applied sciences

MDPI

Article

Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction

Da Huang †, Zhaoguo Liu *,† and Dan Wu

Institute for Quantum Information & State Key Laboratory of High Performance Computing, College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China; huangda1109@163.com (D.H.); daisydanwu@nudt.edu.cn (D.W.)

* Correspondence: liuzhaoguo21@nudt.edu.cn

† These authors contributed equally to this work.

Abstract: Feature selection has perennially stood as a pivotal concern in the realm of time-series forecasting due to its direct influence on the efficacy of predictive models. Conventional approaches to feature selection predominantly rely on domain knowledge and experiential insights and are, therefore, susceptible to individual subjectivity and the resultant inconsistencies in the outcomes. Particularly in domains such as financial markets, and within datasets comprising time-series information, an abundance of features adds complexity, necessitating adept handling of high-dimensional data. The computational expenses associated with traditional methodologies in managing such data dimensions, coupled with vulnerability to the curse of dimensionality, further compound the challenges at hand. In response to these challenges, this paper advocates for an innovative approach—a feature selection method grounded in ensemble learning. The paper explicitly delineates the formal integration of ensemble learning into feature selection, guided by the overarching principle of "good but different". To operationalize this concept, five feature selection methods that are well suited to ensemble learning were identified, and their respective weights were determined through K-fold cross-validation when applied to specific datasets. This ensemble method amalgamates the outcomes of diverse feature selection techniques into a numeric composite, thereby mitigating potential biases inherent in traditional methods and elevating the precision and comprehensiveness of feature selection. Consequently, this method improves the performance of time-series prediction models.

Keywords: ensemble learning; feature selection; time-series prediction

Citation: Huang, D.; Liu, Z.; Wu, D. Research on Ensemble Learning-Based Feature Selection Method for Time-Series Prediction. *Appl. Sci.* **2024**, *14*, 40. https://doi.org/10.3390/app14010040

Academic Editors: Carlos J. Costa and Manuela Aparicio

Received: 21 November 2023
Revised: 16 December 2023
Accepted: 18 December 2023
Published: 20 December 2023

1. Implication and Contributions of the Research

Time-series data prediction has perennially occupied a position of paramount importance, manifesting widespread applications across diverse domains, such as finance, weather forecasting, traffic planning, and sales forecasting. In the financial sector, the prediction of time-series data serves as the bedrock for investment decisions. Investors heavily lean on precise forecasts of time-series data, encompassing variables like stock prices, exchange rates, and commodity prices to formulate astute buying and selling strategies. Concurrently, governmental and regulatory bodies also utilize accurate time-series data to vigilantly monitor the stability of the financial markets. In meteorology, the prediction of time-series data assumes critical significance in the accurate forecasting of weather patterns, climate changes, and the onset of natural disasters. Within the domain of traffic planning, the prediction of traffic flow is instrumental in aiding urban planners to adeptly manage traffic congestion and enhance overall traffic efficiency. Equally vital is sales forecasting, which empowers retailers to ascertain optimal inventory requirements, ensuring timely product supply while minimizing inventory costs.

The challenges associated with feature selection in time-series data primarily stem from the unique characteristics of such data. Time-series data exhibit temporal correlations, sequence patterns, and seasonality, often making traditional feature selection methods less

suitable. Dealing with time-series data requires consideration of various complex factors, including the influence of historical information on current observations, the extraction of temporal features, and addressing issues such as missing values and noise. Consequently, developing feature selection methods tailored for time-series data is a crucial research area with the goal of constructing more precise, efficient, and interpretable time-series prediction models.

Against this backdrop, this paper introduces a pioneering innovation designed to improve the efficacy of feature selection in time-series data prediction. We leverage an ensemble learning approach that seamlessly integrates five distinct feature selection methods. This integrative framework aims to redress the limitations inherent in various feature selection methods, consequently yielding more precise and resilient feature selection outcomes. The chosen five feature selection methods span diverse dimensions, each endowed with its distinct strengths. Through the amalgamation of these methods, our objective was to comprehensively capture information and patterns within time-series data, thereby enhancing the performance and resilience of the prediction model.

The specific innovations and contributions of this paper are outlined as follows:

1. This paper introduces a feature selection method grounded in ensemble learning. It furnishes a formal definition of the application of ensemble learning to feature selection and, adhering to the principle of "good but different", identifies five feature selection methods for integration. The weights of these methods were determined through K-fold cross-validation when applied to specific datasets. This ensemble approach considers the outcomes of multiple feature selection methods, consolidating them into a numerical outcome. This process aids in mitigating potential biases in traditional methods, thereby enhancing the accuracy and comprehensiveness of feature selection;
2. This paper deploys an LSTM model, incorporating features selected through ensemble learning and those identified by five different feature selection methods as inputs to the model. A series of experiments was conducted to validate the effectiveness of this approach. Through the practical applications in time-series prediction tasks, the paper presents concrete data and results to demonstrate the performance and efficacy of the proposed feature selection method in real-world scenarios.

By introducing ensemble learning into the field of feature selection, this study broadens the application scope of ensemble learning and empirically supports its effectiveness in feature selection tasks. We conducted robust comparisons with traditional methods, highlighting the innovative aspects of ensemble learning in enhancing both the accuracy and comprehensiveness of feature selection. Adopting an ensemble learning approach, our research offers a more comprehensive consideration of the strengths of various methods compared to traditional single-method approaches. The experimental results on different datasets demonstrate the superior performance of ensemble learning in terms of MAE and MSE metrics, validating its theoretical value in enhancing the robustness and effectiveness of feature selection.

2. Background

Time-series prediction has always been a daunting challenge. Scholars in this domain have diligently delved into the intrinsic laws governing time-series data through extensive exploration and research. Consequently, they have amassed a considerable repertoire of prediction methods based on the evolving patterns of this data, broadly categorized into linear and nonlinear prediction methods.

