
LSTM	0.330	0.436	427.094	511.490	0.004	0.005
	(1.734)	(2.363)	(1.674)	(2.184)	(0.369)	(0.457)
GRU	0.322	0.436	554.111	662.773	0.004	0.005
	(1.700)	(2.387)	(2.173)	(2.626)	(0.376)	(0.468)

Table 3. Estimated parameters for the new hybrid model in (13). The acronyms AL and AG refer to the hybrid model that combines ARIMA and LSTM as well as ARIMA and GRU, respectively.

	Unicredit		Bitcoin		EUR/USD	
	AL	AG	AL	AG	AL	AG
$\hat{\alpha}$	0.977	0.974	0.990	0.987	0.990	0.988
$\hat{\gamma}$	0.59	0.36	0.53	0.27	0.41	0.27

5.2. Bitcoin Closing Price

The Bitcoin closing price data set refers to the daily closing price, in euros, of the most popular cryptocurrencies, from 1 January 2019, to 26 September 2023, providing a total of 1730 temporal observations. Figure 7a shows the temporal behavior of the Bitcoin daily closing price. The average closing price recorded during the observation period was EUR 21,269, with a mean square deviation of EUR 13,888; the lowest observed price was EUR 2998.2 on 7 February 2019, while the highest was EUR 58,305.0 on 8 November 2021. After increasing closing prices in 2021 and the first six months of 2022, Bitcoin's price declined from around EUR 55,000 to a low of about EUR 25,000; this negative trend is one of the signs of the instability of financial markets, after the European Central Bank's announcement to increase interest rates in order to address the surge in inflation.

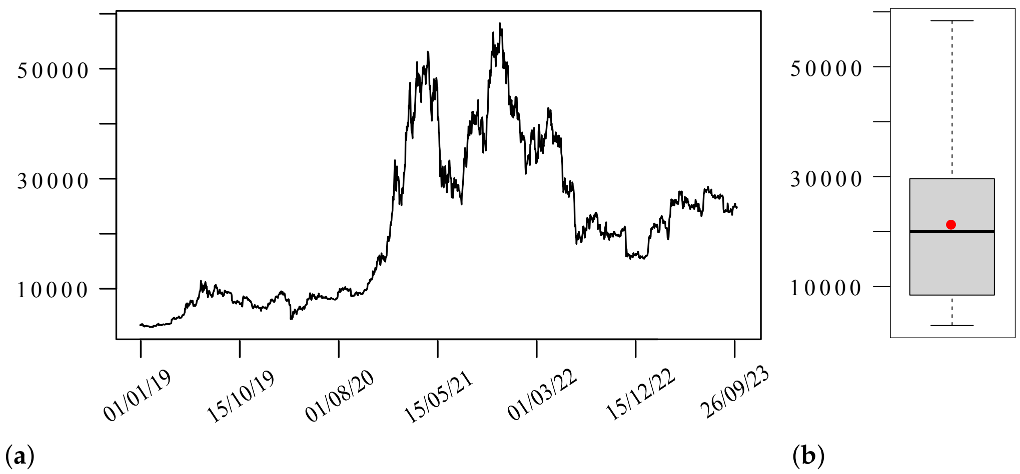


Figure 7. (**a**) Daily temporal evolution and (**b**) box plot of Bitcoin daily closing price.

After a preliminary logarithmic transformation of the data performed to stabilize the variance, the ACF was computed and decayed slowly to zero for very high lags (Figure 8a). As in the previous case study, this behavior of the ACF confirms the assumption that the time series is the finite realization of a non-stationary stochastic process. Thus, the second differences ($d = 2$) of the data were made. Moreover, looking at the ACF and PACF of the differentiated time series (Figure 8b,c), an ARIMA of order $(1, 2, 2)$ was identified as follows:

$$\phi(B)\nabla^2 X_t = \theta(B)Z_t, \tag{23}$$

where $\phi(B) = (1 - \phi_1 B)$, $\theta(B) = (1 + \theta_1 B + \theta_2 B^2)$, and Z_t is a white noise random process with variance ${\sigma_Z}^2$. Looking at the ACF and the PACF of the residuals (Figure 8b,c), a statistically significant peak at the fourth lag was considered in the model; hence, the model's parameters were estimated using the maximum likelihood estimation technique and the fitted model, as follows:

$$W_t = 0.0593938W_{t-4} + Z_t - 1.05180Z_{t-1} + 0.0536966Z_{t-2}, \tag{24}$$

where $W_t = \nabla^2 X_t$. The 95% confidence intervals for the coefficients ϕ_4, θ_1 and θ_2 are [0.0121609; 0.106627], [-1.06402; -1.039959] and [0.0409314; 0.0664618], respectively, with associated p-values being lower than 1%.

