mathematics

MDPI

Article

Enhancing Bitcoin Price Volatility Estimator Predictions: A Four-Step Methodological Approach Utilizing Elastic Net Regression

Georgia Zournatzidou [1], Ioannis Mallidis [2,*], Dimitrios Farazakis [3,4] and Christos Floros [1]

[1] Department of Accounting and Finance, Hellenic Mediterranean University, 71410 Heraklion, Greece; zournatzidou.georgia@gmail.com or zournatzidou@hmu.gr (G.Z.); cfloros@hmu.gr (C.F.)
[2] Department of Statistical and Insurance Science, University of Western Macedonia, 50100 Kozani, Greece
[3] Institute of Applied and Computational Mathematics, Foundation for Research and Technology Hellas (FORTH), 71110 Heraklion, Greece; d.farazakis@outlook.com
[4] Department of Mathematics, University of Western Macedonia, 52100 Kastoria, Greece
* Correspondence: imallidis@uowm.gr

Abstract: This paper provides a computationally efficient and novel four-step methodological approach for predicting volatility estimators derived from bitcoin prices. In the first step, open, high, low, and close bitcoin prices are transformed into volatility estimators using Brownian motion assumptions and logarithmic transformations. The second step determines the optimal number of time-series lags required for converting the series into an autoregressive model. This selection process utilizes random forest regression, evaluating the importance of each lag using the Mean Decrease in Impurity (MDI) criterion and optimizing the number of lags considering an 85% cumulative importance threshold. The third step of the developed methodological approach fits the Elastic Net Regression (ENR) to the volatility estimator's dataset, while the final fourth step assesses the predictive accuracy of ENR, compared to decision tree (DTR), random forest (RFR), and support vector regression (SVR). The results reveal that the ENR prevails in its predictive accuracy for open and close prices, as these prices may be linear and less susceptible to sudden, non-linear shifts typically seen during trading hours. On the other hand, SVR prevails for high and low prices as these prices often experience spikes and drops driven by transient news and intra-day market sentiments, forming complex patterns that do not align well with linear modelling.

Keywords: elastic net regression; volatility estimators; time-series analysis

MSC: 37M10

Citation: Zournatzidou, G.; Mallidis, I.; Farazakis, D.; Floros, C. Enhancing Bitcoin Price Volatility Estimator Predictions: A Four-Step Methodological Approach Utilizing Elastic Net Regression. *Mathematics* **2024**, *12*, 1392. https://doi.org/10.3390/math12091392

Academic Editor: Andrea Scozzari

Received: 5 March 2024
Revised: 24 April 2024
Accepted: 28 April 2024
Published: 2 May 2024

1. Introduction

In the rapidly evolving landscape of financial markets, the accurate prediction of cryptocurrency prices is critical for a wide range of financial stakeholders, due to their highly volatile behaviour.

Volatility estimators, which are critical tools necessary for analysing cryptocurrency price movements, constitute a key component used to quantify the intrinsic uncertainty and risk present in the market [1]. However, their prediction encompasses inherent challenges associated with the stochastic behaviour of financial time series, the high frequency of outlier occurrences, the impact of unforeseen global events which can disrupt established patterns, and their nonstationary behaviour that exhibits periods of high volatility clustering [2].

ENR stands out as a highly effective tool for addressing the challenges emerging from financial time series analysis. With respect to the stochastic behaviour of financial data and through its L2 penalty, ENR stabilizes the model's predictions by reducing the

magnitude of the impacts of correlated predictors, which is particularly useful when data encompass extreme variations [3]. Additionally, by penalizing the size of the coefficients, ENR minimizes the impact of outliers that could affect predictions in more simplified regression models. This adjustment reduces the probability of overfitting to noise and outliers, which is in turn critical for maintaining predictive accuracy [4]. In the context of global unforeseen events, ENR's focus on core and influential variables allows the model to adapt more effectively. By avoiding overfitting and focusing on significant predictors, the model pertains higher degrees of freedom to adapt to major changes [5]. Finally, for non-stationarity and volatility clustering issues, ENR addresses relevant predictors and reduces the influence of less pertinent volatilities, ensuring continued effectiveness during high-volatility periods, which is a common feature in financial markets [6].