During the initial stages, prediction methods heavily leaned towards linear approaches, employing classic algorithms like the exponential smoothing method [1,2] and the autoregressive integral moving average prediction method [3–5]. These methods boasted advantages such as simplicity, reduced computational demands, and superior performance in short-term predictions. However, they fell short in capturing the inherent nonlinear

relationships within financial time-series data, particularly when tackling long-term predictions. Consequently, they exhibited certain limitations in such scenarios.

To overcome the limitations of linear methods, subsequent research proposed the integration of nonlinear models to enhance the comprehension of complex data, thereby giving rise to nonlinear prediction methods. Prominent among these methods are BP neural networks [6–8], support vector machines, recurrent neural networks [9–11], generative adversarial networks [12,13], and reinforcement learning [14–16]. By employing these methodologies, researchers have achieved a more comprehensive capture of the nonlinear relationships embedded in financial time-series data, leading to relatively accurate prediction outcomes. This direction represents the focal point of future research and the prevailing trend in the field of financial time-series data.

In recent years, deep learning has attracted considerable attention from researchers in various fields. Deep learning methods have demonstrated remarkable performance when compared to traditional algorithms in time-series prediction tasks, which have undergone extensive development and widespread application. Of particular note, deep neural networks possess superior capabilities in extracting both linear and nonlinear features, outperforming shallow neural networks in this regard. This advantage enables them to capture underlying patterns that may be overlooked by their shallower counterparts, making them well suited for high-precision prediction tasks [17]. In light of these advancements, this section aims to introduce three primary categories of deep learning models that are particularly suitable for addressing challenges in time-series forecasting.

Convolutional Neural Networks (CNNs) represent a class of deep feed-forward neural networks that center around convolution and pooling operations. Originally developed for image recognition in the domain of computer vision [18,19], CNNs have since demonstrated their versatility in various fields. In 2018, Shaojie Bai et al. [20] proposed an innovative architecture called Temporal Convolutional Networks (TCNs), a variant of CNNs designed with reduced memory consumption and increased parallelizability. TCNs introduced causal convolution to ensure that future information is not accessed during training, thus mitigating issues related to gradient vanishing and gradient explosion. Additionally, the backpropagation path in TCNs differs from the temporal direction, providing added stability. To address the problem of information loss caused by an excessive number of layers in CNNs, TCNs incorporate residual connectivity, facilitating seamless information transfer across layers within the network.

Recurrent Neural Networks (RNNs) are a form of deep learning model introduced by M. I. Jordan in 1990 specifically designed to capture time-dimensional features. Later, in 1997, Mike Schuster et al. [21] extended the RNN architecture, leading to the creation of Bidirectional Recurrent Neural Networks (Bi-RNNs).

To address some limitations of RNN models, Hochreiter proposed Long Short-term Memory (LSTM) in 1997 [22]. Subsequently, in 2005, A. Graves et al. [23] further expanded LSTM to create Bidirectional Long Short-term Memory (BiLSTM). The structure of BiLSTM closely resembles that of Bi-RNN, incorporating two independent LSTM units concatenated together. By doing so, the BiLSTM model effectively addresses the limitation of LSTM's inability to incorporate future information, enabling feature data obtained at time t to encompass both past and future information [24].

Vaswani et al. [25] introduced the Transformer in 2017 as an innovative deep learning framework, distinct from the conventional structures of CNNs or RNNs. The Transformer relies entirely on the attentional mechanism to capture global dependencies between model inputs and outputs. This remarkable ability to handle long-term dependencies and interactions renders the Transformer well suited for time-series modeling tasks, leading to high performance in various time series-related endeavors [26].

To address specific limitations of the Transformer in long time-series prediction, Haoyi Zhou et al. [27] proposed the Informer model in 2021. Building upon the classical Transformer encoder–decoder structure, the Informer model aims to tackle challenges encountered in long time-series prediction tasks. In the same year, Lim B et al. [28] presented

Temporal Fusion Transformers (TFTs), implementing a multiscale prediction model with a static covariate encoder, a gated feature selection module, and a temporally self-attentive decoder. TFTs not only deliver accurate predictions but also retain interpretability, considering global, temporal dependencies, and events.

3. Feature Selection Based on Ensemble Learning

The incorporation of ensemble learning in feature selection endeavors to improve the robustness, comprehensiveness, and stability of models, thereby mitigating the risk of overfitting and enhancing predictive accuracy. These advantages position ensemble learning as a potent tool in time-series prediction tasks, thereby contributing significantly to the enhancement of model performance and reliability.

Given a dataset D, consisting of m samples and p features, we propose a set of feature selection methods $M_1, M_2, \ldots, M_n$. Each method M_i employs a specific feature selection rule, assigning a score s_{ij} to each feature, where i denotes the index of the method and j represents the index of the feature.

For each method M_i, a binary function $f_i(X, Y)$ can be defined, where X represents the original feature set and Y represents the target variable. This function returns a subset containing the selected features, denoted as S_i. The specific definition is as follows:

$$S_i = f_i(X, Y) \tag{1}$$

Each S_i is a subset of the original feature set X, representing the features selected by method M_i. Combining the selection results of all methods forms a feature selection set $\{S_1, S_2, \ldots, S_n\}$.

The objective of ensemble learning methods is to combine these individual methods to select the final feature subset S, maximizing a performance metric P, typically representing the performance of a predictive model. The formal expression is as follows:

$$S_{\text{final}} = \arg\max_{P} \sum_{i=1}^{n} w_i P_i\left(s_{i1}, s_{i2}, \ldots, s_{ip}\right) \tag{2}$$

where S_{final} represents the final feature subset, i.e., the indices of the selected features. P is the performance metric function, which could be accuracy, mean squared error, etc., measuring the model's performance. P_i is the performance metric function for the i-th feature selection method, assessing the performance of the feature subset based on the scores s_{ij}. w_i is the weight used to balance different methods and can be determined based on the performance of each method.