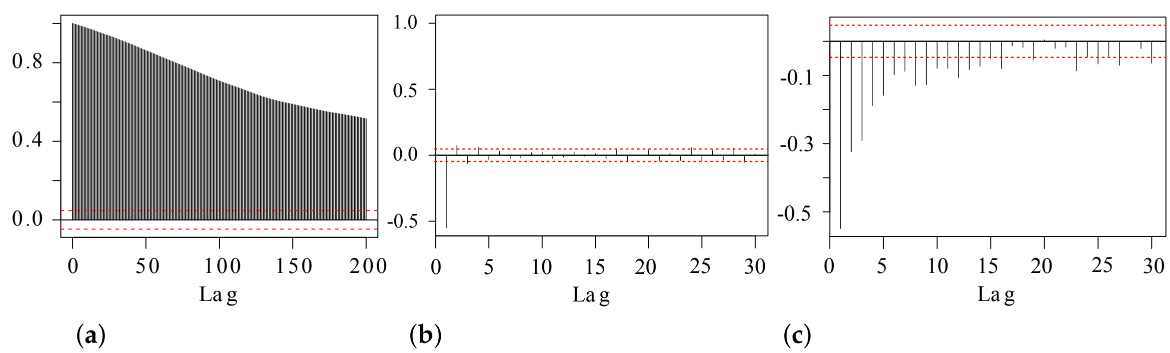


Figure 8. ACF of the Bitcoin daily logarithmic closing price (**a**), ACF (**b**) and PACF (**c**) of the second-order logarithmic differences.

As already detailed in the previous case study, for applying ANN algorithms, min–max normalization was used for scaling the time series between 0 and 1, and the data set was divided into the training set (80% of the data), a validation set (10% of the data) and a test set (10% of the data). In order to determine the appropriate hyperparameters of the two ANNs, different combinations with varying number of neurons and layers were compared. A minimum number of hidden layers equal to 3 and a minimum number of neurons in the input layer equal to 32 were considered. As shown in Figure 9, too many neurons do not provide better predictions; thus, the number was fixed to 64. Moreover, increasing the number of hidden layers causes a decrease in accuracy, suggesting vanishing gradient issues. Hence, a minimum number of 3 layers was chosen. The following sequential structure shared by LSTM and GRU models was defined (Table 1): one linear input layer comprising 64 neurons and three dense layers of 32, 24 and 16 neurons, respectively, with the hyperbolic tangent as an activation function and one output layer with the number of units equal to 1. Moreover, when compiling the models, the mean square error was chosen as the loss function and the Adam as the optimization algorithm, with its default learning rate value being equal to 0.001. The early stopping function of the Keras Python package was used to set a number of epochs equal to 128, when the loss function was minimized; finally, the batch size was tuned to 64, after gradually increasing it until a balance between speed of convergence and stability was reached.

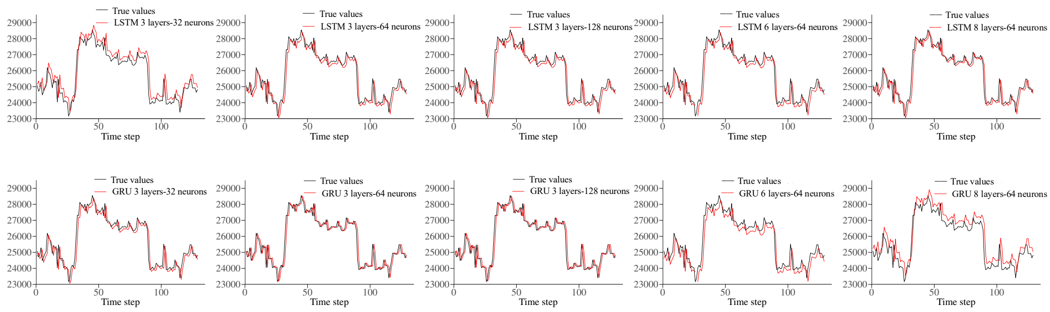


Figure 9. Bitcoin closing price prediction accuracy for LSTM and GRU with varying number of hidden layers and neurons.

In order to validate ARIMA, LSTM and GRU models, the obtained predictions were compared with the values on the test set. Considering the range of the data, the MAEs and RMSEs were quite low (400–900), and the models were also validated in relative terms, since the MAPEs and the RMSEP ranged between 2% and 4% (Table 2). These results confirmed that the ARIMA model in (24) and the ANNs are appropriate to describe the temporal correlation; thus, they were used to forecast the Bitcoin closing price in the out-of-sample time points, i.e., from the 27th of September to the 26th of October (30-day open market). Finally, the proposed hybrid model in (13) was defined using the estimated parameters in Table 3 and using the predictions obtained using the above-mentioned models.