ENR is however, also characterized by significant limitations: (a) ENR's parameters (alpha and lambda) are not optimized during the model's training process and must be specified in advance, often requiring extensive cross-validation to determine their optimal values [4]; (b) while ENR handles multicollinearity and feature selection well, it struggles in high-dimensional spaces. This situation can lead to computational difficulties and the need for dimensionality reduction techniques [7]; and (c) although ENR generally offers improved interpretability due to its feature-selection and noise-elimination capabilities, it may lag in predictive accuracy, especially when the underlying model is complex or when balancing bias and variance is challenging. This trade-off between interpretability and accuracy is particularly crucial in applications where accurate predictions are more important than model simplicity and interpretability [8].

To address the first limitation, we employ Python's RandomizedSearchCV method to optimize hyperparameters (alpha and lambda). This method involves a non-deterministic search that randomly samples parameter values from a defined distribution over a specified number of iterations, avoiding the exhaustive characteristic of traditional grid searches and potentially enhancing the model's performance significantly [9]. For addressing the second limitation, and as autoregressive lags are key variables in financial time series, we employ the Random Forest Regression (RFR) methodological approach developed by Polyzos and Siriopoulos [10] to optimize the selection of time lags in autoregressive models using an ensemble of decision trees. To this end, and for addressing the third limitation, we provide a comparative analysis of ENR's accuracy against traditional machine learning regression models such as DTR, RFR, and SVR. DTR offers a straightforward and interpretable model structure, making it a useful baseline for comparison [11], while RFR is known for its robustness and consistently high accuracy across various datasets, serving as an excellent benchmark for evaluating the accuracy of ENR [12]. Finally, SVR excels at handling complex, non-linear relationships, allowing us to assess how well ENR performs compared to these sophisticated techniques in capturing predictive accuracy [13].

The remainder of this paper is organized as follows: A literature review of the predictive models utilized for forecasting high-frequency data is presented in Section 2. Subsequently, the employed methodology is outlined in Section 3. The model's numerical implementation and derived results are presented in Section 4. Section 5 provides a discussion of the paper's objectives and insights, while Section 6 concludes with the project's results and future research perspectives.

2. Literature Review

Predictive models play a critical role guiding trading decisions in financial markets with intraday data. Numerous forecasting methods have been developed to handle the specificities of these markets. Studies focusing on cryptocurrency market trends have led to the utilization of predictive models that mainly leverage machine learning and deep learning techniques focusing on cryptocurrency prices solely [14].

On this basis Long Short-Term Memory (LSTM) networks have proven effective for capturing the complexities of cryptocurrency price fluctuations with Kumar et al. [15] assessing Long Short-Term Memory (LSTM) networks for predicting prices, while empha-

sizing the model's ability to adjust to the cryptocurrency price fluctuations. The authors employed numerous market data features such as high, low, open, close, and market cap values. These values were then transformed into a multi-dimensional array to effectively utilize the LSTM's capability to process sequential data. The dataset included approximately 78,922 rows of cryptocurrency data, with a specific focus on Bitcoin and Ethereum. The LSTM model's performance was then compared with other models like the autoregressive (AR), the moving average (MA), and the Autoregressive Integrated Moving Average Model (ARIMA), with the LSTM achieving the highest accuracy of about 71%.

Similarly, Jethani et al. [16] employed the LSTM network for the prediction of stock market trends. The authors evaluated the model's performance compared to traditional models and advanced deep learning techniques. The results obtained from their study indicated that the proposed LSTM model can predict more accurately compared to alternative models, in cases of highly volatile financial time series. The authors employed feature engineering techniques with features like opening, close, high, low prices, and volumes of traded stocks. The dataset employed, involved data from January 2018 to April 2021, covering diversified Indian companies and capturing the impacts of major economic events, including the occurrence of the COVID-19 pandemic.

Lumoring et al. [17] highlighted the significance of machine learning model implementation to predict trends in the cryptocurrency sector and stock market, focusing on the effectiveness of Support Vector Machines (SVM) and Long Short-Term Memory (LSTM) network models. The features considered typically included historical price data and trading volumes. Evaluation metrics such as accuracy and Mean Absolute Percentage Error (MAPE) were used, with LSTM models achieving the highest predictive accuracy.

Finally, Liu [18] compared deep learning techniques with traditional time series analysis in stock market prediction. The results have shown that the LSTM model attains higher predictive accuracy in AAPL stock compared to the classical ARIMA model. The employed dataset spans from September 2013 through to September 2023 and the prediction accuracy was measured using evaluation metrics such as the Mean Squared Error (MSE) and Mean Absolute Error (MAE).