This formula expresses that the objective of ensemble learning is to optimize the weights of features to select the final feature subset S_{final}, aiming to achieve the best performance metric P. The optimization of weights can be realized through the combination and adjustment of various methods, tailored to meet the requirements of the problem and the characteristics of the data.

Therefore, to obtain a feature selection subset that maximally enhances the accuracy of a time-series prediction model, i.e., maximizing the performance metric function p for S_{final}, we need to evaluate the weights λ_i for each feature selection method and calculate the importance scores s_{ij} for each feature. For the former, we must determine an appropriate method for weight calculation that comprehensively considers the varying performances of different feature selection methods when dealing with diverse types of datasets. Regarding the latter, we need to establish a set of alternative feature selection methods $\{M_1, M_2, \ldots, M_n\}$, covering diverse feature selection strategies to meet the requirements of different scenarios. For each method M_i, performance metrics are employed to assess its effectiveness, resulting in feature importance scores S_i. Finally, adopting a feature scoring weighting approach using S_i, we generate the ultimate feature subset S_{final}. The entire ensemble learning-based feature selection process is illustrated in Figure 1.

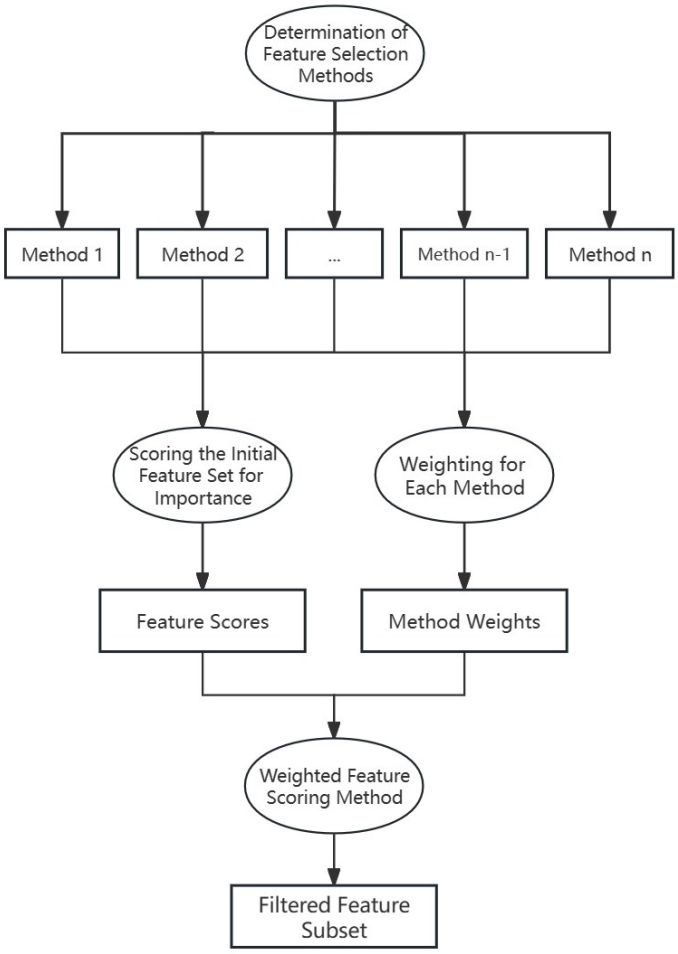

Figure 1. Flowchart of feature selection based on ensemble learning.

3.1. Determination of Ensemble Learning Strategies

Given the set of feature selection methods $\{M_1, M_2, \ldots, M_n\}$, an appropriate ensemble learning strategy is needed to obtain the final postfiltered feature subset S_{final} . To integrate information from multiple feature selection methods, mitigate dependence on a single method, and enhance overall robustness, this study adopts the feature scoring weighting approach as its ensemble learning strategy. The feature scoring weighting approach exhibits comprehensive, flexible, and performance-enhancing characteristics across various problems and data contexts. This method facilitates the amalgamation of information from multiple feature selection methods, thereby improving model performance and robustness, diminishing the risk of overfitting, all while upholding a certain level of interpretability. The specific implementation steps are outlined as follows:

Firstly, through each feature selection method M_i, the score $S_{i,j} = f_{i,j}(X, Y)$ for each feature is calculated according to its respective feature scoring strategy. Here, $s_{i,j}$ represents the score assigned by method M_i to feature j.

To ensure comparability among importance scores under different feature selection methods, it is imperative to constrain the scores of features within the range of 0 to 1. Therefore, normalization of the feature scores provided by each method is requisite. This paper employs a softmax transformation for the normalization of importance scores.

In this way, the raw feature score s_i is transformed into a probability representing each feature, ensuring that they range between 0 and 1, with a total sum of 1. These normalized scores can be utilized as comparable values across different feature selection methods.

Different feature selection methods may have varying applicability in different domains and for different types of time-series data. In order to comprehensively consider the scores from multiple methods, this paper introduces a weight vector W = $[w_1, w_2, \ldots, w_n]$,

where w_i represents the weight assigned to method M_i. These weights are utilized to adjust the relative contributions of each method.

This paper employs the K-fold cross-validation method to determine the weights for each feature selection method. This method divides the dataset into K folds and performs the following steps for each fold: First, select $K-1$ folds as the training set, and reserve one fold as the test set. Then, apply each feature selection method M_i on the training set, obtaining the feature selection results S_i for each method. Next, train the regression model using the filtered feature subset and evaluate the model's performance on the test set. Mean Squared Error (MSE) is used as the performance metric for each method and is recorded. For each feature selection method M_i, based on the performance metric results from K-fold cross-validation, calculate the average performance metric avg_i. These average performance metrics serve as indicators for the weights w_i. The weights w_i are then normalized based on the average performance metrics to ensure that they sum up to 1, i.e.,

$$w_i = \frac{avg_i}{\sum_{i=1}^{5} avg_i} \tag{3}$$

Finally, the learned weights w_i are utilized in the output of the feature scoring weighting method, forming the ultimate feature selection results:

$$S_{\text{final}} = \sum_{i=1}^{n} w_i \cdot S_i \tag{4}$$

where S_{final} represents the ultimate feature subset, which is the weighted sum of scores from each method. The weight vector W governs the relative importance of each method. The final feature subset S_{final} incorporates information from multiple method scores, representing the features ultimately selected.