5.3. EUR/USD Exchange Rate

The daily euro–dollar (EUR/USD) exchange rate represents how many US dollars are needed to buy one euro, i.e., the amount of foreign currency (dollars) required to purchase a unit of the domestic currency (euro). After the gradual drop from June 2021, Figure 10a shows an increasing trend of the variable under study starting from the end of 2022. During the observed period, from 1 January 2019, to 26 September 2023, the average exchange rate was approximately USD 1.12 per euro, with a standard deviation of 0.06. Data ranged between a minimum value of USD 0.96, observed on 28 September 2022, and a maximum value of USD 1.23, registered on 7 January 2021, respectively. Note that in 2022, the euro fell below parity with the dollar, due to expectations for rapid interest rate increases by the U.S. Federal Reserve to combat inflation and due to the energy crisis. The recovery in 2023 was determined by the slowdown of the Federal Reserve's increases and the encouraging signals from American inflation.

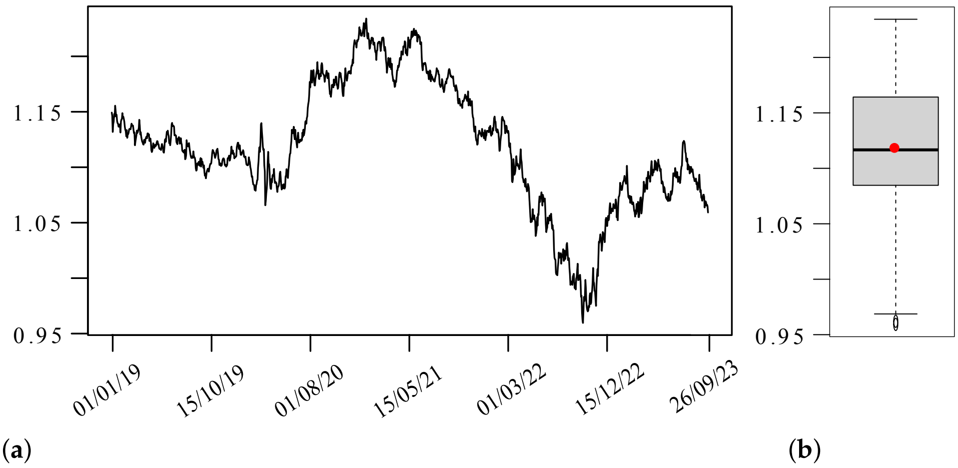


Figure 10. (**a**) Daily temporal evolution and (**b**) box plot of the daily euro–dollar (EUR/USD) exchange rate.

Before applying the new proposed approach, the time series was modeled through ARIMA and ANNs, which were validated. As in the first two case studies, Figure 11a shows that the performed ACF decays slowly to zero and confirms the non-stationarity of the time series.

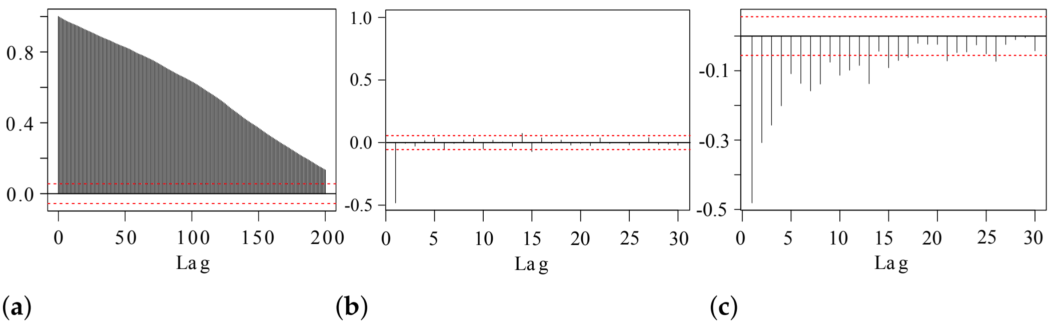


Figure 11. ACF of the daily euro–dollar (EUR/USD) exchange rate (**a**), ACF (**b**) and PACF (**c**) of the second-order differences.

The ARIMA model defined in (21) was fitted to the time series. Moreover, since the ACF and PACF (Figure 11b,c) of the residuals in the validation step showed a statistically significant peak at the sixth lag, the following model was fitted and then used to obtain the out-of-sample forecasts (22-day open market from the 27th of September to the 26th of October):

$$W_t = -0.101956 W_{t-6} + Z_t - 0.945228 Z_{t-1}, \quad (25)$$

where $W_t = \nabla^2 X_t$. The model coefficients, which were estimated using the maximum likelihood estimation technique, were statistically significant since the associated p-values were less than 0.01; the 95% confidence intervals for ϕ_6 and for θ_1 were $[-0.148944; -0.0549683]$ and $[-0.962380; -0.928076]$, respectively. Since neural networks, particularly the LSTM, are sensitive to the input data scale, the time series was normalized through the above-mentioned min–max normalization and divided into the training set (80%), the validation set (10%) and the test set (10%). Then, different models with a varying number of neurons and layers were fitted to the validation set to select an appropriate number of parameters of the ANNs structure. As in the previous case, the starting number of hidden layers and input neurons were chosen to be equal to 3 and 32, respectively. As shown in Figure 12, the prediction accuracy was not particularly affected by increasing the number of hidden layers and neurons; thus, the LSTM and the GRU networks were built with the same sequential configuration (Table 1), which includes one linear input layer comprising 64 neurons and three dense layers of 32, 24, and 16 neurons, respectively, with the hyperbolic tangent activation function and one output layer of 1 unit.