Researchers have also delved into different frameworks of neural networks with the purpose of further improving the forecasting accuracy. The authors focused on stochastic neural networks (SNNs), which encompass stochastic processes that manage to accurately emulate the stochastic nature of the financial markets. As reported by Jay et al. [19], these models are an attempt to predict cryptocurrency price volatility with the use of processed market indicators and systematically outperform the more deterministic and simplistic models. The authors focus on key cryptocurrencies, including Bitcoin, Ethereum, and Litecoin, with the multitude of market indicators namely, price, volume, and even mining difficulty, as features. The dataset being considered ranged from mid-2017 to the end of 2019, while MAPE, MAE, and RMSE have been employed for assessing predictive accuracy. The findings from this study indicated that stochastic models adapted well to market dynamics and outperformed the deterministic models in a consistent manner.

Hemant et al. [20] proposed a method in which the kernel extreme learning machine (KELM) is coupled with a variational autoencoder (VAE) for predicting stock market trends. The approach of this paper is a combination of the CNN technique for extracting features and the VAE model for predictions towards improving the overall accuracy in stock price prediction. This single approach shows great potential in applying advanced machine learning algorithms for improving predictive accuracy. The model employs an exhaustive set of features, including the use of technical indicators and the history of the data transformed through CNN for deep feature extraction, further processed by VAE to add on to its prediction capabilities. The dataset involved daily price movements of stocks over a 2-year time horizon. The model's accuracy was assessed through MAPE, MAE, and RMSE, with the findings revealing that the combined developed VAE-KELM approach greatly improves the performance of the standard prediction models.

Finally, Charandabi and Kamyar [21] provided a comparative analysis of multiple artificial neural network-based (ANN) approaches to predict cryptocurrency prices, while highlighting the pros and cons of each of the methods presented in terms of the time elapsed and their prediction accuracy. The datasets encompassed a range of historical cryptocurrency price data, with a particular focus on Bitcoin. Overall, their findings suggested that hybrid neural network models could predict cryptocurrency prices and time effectively and accurately, thus providing a foundational approach that supports the use of advanced machine learning techniques in financial market prediction.

The integration of sentiment analysis from social media with historical price data has opened new avenues for predictive accuracy. On this basis, Pathak and Kakkar [22] combine data and sentiment analysis from media using a new sentiment-based neural network model (SBNNM) to improve predictive modelling. Their approach involved extensive feature engineering, integrating sentiment analysis from social media with historical cryptocurrency price data to create a comprehensive feature set. The dataset used comprised real-time and historical data including pricing, volume, and market sentiment extracted from Twitter using advanced natural language processing techniques. The evaluation criteria focused on prediction accuracy and error metrics, with the neural network achieving a prediction accuracy of 77.89%.

Further emphasizing the breadth of methodologies, hybrid approaches involving both numerical and textual data have been employed to predict trends influenced by dynamic market news. Usmani and Shamsi [23] investigated the effectiveness of combining numerical and textual data to predict stock market trends influenced by news, introducing a framework that could also be applied for analysing cryptocurrency markets and emphasizing the increasing role of artificial neural networks (ANN) in improving prediction accuracy through diverse data integration. They employ sophisticated feature engineering, utilizing both structured and unstructured textual features, such as Bag-of-Words and event extraction techniques, alongside numerical data. This integration allows for a deep analysis of the news impact on stock trends. Their findings suggest that neural networks, capable of processing and learning from the complexity of combined data types, offer significant improvements in prediction accuracy, showcasing the potential for these methods in broader financial applications.

Finally, a comprehensive literature review was undertaken by Dopi et al. [24], which focused on the application of machine learning and deep learning techniques in the prediction of cryptocurrency market stock prices. Their study investigated the correlation between the stock and cryptocurrency markets, with a particular focus on the potential and extensive utility of predictive modelling tools in comprehending complex market dynamics. The review highlighted that the researchers predominantly utilize LSTM, MLP, RF, and SVM methods, with MLP showing the best performance at a 71.63% accuracy rate. Feature engineering was extensively used, incorporating both technical and fundamental analysis data, including sentiment analysis from social media. The datasets featured varied widely but commonly included historical price data, volume, and sentiment indicators from news and social media. The evaluation of the models was typically based on accuracy, precision, and loss metrics such as MAE and RMSE, reflecting a rigorous approach to assessing prediction performance. The findings from the review suggest that combining multiple forms of analysis and using advanced machine learning and deep learning techniques can significantly enhance the accuracy of stock price predictions in highly volatile markets like those of cryptocurrencies.