3.2. Determination of Feature Selection Methods

To establish a suitable set of feature selection methods $\{M_1, M_2, \ldots, M_N\}$, this paper introduces a "good but different" principle. According to this principle, individual learners should contribute to performance and exhibit differences among themselves. This ensures mutual complementarity and deficiency compensation during the ensemble process, ultimately enhancing the overall performance.

When applying ensemble learning to feature selection, it is equally crucial to ensure that the feature selection methods meet the "good but different" criteria. This implies that they should exhibit diversity, independence, stability, reliability, efficiency, adaptability, and robustness. Adhering to these requirements is essential in guaranteeing that feature selection methods can offer robust support for ensemble learning, thereby improving the performance and reliability of the model.

Grounded in the principles outlined above, this paper selects the Pearson correlation coefficient method, recursive feature elimination method, random forest method, gradient boosting decision tree, and XGBoost algorithm as the foundational feature selection methods. The specific rationale was as follows:

The Pearson correlation coefficient method (Pearson) is a statistical approach employed to measure the linear correlation between two continuous variables. It was deemed suitable as a feature selection method owing to its capability to identify the strength and direction of the linear relationship between features and the target variable. This method proves particularly valuable for features that manifest a linear relationship with the target variable. Pearson correlation coefficient is calculated using the following formula:

$$\mathbf{r} = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}} \tag{5}$$

where r represents the Pearson correlation coefficient, X_i and Y_i denote the i-th observation of the two variables, and $\bar{X}$ and $\bar{Y}$ represent the means of the two variables, respectively.

Recursive Feature Elimination (RFE) is a stepwise feature selection method that identifies the most informative feature subset by iteratively training models and eliminating the least important features. It is well suited as a feature selection method due to its ability to automatically identify and select important features, mitigate overfitting, enhance model interpretability, and the fact that it requires minimal manual intervention. Let X represent the feature set, and M be the metric used to evaluate the performance of the model. The process of Recursive Feature Elimination (RFE) can be expressed as follows:

$$\mathrm{RFE}(X) = \begin{cases} X & \text{if } |X| \leq \text{desired\_num\_features} \\ \mathrm{RFE}(\text{remove\_least\_important\_feature}(X)) & \text{otherwise} \end{cases} \tag{6}$$

where remove_least_important_feature(X) denotes the operation of removing the least influential feature in terms of model performance from the feature set X.

The Random Forest method (RF) is an ensemble learning approach that improves overall performance by aggregating predictions from multiple decision trees. It is well suited as a feature selection method due to its capability to estimate feature importance and utilize a voting method for classification problems. Random forests exhibit a notable level of robustness against outliers and noise, rendering them suitable for addressing complex data scenarios. The importance calculation for feature X_i is as follows:

$$\mathrm{Importance}(X_i) = \frac{1}{N} \sum_{j=1}^{N} \mathrm{Impurity\_Decrease}_j(X_i) \tag{7}$$

where N is the number of decision trees in the random forest, and $\mathrm{Impurity\_Decrease}_j(X_i)$ represents the decrease in impurity in the j-th decision tree due to the introduction of the feature X_i.

Gradient Boosting Decision Trees (GBDTs) are ensemble learning algorithms that amalgamate the principles of decision trees and gradient boosting, consistently enhancing model performance through iterative training of multiple decision trees. They are well suited as feature selection methods owing to their ability to automatically estimate the importance of features, demonstrate adaptability to high-dimensional and large-scale datasets, and furnish insights and understanding of the data. The update rule for gradient boosting decision trees is as follows:

$$F_m(x) = F_{m-1}(x) + \gamma \cdot h_m(x) \tag{8}$$

where $F_m(x)$ represents the model's prediction at the m-th round, γ is the learning rate, and $h_m(x)$ is the decision tree fitted in the m-th iteration.

XGBoost is an efficient, flexible, and scalable machine learning algorithm based on the gradient boosting framework. It consistently enhances predictive performance through the iterative training of multiple decision tree models. In comparison to traditional gradient boosting methods, XGBoost introduces additional regularization terms and tree depth limitations, thereby improving model stability and generalization. It is well suited as a feature selection method due to its notable advantages in both performance and robustness, while also aiding in mitigating the risk of overfitting. The objective function of XGBoost comprises the loss function, regularization term, and model complexity term. For regression problems, the objective function is as follows:

$$\mathrm{Obj} = \sum_{i=1}^{n} L(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{9}$$

where L is the loss function, Ω is the regularization term, and f_k represents the k-th decision tree model.

The five methods mentioned above each exhibit distinct characteristics in feature selection and all adhere to the "good but different" principle. From the linear relationship measurement of the Pearson correlation coefficient method to the iterative feature elimination of the recursive feature elimination method, and from the robust performance of the random forest and gradient boosting decision tree in handling outliers and noise to the efficiency and versatility of XGBoost, each method is guided by the principle of being "good but different". The goal was to select features that contribute significantly while possessing unique advantages to adapt to the varied requirements of different problems and datasets.

By integrating these five diverse feature selection methods, this paper aims to derive more powerful, robust, and comprehensive feature selection results, thereby contributing to the construction of enhanced time-series prediction models.

4. Experimental Results and Analysis of Feature Selection Based on Ensemble Learning

4.1. Experimental Preparation

4.1.1. Initial Feature Set

To better illustrate the performance enhancement achieved by the ensemble learning-based feature selection method for time-series data prediction models, this paper utilizes financial time-series data to construct the initial feature set for subsequent experiments. Financial time-series data typically encompass multiple features such as stock prices, trading volume, market capitalization, financial indicators, etc. Financial time-series data are characterized by their richness in information, high dimensionality, temporal nature, and real-time aspects. These characteristics render them well suited for applying ensemble learning methods to feature selection, thereby improving model performance, mitigating risks, and fostering a more profound understanding of financial markets.