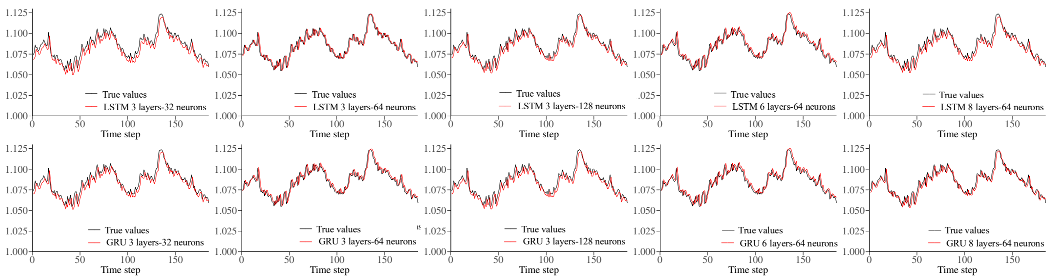


Figure 12. EUR/USD exchange rate prediction accuracy for LSTM and GRU with a varying number of hidden layers and neurons.

As in the previous case studies, artificial networks were compiled by choosing the MSE loss function and the *Adam* optimization algorithm, by using its default learning rate value equal to 0.001. The number of epochs was set equal to 102 by using the early stopping function of the Keras Python package, which stops the process when the loss function is minimized. Starting with a small value and gradually increasing it, the batch size was set equal to 32 by monitoring the performance of the model in order to balance the speed of convergence and stability of the ANNs.

Error metrics computed between estimated and real values (Table 2) confirmed the reliability of the ARIMA model, LSTM and GRU networks. Indeed, MAEs and RMSEs were significantly low when comparing them to the range of data and relative errors. Moreover, MAPEs and RMSEPs were very close to zero (0.4–0.5%). Since these error metrics confirmed the adequacy of the ARIMA model in (25) and ANNs to describe a temporal correlation, they were combined to forecast daily euro–dollar exchange rates for the out-of-sample period (22 days), i.e., from the 27th of September to the 26th of October.

The estimates of the parameters α and γ of the new hybrid model are shown in Table 3. They were obtained by carrying out the non-linear regression method developed in SPSS on the basis of the Newton–Lagrange approach [47]. Note that the estimates of the remaining parameters β_t were computed as specified in Section 3.

6. Comparative Analysis Using the Fitted Models

In this section, the performance of the proposed hybrid model was compared with the one proposed in [7]. Zhang's approach involves two distinct phases: initially, an ARIMA model is employed to examine the linear component of the time series; then, a neural network model is developed to address the residuals from the ARIMA analysis. The ARIMA model is limited in capturing the data's linear aspects; thus, the linear model's residuals are indicative of these non-linear features.

On the other hand, the hybrid approach proposed in this paper is based on the convex combination of the ARIMA and ANN estimated values at the current time and at the previous temporal point, including weights, which reflect the advantage of using an ANN model with respect to the ARIMA model.

After the validation phase (Section 4), out-of-sample predictions were obtained for the period from the 27th of September to the 26th of October 2023. The performance of the proposed hybrid model was evaluated by comparing the discrepancy between the true values and the fitted ones, in terms of MAE, RMSE, MAPE and RMSEP. In particular, the error statistics associated with the pure ARIMA, ANN models and the hybrid models are specified in Table 4. Then, the relative variation (Δ, expressed as percentage), which measures the relative enhancement/worsening (negative/positive sign, respectively) of the novel hybrid model against the others, was calculated for each of the error metrics.

Results of Unicredit stock prices forecasts in Table 4 show that the proposed model has much more accuracy than the other models used for a comparison: the proposed combination of ARIMA and LSTM predictions improves the pure ARIMA and the pure LSTM by around 5%, and it also outperforms the Zhang hybrid model [7] by around 3–4% in terms of the MAE and MAPE. Similar results were obtained from the convex combination of the ARIMA and GRU model predictions: the new proposed model improves upon the ARIMA by approximately 2–3%, the pure GRU by about 23–24%, and the Zhang hybrid model [7] by around 0.5–1%, in terms of the MAE and MAPE. The new combination comprising ARIMA and ANNs also outperforms the other models in terms of quadratic error.

Table 4 also shows Bitcoin price forecasts obtained by the new combinations of ARIMA with LSTM and GRU predictions, which improve the accuracy of the pure ARIMA by approximately 6% and 1% in terms of MAE and MAPE, respectively; and surpass the pure LSTM and the pure GRU models by approximately 1% and 12%, respectively. Furthermore, compared to the competing hybrid model, predictions obtained using the proposed hybrid model exhibit lower MAE and MAPE (between 6 and 9%). When considering the squared errors (RMSEs and RMSEPs), the new model outperforms ARIMA and the Zhang hybrid by approximately 2–5%. However, the combination of ARIMA and the GRU network provides a significant improvement to the pure GRU (equal to 8%).