Table 1 provides a critical synthesis of academic research efforts based on the model type that these employs.

The findings from our critical synthesis of the literature highlight several key trends and gaps in the field of cryptocurrency price prediction using machine learning and deep learning techniques. Firstly, the review emphasizes a growing trend in the application of sophisticated machine learning and deep learning models, particularly LSTM, which are renowned for their precision in predicting cryptocurrency prices. Despite their accu-

racy, these models often lack interpretability and require extensive and complex training procedures. Secondly, it is evident that LSTM models dominate the research landscape, reflecting their effectiveness in capturing the dynamics of cryptocurrency prices. Thirdly, our review revealed a general absence of methods that quantitatively assess the impact of delays associated with the transposed volatility estimator of time series. This indicates a need for improved methodologies that can effectively incorporate temporal delays to enhance prediction accuracy. Fourthly, there appears to be a scarcity of research focused specifically on predicting cryptocurrency volatility estimators.

Table 1. Critical synthesis of academic research effort.

References	ML Model(s)	Feature Engineering	Lag Selection	Bitcoin Features
[15]	LSTM, ARIMA, AR, MR	Yes	No	price, market cap
[16]	LSTM	Yes	No	prices
[17]	LSTM, SVM	Yes	No	prices, volumes
[18]	ARIMA, LSTM	Yes	No	prices
[19]	SNN	Yes	No	prices, volumes, and mining difficulty
[20]	VAE, SNN	Yes	No	prices, technical indicators
[21]	ANN	Yes	No	prices
[22]	SBNNM	Yes	No	prices, volume, and market sentiment
[23]	ANN	Yes	No	prices, volumes, and market sentiments
[24]	LSTM, MLP, RF, SVM	Yes	No	prices, volumes, and market sentiments

To address these identified gaps, our paper proposes several approaches:

- ElasticNet Regression Approach: We propose the development and implementation of an ENR model, with optimized hyperparameter values, that balances computational efficiency with high prediction accuracy, providing a more interpretable model compared to deep learning methods.
- Two-Step Methodological Framework for Lag Optimization: To tackle the challenge of quantifying the impacts of lags, our study employs a two-step methodology. The first step involves identifying the optimal number of autoregressive delays, which are then incorporated as additional independent variables in the dataset. This integration allows for a more comprehensive examination and enhances the precision of predictions.
- Volatility Estimators from Bitcoin Prices: We focus on designing effective machine learning models, particularly utilizing volatility estimators derived from Bitcoin prices. This initiative aims to fill the gap in research concerning the prediction of cryptocurrency volatility estimators.
- Comparative Analysis of Model Accuracy: We provide a comparative analysis of the predictive accuracy of the ENR model against other traditional models such as Decision Trees, Random Forests, and Support Vector Machines.

3. Methodology

This section provides the mathematical analysis employed for deriving the functions quantifying the volatility estimators of open, high, low, and close cryptocurrency prices.

The first volatility estimator employed is the Parkinson Volatility Estimator [25] of Equation (1).

$$\widehat{\sigma_p^2} = \frac{(p_{max} - p_{min})^2}{4ln2}, \tag{1}$$

where $p_{max} = ln(High) - ln(Open)$, $p_{min} = ln(Low) - ln(Open)$ and *High*, *Open*, and *Low* correspond to the *High*, *Open*, and *Low* cryptocurrency price data. In our analysis, we employ this function for exclusively quantifying the volatility estimators of open prices solely. The intuition behind the selection of the Parkinson Volatility Estimator solely for open prices hinges upon the fact that this estimator exhibits a unique ability to capture the market's immediate reaction to overnight news and events at the opening bell [26].

Regarding high cryptocurrency prices, the second volatility estimator employed involves the Garman–Klass Estimator [27] of Equation (2)

$$\widehat{\sigma_{GK}^2} = 0.5 \cdot (p_{max} - p_{min})^2 - (2ln2 - 1) \cdot p_{close}^2, \quad (2)$$

where $p_{close} = ln(Close) - ln(Open)$, with *Close* corresponding to the *close* cryptocurrency prices. This volatility estimator is specifically tailored for high volatility assets like cryptocurrencies due to its inclusion of the full range of price movements (high, low, and close) within the trading day [27].