In theory, a broader range of features in the initial feature set implies a more comprehensive coverage of information, leading to more accurate predictive results for the model. It is imperative to choose features that are highly correlated with stock prices or returns to construct the initial feature set for the prediction model. This paper proposes selecting a total of 20 candidate features from three major categories: market-related, trading-related, and market capitalization-related features, with the objective of comprehensively considering various types of features.

The present study incorporates a set of market-related features, namely, high, open, low, pre_close, pct_change, change, and avg_price, totaling seven features. These features denote the daily stock trading metrics of highest price, opening price, lowest price, previous day's closing price, percentage change, and stock price change, respectively. Market-related features can reflect short-term fluctuations and trends in the market, aiding in capturing instantaneous changes in stock prices.

For trading-related features, we selected vol_ratio, vol, turn_over, amount, selling, and buying, constituting six features. These features represent volume ratio, trading volume, turnover rate, transaction amount, selling transactions, and buying transactions. Analyzing these trading-related features can provide insights into market activity, trading volume, price fluctuations, and other aspects, contributing to a better understanding of the overall market conditions.

In terms of market value-related features, the study includes pe, float_mv, total_mv, swing, activity, strength, and attack, totaling seven features. These features signify price-to-earnings ratio, float market value, total market value, amplitude, activity level, strength, and market aggressiveness, respectively. Changes in stock market value may be correlated with investor sentiment, market cycles, and other factors, making market value-related features crucial for predicting stock returns.

By comprehensively considering these three major categories of features, the initial feature set can encompass information from different aspects, thereby enhancing the predictive accuracy of the model. Table 1 displays the initial feature set selected from these three major categories.

Table 1. Initial feature set.

Categories of Features	Specific Features
Market Features	high, open, low, pre_close, pct_change, change, avg_price
Trading Features	vol_ratio, vol, turn_over amount, selling, buying
Market Value Features	pe, float_mv, total_mv, swing activity, strength, attack

4.1.2. Experimental Dataset

In terms of dataset selection, to ensure diversity and representativeness, this paper chose stock time-series data from three distinct industries: finance, power, and technology. Specifically, the stock data from three datasets, namely, China Industrial and Commercial Bank (ICBC), GD Power Development (GD Power), and China Unicom, were selected. This approach enabled us to comprehensively explore and evaluate the applicability of ensemble learning-based feature selection across different industry datasets.

Each stock dataset encompasses a total of 716 trading days, spanning from 1 January 2020 to 1 January 2023. Specifically, each dataset is composed of 20 features from the initial feature set in Table 1, along with the closing prices used for prediction, forming a 716×21 dimensional data matrix. The datasets were sequentially split into training and testing sets in an 8:2 ratio. Descriptive statistics for the closing prices in each dataset, including mean, standard deviation, minimum, quartiles, and maximum values, are presented in Table 2.

Table 2. Descriptive statistics for each dataset.

Statistical Metrics	ICBC	GD Power	China Unicom
count	716	716	716
mean	4.87	2.72	4.35
std	0.38	0.80	0.68
min	4.06	1.80	3.31
25%	4.63	2.11	3.81
50%	4.85	2.42	4.26
75%	5.15	3.02	4.92
max	6.01	4.94	6.18

4.1.3. Data Preprocessing

To ensure the quality of the subsequent feature selection and model construction, data preprocessing is an indispensable step. In this paper, three specific data preprocessing methods were employed, namely, missing value handling, outlier treatment, and standardization. This series of preprocessing steps significantly enhanced the data quality, reduced potential sources of errors in the models, facilitated a better understanding and interpretation of the data, and also led to a reduction in the computational burden, thereby improving computational efficiency. The specific procedures are outlined below.

Stock feature data may exhibit issues such as missing values, format errors, or precision discrepancies due to network problems, time periods, or the absence of original data. Typically, it is necessary to address missing values, which are often represented as NaN or other placeholders and can be detected by examining the dataset. In this study, forward filling was employed to address missing values, where previous values are used to fill the gaps, preserving the continuity of the time series.

Individual feature data may contain exceptionally large deviations, which can impact the standard deviation of the data and even lead to the distortion of the overall dataset. To address this issue, this paper employed the Median Absolute Deviation (MAD) method. For each feature, the median and MAD were calculated, where the median was the middle

value, and MAD was the median of the absolute differences between each data point and the median. Outliers are typically defined as data points deviating from the median by a certain extent, and in this study, the threshold for identifying outliers was set at three times the MAD. Subsequently, each data point for each feature was examined, comparing the absolute difference between each data point and the median with the threshold for outliers. If the absolute difference was greater than the threshold, the data point was flagged as an outlier. Finally, each identified outlier was set to the median to mitigate its impact on the data. Removing outliers contributes to enhancing the robustness and accuracy of the model.

Stock feature data may exhibit differences in magnitude, necessitating standardization to ensure comparability among different features. This paper employed the z-score normalization method due to its simplicity in calculation and its suitability for data approaching a normal distribution. This standardization method helps ensure that the magnitudes of different features do not adversely affect the interpretability of the model and facilitates the exploration of relationships between features and stock price trends. By transforming the data into a standard normal distribution with a mean of 0 and a standard deviation of 1, the values of different features share the same scale, making them suitable for comparison and modeling.

4.2. Experimental Results and Analysis

Next, we employed the five feature selection methods determined in Section 3.2 to assess the importance scores of each feature in the initial feature set for different stocks. The results are presented in Figure 2.

Two main observations are clearly evident from the graph. Firstly, different feature selection methods exhibited significant variations when analyzing the same stock data. Using ICBC as an example, various methods assigned relatively high scores to the "low" feature, but there were substantial score differences for the "pe", "float_mv", and "avg_price" features among different methods. This indicates a noteworthy variability in the impact of stock features under different feature selection methods.