Error metrics about EUR/USD exchange rate forecasts obtained by combining the ARIMA and ANNs are also reported in Table 4. Similarly to the previous cases, the new combination enhances ARIMA forecasts from 2% to 6% in terms of the MAE and MAPE; moreover, the proposed model combined with GRU significantly outperforms the pure GRU and the classical hybrid model [7] by 16–17%. Similar results were obtained in terms of RMSEs and RMSEPs, since the new combination enhances the ARIMA model by 1 to 4%, the pure ANNs by 4–9% and the Zhang hybrid model [7] by around 2–8%.

In order to prove the goodness of the novel hybrid approach, a hypothesis test was performed to check that the differences between true values and estimated ones are on average equal to zero ($\delta = 0$); hence, the test statistics and the associated p-values are also reported in Table 4.

As expected, the null hypothesis was not rejected at the significance level 5%. Thus, no relevant differences between the estimated values and the true values were detected. However, it is worth pointing out that the test on the proposed model has higher p-values than the ARIMA model and the Zhang hybrid models regarding the Bitcoin time series. Moreover, for the EUR/USD exchange rate predictions, the hypothesis test on the new hybrid combination of ARIMA and GRU presents the highest associated p-value, equal to 0.567. Finally, the ARIMA model obtained the highest p-value only for Unicredit SPA stock prices; however, the p-values of the proposed models were much higher than those associated with the pure neural networks models, confirming the importance of combining traditional models with a machine learning approach. The forecasting performance of the new hybrid model respect to the other models has also been evaluated by employing the DM test in (20), referring to the absolute error loss function. From the results in Table 4, it is evident that the forecast accuracy of the new hybrid model is generally more efficient than the pure ARIMA or ANN and Zhang hybrid model (LSTM), since the null hypothesis states that the average difference between the errors obtained by using the new hybrid model and the ones determined by adopting the other existing models is not positive ($H_0 : E(d_t) \leq 0 \quad H_1 : E(d_t) > 0$), so it is not rejected (the p-values are very high and sometimes close to 1). On the other hand, there are few cases where the p-values of the test statistic for the fixed null hypothesis are lower than 0.05 but higher than 0.01. Thus, there is no strong evidence that the null hypothesis is rejected, especially when the forecast errors of the new hybrid model and the Zhang hybrid model (GRU) are compared.

Table 4. Statistics on the prediction errors for the three time series.

	Unicredit		**Bitcoin**		**EUR/USD**	
	MAE (MAPE%)	**RMSE (RMSEP%)**	**MAE (MAPE%)**	**RMSE (RMSEP%)**	**MAE (MAPE%)**	**RMSE (RMSEP%)**
ARIMA (2)	0.348 (1.547)	0.394 (1.754)	411.834 (1.500)	677.612 (2.319)	0.004 (0.326)	0.004 (0.414)
LSTM (3)	0.341 (1.543)	0.415 (1.836)	391.580 (1.427)	645.282 (2.216)	0.004 (0.343)	0.004 (0.420)
Zhang Hybrid (LSTM)	0.341 (1.917)	0.518 (1.771)	415.178 (1.600)	677.991 (2.321)	0.003 (0.326)	0.004 (0.409)
New Hybrid (LSTM)	0.331 (1.467)	0.394 (1.745)	387.61 (1.413)	644.893 (2.210)	0.003 (0.335)	0.004 (0.400)
GRU (9)	0.446 (1.965)	0.571 (2.511)	465.863 (1.705)	721.583 (2.473)	0.004 (0.382)	0.005 (0.449)
Zhang Hybrid (GRU)	0.341 (1.516)	0.398 (1.765)	433.537 (1.635)	695.717 (2.380)	0.004 (0.377)	0.005 (0.444)
New Hybrid (GRU)	0.339 (1.510)	0.393 (1.743)	408.077 (1.484)	661.076 (2.270)	0.003 (0.316)	0.004 (0.410)
Δ%						
New Hybrid (LSTM) vs. ARIMA	−4.883% (−5.051%)	−0.159% (−0.545%)	−5.881% (−5.769%)	−4.829% (−4.695%)	−2.306% (−2.284%)	−3.554% (−3.485%)
New Hybrid (LSTM) vs. LSTM	−4.887% (−4.847%)	−5.134% (−4.967%)	−1.013% (−0.931%)	−0.060% (−0.290%)	−4.745% (−4.724%)	−5.019% (−4.941%)
New Hybrid vs. Zhang Hybrid (LSTM)	−3.131% (−3.700%)	−1.386% (−1.514%)	−6.639% (−9.397%)	−4.882% (−4.789%)	0.512% (0.494%)	−2.308% (−2.224%)
New Hybrid (GRU) vs. ARIMA	−2.409% (−2.560%)	−0.254% (−0.623%)	−0.912% (−1.066%)	−2.440% (−2.117%)	−5.567% (−5.482%)	−1.214% (−1.057%)
New Hybrid (GRU) vs. GRU	−23.939% (−23.291%)	−31.131% (−30.587%)	−12.404% (−12.959%)	−8.385% (−8.218%)	−17.476% (−17.382%)	−8.870% (−8.671%)
New Hybrid vs. Zhang Hybrid (GRU)	−0.546% (−1.060%)	−1.080% (−1.245%)	−5.873% (−9.243%)	−4.979% (−4.657%)	−16.125% (−16.214%)	−7.851% (−7.686%)
Test Statistics (p-value)						
ARIMA		−0.075 (0.940)		0.566 (0.574)		0.597 (0.554)
LSTM		0.870 (0.389)		0.380 (0.705)		0.640 (0.526)
Zhang Hybrid (LSTM)		0.302 (0.764)		0.583 (0.562)		0.578 (0.567)
New Hybrid (LSTM)		0.499 (0.621)		0.473 (0.638)		0.615 (0.542)
GRU		3.413 (0.001)		0.618 (0.539)		1.260 (0.215)
Zhang Hybrid (GRU)		0.233 (0.817)		0.667 (0.508)		1.174 (0.247)
New Hybrid (GRU)		0.577 (0.567)		0.421 (0.675)		0.577 (0.567)
DM Test Statistics (p-value)						
New Hybrid (LSTM) vs. ARIMA		−4.649 (0.999)		−1.823 (0.961)		−3.765 (0.999)
New Hybrid (LSTM) vs. LSTM		−3.289 (0.998)		−1.976 (0.971)		2.311 (0.010)
New Hybrid vs. Zhang Hybrid (LSTM)		−5.542 (0.999)		−0.998 (0.837)		−0.454 (0.673)
New Hybrid (GRU) vs. ARIMA		−4.424 (0.999)		−1.849 (0.962)		−3.260 (0.998)
New Hybrid (GRU) vs. GRU		−5.057 (0.999)		−3.972 (0.999)		2.280 (0.011)
New Hybrid vs. Zhang Hybrid (GRU)		1.900 (0.029)		1.738 (0.041)		2.154 (0.016)

7. Discussion

The empirical results, summarized in Table 4, have shown that the new hybrid model improves the accuracy of the prediction and minimizes both the absolute errors (MAE and RMSE) and relative ones (MAPE and RMSEP). Therefore, the new hybrid model has provided more accurate estimates than the pure ARIMA, ANNs (LSTM or GRU) and the Zhang hybrid model. Furthermore, to prove the goodness of the novel hybrid approach, two types of hypothesis tests were conducted. With the first one, were verified that the differences between true and estimated values were, on average, equal to zero. As expected, no relevant discrepancies were detected between the estimated values and the actual values; therefore, all methodologies have good prediction capabilities. However, the *p*-values associated with the new hybrid model were, in most cases, higher than the ones associated with the pure models. Hence, the new model has improved it ability to forecast errors since it is able to capture linear and non-linear effects, as also highlighted by other authors who proposed hybrid ARIMA-ANN models [7,23–25,39].

Finally, the Diebold–Mariano test was applied to check whether the average difference between errors obtained by using the new hybrid model and those derived from the other competing models was not positive. The prevalence of negative values of the DM statistic indicated that the new hybrid model outperforms the others; moreover, since the null hypothesis was not rejected, the difference in forecasts was, statistically, not positive. However, in the case study examining euro–dollar exchange rates, at a significance level equal to 0.05, the null hypothesis was rejected, when the forecast errors between the novel hybrid model and ANNs were compared. In this case, forecasting using ANN models might be the better choice, since these models are able to capture long-term relationships more effectively, as pointed out in [48,49].

8. Conclusions

In this paper, after a brief theoretical review of the ARIMA and ANN models (i.e., LSTM and GRU), which are widely used for time series prediction purposes, a new hybrid model, which integrates the benefits of the ARIMA model and ANN, was proposed. The new model combines the strengths of both approaches and improves the accuracy of financial time series forecasting: the ARIMA component of the model is effective in capturing linear temporal dependencies and trends in the data, while the ANN component is capable of modeling complex, non-linear patterns. By integrating these models, the new hybrid approach is capable of identifying linear and non-linear effects. Three case studies on financial time series (i.e., Unicredit SpA stock price, Bitcoin closing price, EUR/USD exchange rate) were presented, and the steps for the parameter estimates of the pure ARIMA, pure ANNs (i.e., LSTM and GRU) and hybrid model were discussed. After the validation phase, out-of-sample predictions for the period from the 27th of September to the 26th of October 2023 were estimated, and true values were used to calculate error metrics and assess the reliability of the fitted models. Furthermore, this study compared the results of the proposed model with those obtained from single approach models and the hybrid model of Zhang [7].