Secondly, concerning stock data from different industry sectors, there were notable variations in the importance score distributions for each feature. Using the GD Power dataset as an example, the "low" and "avg_price" features received relatively high scores across various evaluation methods, while other features had comparatively lower scores. In contrast, in the dataset for ICBC, features such as "float_mv" and "pe" obtained higher scores. This emphasizes the distinct importance of various features in different industry sectors.

Therefore, to fully leverage the advantages of various feature selection methods on different data types, it is essential to employ ensemble learning methods to integrate the scores from different methods. Through the combination of scores from different methods, a more comprehensive consideration of the importance of different features under diverse data contexts can be achieved, thereby enhancing the model's robustness and performance.

After obtaining the importance scores for each feature, weights for each feature selection method were calculated using the K-fold cross-validation method, as outlined in Section 3.2. In this study, K was set to 5, indicating the use of 5-fold cross-validation. The conclusive results are presented in Table 3.

According to the data in Table 2, it is evident that the same feature selection method carries different weights across various types of stock data. Higher weights indicate that the features selected by that method exhibit superior predictive performance in the corresponding stock data. Consequently, the method is more suitable for this type of data.

By assigning distinct weights to these methods, ensemble learning can select the most suitable feature selection method for each time-series data context and conduct comprehensive screening. Ultimately, this approach can achieve superior predictive performance and higher robustness when facing diverse types of time-series data requirements.

After obtaining the importance scores $s_{i,j}$ for each feature and the weights w_i for each feature selection method, the feature score weighting method was applied to calculate the top five ranked features for each stock dataset. The final feature subset S_{final} is presented in Table 4.

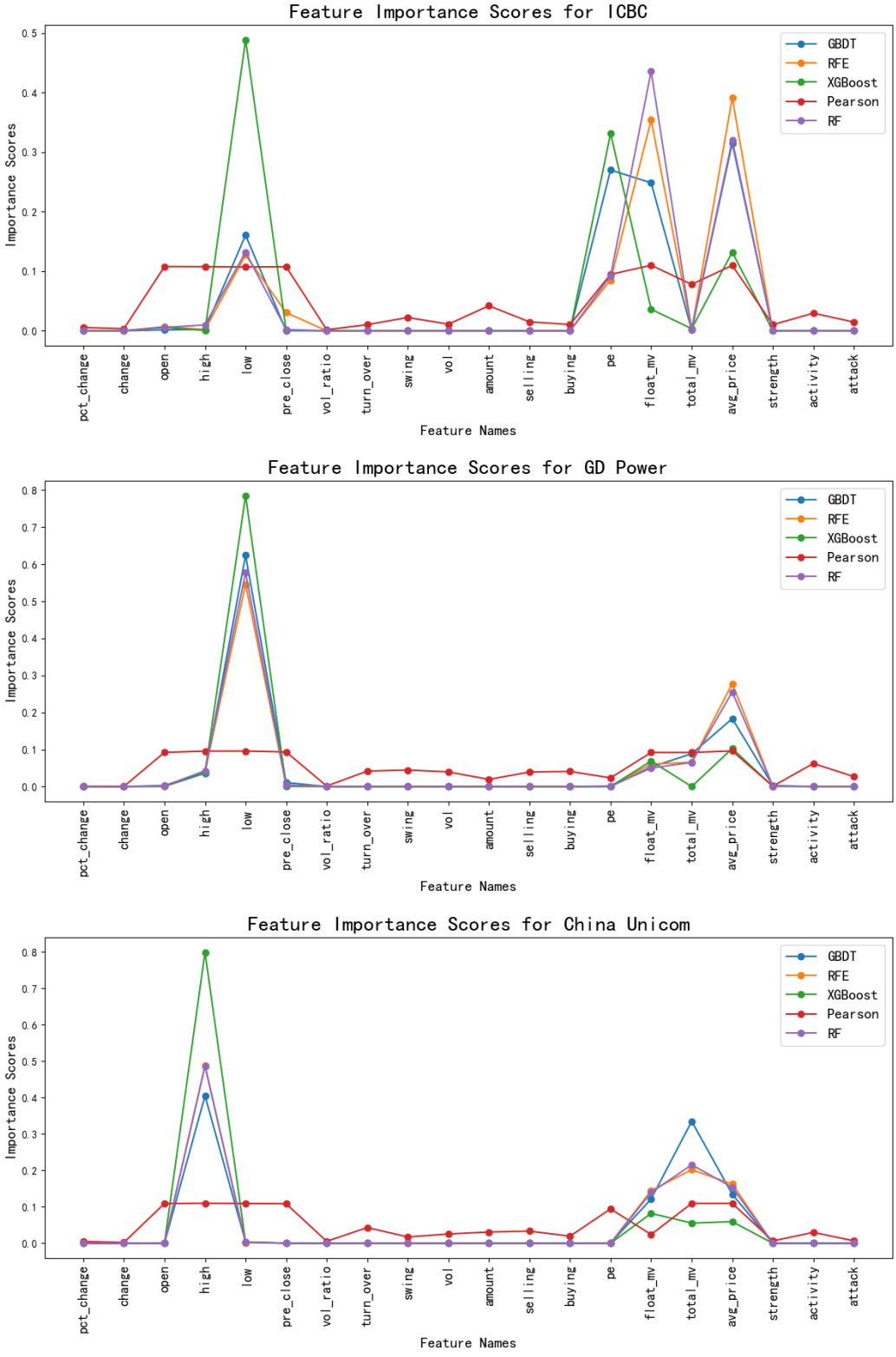

Figure 2. Importance scores of features for each stock.

Table 3. Table of weights for feature selection methods.

Feature Selection Methods	ICBC	GD Power	China Unicom
GBDT	0.21	0.2	0.21
RFE	0.17	0.21	0.2
XGBoost	0.21	0.22	0.2
Pearson	0.2	0.16	0.18
RF	0.21	0.21	0.21

Table 4. The feature subsets obtained through various feature selection methods.