It is worth highlighting that the convenience in using the proposed hybrid model depends on the impact of linear and non-linear characteristics on the behavior of the variable under study. Although the new hybrid model produced appreciable results on daily data in the financial context, further research could be devoted to the application of this model to different frequencies of data (intra-day, weekly) or other financial instruments in order to test the performance of the new hybrid model constructed in this study. Future studies might should also consider focusing their attention on improving the combination of models and parameter estimation. Moreover, further advances might regard the computational aspects and consequent implementation of appropriate routines useful for applying this model to other contexts.

Author Contributions: Conceptualization, S.D.I.; Methodology, C.C. and A.C.; Software, A.C.; Validation, C.C.; Formal analysis, C.C. and A.C.; Writing—original draft, C.C. and A.C.; Supervision, S.D.I. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data derived from public domain resources.

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

References

1. Jin, X.H.; Zhang, G. Modelling optimal risk allocation in PPP projects using artificial neural networks. *Int. J. Proj. Manag.* **2011**, *29*, 591–603. [CrossRef]
2. Wong, L.W.; Tan, G.W.H.; Ooi, K.B.; Lin, B.; Dwivedi, Y.K. Artificial intelligence-driven risk management for enhancing supply chain agility: A deep-learning-based dual-stage PLS-SEM-ANN analysis. *Int. J. Prod. Res.* **2024**, *62*, 5535–5555. [CrossRef]
3. Gottschlich, J.; Hinz, O. A decision support system for stock investment recommendations using collective wisdom. *Decis. Support Syst.* **2014**, *59*, 52–62. [CrossRef]
4. Fox, S.; McDermott, J.; Doherty, E.; Cooney, R.; Clifford, E. Application of neural networks and regression modelling to enable environmental regulatory compliance and energy optimisation in a sequencing batch reactor. *Sustainability* **2022**, *14*, 4098. [CrossRef]
5. Box, G.; Jenkins, G.M. *Time Series Analysis*; Holden-Day: San Francisco, CA, USA, 1970.
6. Zhang, G.; Eddy Patuwo, B.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [CrossRef]
7. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **2003**, *50*, 159–175. [CrossRef]
8. Granger, C.W.J.; Andersen, A. Non-linear time series modelling. In *Applied Time Series Analysis I*; Elsevier: Amsterdam, The Netherlands, 1978; pp. 25–38.
9. Granger, C. New classes of time series models. *J. R. Stat. Society. Ser. D The Stat.* **1978**, *27*, 237–253. [CrossRef]
10. Tong, H. *Threshold Models in Non-Linear Time Series Analysis*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 21.
11. Engle, R.F. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econom. J. Econom. Soc.* 1982, *50*, 987–1007. [CrossRef]
12. De Gooijer, J.G.; Kumar, K. Some recent developments in non-linear time series modelling, testing, and forecasting. *Int. J. Forecast.* **1992**, *8*, 135–156. [CrossRef]
13. Tealab, A. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Comput. Inform. J.* **2018**, *3*, 334–340. [CrossRef]
14. Tealab, A.; Hefny, H.; Badr, A. Forecasting of nonlinear time series using ANN. *Future Comput. Inform. J.* **2017**, *2*, 39–47. [CrossRef]
15. Ghiassi, M.; Saidane, H. A dynamic architecture for artificial neural networks. *Neurocomputing* **2005**, *63*, 397–413. [CrossRef]
16. Khashei, M.; Bijari, M. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Syst. Appl.* **2010**, *37*, 479–489. [CrossRef]
17. Graves, A.; Mohamed, A.-r.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 6645–6649.
18. Yu, Y.; Si, X.; Hu, C.; Zhang, J. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [CrossRef] [PubMed]
19. Gogas, P.; Papadimitriou, T. Machine learning in economics and finance. *Comput. Econ.* **2021**, *57*, 1–4. [CrossRef]
20. Masini, R.P.; Medeiros, M.C.; Mendes, E.F. Machine learning advances for time series forecasting. *J. Econ. Surv.* **2023**, *37*, 76–111. [CrossRef]
21. Zheng, H.; Wu, J.; Song, R.; Guo, L.; Xu, Z. Predicting financial enterprise stocks and economic data trends using machine learning time series analysis. *Appl. Comput. Eng.* **2024**, *87*, 26–32. [CrossRef]
22. Yan, K.; Li, Y. Machine learning-based analysis of volatility quantitative investment strategies for American financial stocks. *Quant. Financ. Econ.* **2024**, *8*, 364–386. [CrossRef]
23. Khandelwal, I.; Adhikari, R.; Verma, G. Time series forecasting using hybrid ARIMA and ANN models based on DWT decomposition. *Procedia Comput. Sci.* **2015**, *48*, 173–179. [CrossRef]
24. Wang, L.; Zou, H.; Su, J.; Li, L.; Chaudhry, S. An ARIMA-ANN hybrid model for time series forecasting. *Syst. Res. Behav. Sci.* **2013**, *30*, 244–259. [CrossRef]
25. Hajirahimi, Z.; Khashei, M. A novel parallel hybrid model based on series hybrid models of ARIMA and ANN models. *Neural Process. Lett.* **2022**, *54*, 2319–2337. [CrossRef]
26. Ma, C.; Wu, J.; Hu, H.; Chen, Y.N.; Li, J.Y. Predicting Stock Prices Using Hybrid LSTM and ARIMA Model. *IAENG Int. J. Appl. Math.* **2024**, *54*, 424–432.

27. Pan, H.; Tang, Y.; Wang, G. A Stock Index Futures Price Prediction Approach Based on the MULTI-GARCH-LSTM Mixed Model. *Mathematics* **2024**, *12*, 1677. [CrossRef]
28. Ghalayini, L. Modeling and forecasting the US dollar/euro exchange rate. *Int. J. Econ. Financ.* **2014**, *6*, 194–207. [CrossRef]
29. Baciu, I.C. Stochastic models for forecasting inflation rate. Empirical evidence from Romania. *Procedia Econ. Financ.* **2015**, *20*, 44–52. [CrossRef]
30. Khan, S.; Alghulaiakh, H. ARIMA model for accurate time series stocks forecasting. *Int. J. Adv. Comput. Sci. Appl.* **2020**, *11*. [CrossRef]
31. Huang, Z.; Zheng, H.; Li, C.; Che, C. Application of Machine Learning-Based K-means Clustering for Financial Fraud Detection. *Acad. J. Sci. Technol.* **2024**, *10*, 33–39. [CrossRef]
32. Habbab, F.Z.; Kampouridis, M. An in-depth investigation of five machine learning algorithms for optimizing mixed-asset portfolios including REITs. *Expert Syst. Appl.* **2024**, *235*, 121102. [CrossRef]
33. Dumitrescu, E.; Hué, S.; Hurlin, C.; Tokpavi, S. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *Eur. J. Oper. Res.* **2022**, *297*, 1178–1192. [CrossRef]
34. Pirani, M.; Thakkar, P.; Jivrani, P.; Bohara, M.H.; Garg, D. A Comparative Analysis of ARIMA, GRU, LSTM and BiLSTM on Financial Time Series Forecasting. In Proceedings of the 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 23–24 April 2022; pp. 1–6.
35. Sako, K.; Mpinda, B.N.; Rodrigues, P.C. Neural Networks for Financial Time Series Forecasting. *Entropy* **2022**, *24*, 657. [CrossRef]
36. Das, J.D.; Thulasiram, R.K.; Henry, C.; Thavaneswaran, A. Encoder–Decoder Based LSTM and GRU Architectures for Stocks and Cryptocurrency Prediction. *J. Risk Financ. Manag.* **2024**, *17*, 200. [CrossRef]
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
38. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2017**, *10*, 841–851. [CrossRef]
39. Liu, X.; Lin, Z.; Feng, Z. Short-term offshore wind speed forecast by seasonal ARIMA-A comparison against GRU and LSTM. *Energy* **2021**, *227*, 120492. [CrossRef]
40. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU neural network methods for traffic flow prediction. In Proceedings of the 2016 31st Youth academic annual conference of Chinese association of automation (YAC), Wuhan, China, 11–13 November 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 324–328.
41. Yu, T.; Zhu, H. Hyper-parameter optimization: A review of algorithms and applications. *arXiv* **2020**, arXiv:2003.05689.
42. Tuba, E.; Bačanin, N.; Strumberger, I.; Tuba, M. Convolutional Neural Networks Hyperparameters Tuning. In *Artificial Intelligence: Theory and Applications*; Springer: Cham, Switzerland, 2021; pp. 65–84.
43. Abdolrasol, M.G.; Hussain, S.S.; Ustun, T.S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Mekhilef, S.; Milad, A. Artificial neural networks based optimization techniques: A review. *Electronics* **2021**, *10*, 2689. [CrossRef]
44. Prechelt, L. Early stopping-but when? In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 55–69.
45. Bai, Y.; Yang, E.; Han, B.; Yang, Y.; Li, J.; Mao, Y.; Niu, G.; Liu, T. Understanding and improving early stopping for learning with noisy labels. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24392–24403.
46. Diebold, F.X.; Mariano, R.S. Comparing predictive accuracy. *J. Bus. Econ. Stat.* **1995**, *13*, 253–263. [CrossRef]
47. Alt, W.; Malanowski, K. The Lagrange–Newton method for nonlinear optimal control problems. *Comput. Optim. Appl.* **1993**, *2*, 77–100. [CrossRef]
48. Galeshchuk, S. Neural networks performance in exchange rate prediction. *Neurocomputing* **2016**, *172*, 446–452. [CrossRef]
49. Panda, M.M.; Panda, S.N.; Pattnaik, P.K. Exchange rate prediction using ANN and deep learning methodologies: A systematic review. In Proceedings of the 2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), Rajpura, India, 7–15 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 86–90.