Feature Selection Methods	Feature Subset of ICBC Stock Data	Feature Subset of GD Power Development Stock Data	Feature Subset of China Unicom Stock Data
GBDT	avg_price, pe, float_mv, low, high	low, avg_price, total_mv, float_mv, high	high, total_mv, avg_price, float_mv, low
RFE	avg_price, float_mv, low, pe, pre_close	low, avg_price, total_mv, float_mv, high	high, total_mv, avg_price, float_mv, low
XGBoost	low, pe, avg_price, float_mv, open	low, avg_price, float_mv, high, open	high, float_mv, avg_price, total_mv, low
Pearson	float_mv, avg_price, open, high, pre_close	avg_price, low, high, pre_close, open	high, total_mv, avg_price, low, open
RF	float_mv, avg_price, low, pe, high	low, avg_price, total_mv, float_mv, high	high, float_mv, avg_price, total_mv, low
Ensemble Learning	avg_price, float_mv, low, pe, pre_close	low, avg_price, total_mv, float_mv, high	high, float_mv, avg_price, total_mv, low

Ultimately, to validate the effectiveness of the ensemble learning-based feature selection method, this study applied the five features S_i selected by each feature selection method and the final five features S_{final} chosen by the ensemble method to the task of stock price time-series prediction. The dataset, again, included the stock data of three companies: ICBC, GD Power, and China Unicom. The time range remained from 1 January 2020 to 1 January 2023.

Long Short-Term Memory (LSTM) was chosen as the specific prediction model to ensure accuracy in the forecasting task. LSTM networks are a variant of recurrent neural networks specifically designed for processing and learning from time-series data. The core components of an LSTM network include cells and gates, with three main gates: the input gate, forget gate, and output gate. The memory cell is the heart of the LSTM network and is responsible for storing and passing information. The input gate determines which information will be written to the memory cell, the forget gate decides which information will be removed from the memory cell, and the output gate determines which information will be extracted from the memory cell. These gates govern the flow of information in and out, and the updating of the memory state within the cell. The primary computational processes of an LSTM network can be represented by the following equations:

$$\begin{aligned} f_t &= \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big) \\ i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t &= f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \\ o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t &= o_t \cdot \tanh(C_t) \end{aligned} \tag{10}$$

where f_t represents the output of the forget gate, i_t is the output of the input gate, $\tilde{C}_t$ is the new candidate memory, C_t is the current state of the memory cell, o_t is the output of the output gate, h_t is the hidden state of the LSTM, and W_f, W_i, W_C, W_o are weight matrices, while b_f, b_i, b_C, b_o are bias vectors. The symbol σ denotes the sigmoid activation function. These equations describe the primary computational processes of the LSTM network, allowing the network to more effectively capture long-term dependencies when handling time-series data.

For the evaluation of prediction performance, we utilized widely used metrics, namely, Mean Absolute Error (MAE) and Mean Squared Error (MSE). The model parameters are specified as outlined in Table 5.

Table 5. Table of prediction model parameter settings.

Parameter Name	Parameter Value	Parameter Description
input_size	5	Input Feature Dimension
hidden_size	64	Number of Hidden Units
num_layers	2	Network Depth
learning_rate	0.001	Learning Rate
batch_size	64	Batch Size
num_epochs	100	Epochs
seq_length	5	Sequence Length

The parameter input_size represents the dimensionality of the input features, indicating the number of features input to the model at each time step. In the context of this predictive task, each time step comprised five features, making this parameter equal to 5. Hidden_size denotes the number of hidden units. In LSTM, these units capture patterns and relationships in time-series data. With 64 hidden units, the model exhibited a more complex learning capacity. Num_layers determines the depth of the network, i.e., the number of stacked LSTM layers. Here, two LSTM layers were stacked together, each with its own hidden state. Learning_rate is the learning rate, controlling the step size of model parameter updates. A smaller learning rate promotes model stability. Batch_size indicates the number of samples input to the model in each update, with larger batch sizes enhancing training efficiency. Num_epochs specifies the number of iterations the model underwent over the entire training dataset. Seq_length is the sequence length, representing the temporal span of historical data considered at each time step. In this case, the model utilized data from the past 5 days to predict the closing price on the 6th day; hence, this parameter was set to 5. The model employed the Adam optimizer and was implemented using the PyTorch framework.

The final prediction results are presented in Table 6, with the best-performing metrics highlighted in bold. It is evident from the table that the gradient boosting decision tree method exhibited the best performance on one dataset, indicating that this method effectively identified the most influential features in that dataset. On the other hand, the recursive feature elimination and random forest methods achieved the best performance on two different datasets each, emphasizing their effectiveness in specific contexts. Moreover, ensemble learning methods demonstrated optimal performance on all three different datasets, underscoring their comprehensive applicability across various data types.

These findings underscore the crucial role of ensemble learning-based feature selection methods in enhancing the accuracy of feature selection and optimizing the performance of time-series predictions. Employing features selected through ensemble learning for modeling demonstrated improved performance in terms of both the MAE and MSE metrics, effectively enhancing the effectiveness of time-series prediction models. This outcome emphasizes the potential of ensemble learning methods in improving the accuracy of feature selection and predictive outcomes, providing robust support for research and applications in the field of time-series prediction.

Table 6. Predicted prices for typical stocks under different feature selection methods. The best values are shown in bold.

Feature Selection Methods	Dataset	MAE	MSE
GBDT	ICBC	0.2177	0.0618
	GD Power	**0.2548**	**0.0667**
	China Unicom	0.1469	0.0318
RFE	ICBC	**0.192**	**0.049**
	GD Power	**0.2548**	**0.0667**
	China Unicom	0.1469	0.0318
XGBoost	ICBC	0.2462	0.0789
	GD Power	0.2635	0.0713
	China Unicom	0.1469	0.0318
Pearson	ICBC	0.235	0.0716
	GD Power	0.2723	0.0761
	China Unicom	0.1438	0.0305
RF	ICBC	0.2177	0.0618
	GD Power	**0.2548**	**0.0667**
	China Unicom	**0.1386**	**0.0284**
Ensemble Learning	ICBC	**0.192**	**0.049**
	GD Power	**0.2548**	**0.0667**
	China Unicom	**0.1386**	**0.0284**

5. Conclusions

This paper proposes a feature selection method based on ensemble learning. Firstly, we clearly defined the application of ensemble learning to feature selection and selected five feature selection methods based on the "good but different" principle for integration. When dealing with specific datasets, we determined the weights of each feature selection method through K-fold cross-validation. Ultimately, we adopted a feature score weighting approach, synthesizing the advantages of each method to select the final feature subset. Subsequently, a series of experiments were conducted to validate the effectiveness of this method. We utilized stock data from ICBC, GD Power, and China Unicom to construct a dataset, employing ensemble learning to filter out the top five features with the most significant impact on prediction results from the dataset's twenty features. Following this, we employed LSTM to build a time-series prediction model, comparing the predictions of the five features selected by each method with those selected by the ensemble learning method. The experimental results demonstrate the superior performance of the ensemble learning method in terms of the MAE and MSE metrics, effectively showcasing the method's efficacy and robustness.

In summary, this study not only offers profound insights into the theoretical application of ensemble learning in feature selection but also validates its effectiveness in practical applications across various domains. It provides valuable insights for research and practice in related fields. Future research directions could include further optimizing the determination methods for ensemble learning weights and extending the application of this method to other time-series data domains, such as electricity and weather. These efforts will contribute to deeper insights and beneficial revelations for both the research and practical applications in the field of feature selection.

Author Contributions: Conceptualization, D.H. and Z.L.; methodology, D.H.; software, Z.L.; validation, D.H., Z.L. and D.W.; formal analysis, D.W.; investigation, Z.L.; resources, D.H.; data curation, D.H.; writing—original draft preparation, Z.L.; writing—review and editing, D.H.; visualization, D.W.; supervision, D.W.; project administration, D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The research was approved by the Research Ethics Committee of National University of Defense Technology. Approval code: LWBMSP202307112728. Approval date: 11 July 2023.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Data for this study are available from the corresponding author on request. The data are not publicly available due to privacy restrictions.

Acknowledgments: The authors would like to thank all the college students who participated in the experiment and the reviewers and editors for their reviews of this study.

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

References

1. Tong, Q.; Zhang, K.; Du, J. Exponential smoothing forecasting method and its application in economic forecast. *Econ. Res. Guide* **2013**, *4*, 11–13.
2. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; OTexts: Melbourne, Australia, 2018.
3. Li, C.; Yang, B.; Li, M. Forecasting analysis of Shanghai stock index based on ARIMA model. *MATEC Web Conf.* **2017**, *100*, 02029. [CrossRef]
4. Wang, B.; Hao, W.N.; Chen, G.; He, D.C.; Feng, B. A wavelet neural network forecasting model based on ARIM. *Appl. Mech. Mater.* **2013**, *347*, 3013–3018. [CrossRef]
5. Choi, H.K. Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model. *arXiv* **2018**, arXiv:1808.01560.
6. Wang, J.Z.; Wang, J.J.; Zhang, Z.G.; Guo, S.P. Forecasting stock indices with back propagation neural network. *Expert Syst. Appl.* **2011**, *38*, 14346–14355. [CrossRef]
7. White, H. Economic prediction using neural networks: The case of IBM daily stock returns. *ICNN* **1988**, *2*, 451–458.
8. Zhang, H. The forecasting model of stock price based on PCA and BP neural network. *J. Financ. Risk Manag.* **2018**, *7*, 369–385. [CrossRef]
9. Kim, T.; Kim, H.Y. Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data. *PLoS ONE* **2019**, *14*, e0212320. [CrossRef]
10. Li, X.; Li, Y.; Yang, H.; Yang, L.; Liu, X.Y. DP-LSTM: Differential privacy-inspired LSTM for stock prediction using financial news. *arXiv* **2019**, arXiv:1912.10806.
11. Sen, J.; Mehtab, S.; Nath, G. Stock price prediction using deep learning models. *Lattice Mach. Learn. J.* **2020**, *1*, 34–40.
12. 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]
13. Zhou, X.; Pan, Z.; Hu, G.; Tang, S.; Zhao, C. Stock market prediction on high-frequency data using generative adversarial nets. *Math. Probl. Eng.* **2018**, *2018*, 4907423. [CrossRef]
14. Lee, J.W. Stock price prediction using reinforcement learning. In Proceedings of the ISIE 2001, International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), Pusan, Republic of Korea, 12–16 June 2001; Volume 1, pp. 690–695.
15. Lee, J.; Kim, R.; Koh, Y.; Kang, J. Global stock market prediction based on stock chart images using deep Q-network. *IEEE Access* **2019**, *7*, 167260–167277. [CrossRef]
16. Li, X.; Li, Y.; Zhan, Y.; Liu, X.Y. Optimistic bull or pessimistic bear: Adaptive deep reinforcement learning for stock portfolio allocation. *arXiv* **2019**, arXiv:1907.01503.
17. Wan, C.; Li, W.Z.; Ding, W.X.; Zhang, Z.; Ye, B.; Lu, S. A multivariate time series forecasting algorithm based on self-evolution and pre-training. *Chin. J. Comput.* **2022**, *45*, 513–525.
18. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, CA, USA, 2016.
19. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [CrossRef]
20. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
21. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
22. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
23. Graves, A.; Schmidhuber, J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Netw.* **2005**, *18*, 602–610. [CrossRef]
24. Siami-Namini, S.; Tavakoli, N.; Namin, A.S. The performance of LSTM and BiLSTM in forecasting time series. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 3285–3292.
25. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł. Illia Polosukhin Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
26. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in time series: A survey. *arXiv* **2022**, arXiv:2202.07125.

27. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
28. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]