The third volatility estimator function employed for low cryptocurrency prices is mathematically expressed through Equation (3) [28]

$$\widehat{\sigma_M^2} = 0.274 \cdot \sigma_1^2 + 0.16 \cdot \sigma_s^2 + 0.365 \cdot \sigma_3^2 + 0.2 \cdot \sigma_4^2, \quad (3)$$

where $\sigma_1^2 = 2\left[\left(p'_{max} - p'_{close}\right)^2 + p'_{low}\right]$, $\widehat{\sigma_s^2} = p_{close}^2$, $\sigma_3^2 = 2\left(p'_{max} - p'_{close} p'_{min}\right) p'_{close}$, $\sigma_4^2 = -\frac{\left(p'_{max} - p'_{close}\right) p'_{min}}{2ln2 - \frac{5}{4}}$ and $p'_{close} = p_{close}, p'_{max} = p_{max}, p'_{min} = p_{min}$ if $p_{close} > 0$ and $p'_{close} = -p_{close}, p'_{max} = -p_{min}, p'_{min} = -p_{max}$ if $p_{close} < 0$.

We selected the above volatility estimator for low cryptocurrency prices since this estimator has the capability to integrate $\sigma_1^2, \sigma_s^2, \sigma_3^2$ and σ_4^2, in a single comprehensive measure. This approach is particularly effective in low-volatility scenarios, where traditional estimators might not capture the full extent of market dynamics or might overemphasize the impact of minor price changes [28].

The final fourth volatility estimator employed for *close* prices is the Rogers–Satchell (*RS*) estimator [29] of Equation (4).

$$\widehat{\sigma_{RS}^2} = p_{max} \cdot (p_{max} - p_{close}) + p_{min} \cdot (p_{min} - p_{close}), \quad (4)$$

The reason for selecting this volatility estimator for close prices is mainly attributed to the fact that the estimator captures the entire range of intraday price movements in relation to both the opening and closing prices. Its formulation incorporates the high, low, open, and close prices, enabling it to reflect the volatility surrounding the closing price effectively [29].

Given the selection of the appropriate mathematical functions for quantifying the volatility estimators of cryptocurrency prices, the second step of our approach involves the determination of the optimal number of lags for capturing both the behaviour of the recent volatility coefficient values along with the inherent autocorrelation in the data [30].

We then employ the RFR model to optimize the selection of time-series lags in our predictive model. The approach begins by constructing a forest of decision trees, where each tree is built from a bootstrap sample of the data. The RFR model utilizes the Mean Decrease in Impurity (MDI) criterion to evaluate the importance of each lag in predicting the volatility estimator [10]. The optimal number of lags are the lags resulting to an 85% cumulative importance. This criterion has been empirically selected as it achieves a balance between model complexity and explanatory power, ensuring that the model retains the most significant predictors while avoiding overfitting by excluding less impactful variables.

Having selected the optimal number of lags, we finalized the dataset structure and split the dataset to an 80% train and 20% test set. We then formulated the ENR model and fit the model on the train set. The ENR model constitutes a combined statistical tool of Ridge and Lasso (Least Absolute Shrinkage and Selection Operator) regression, based on the OLS method. Ridge regression, Lasso regression, and Elastic Net regression are techniques used in the field of machine learning and statistics for regularization, which

helps in reducing model complexity and preventing overfitting. The linear problem of the form [31] is presented in Equation (5).

$$\widehat{y_t} = \beta_0 + \left(\sum_{j=1}^{J} \beta_j x_{tj}\right) + \varepsilon_i, \forall t \in T \tag{5}$$

Here, the vector x_{tj} represents the value of the independent variable $j \in J$ at time instance $t \in T$ and β_j is the regression coefficient vector of the independent variables. Then, the sum of squared residuals is defined through Equation (6) as follows:

$$SSR = \sum_{t=1}^{T} (y_t - \widehat{y_t})^2, \tag{6}$$

We then estimated the values of the coefficients by using the ordinary least square (OLS) model, and through the minimization of the sum of the squared residuals of Equation (7).

$$\left(\hat{\beta}_0, \widehat{\beta}_j\right)_{OLS} = argmin_{(\hat{\beta}_0, \hat{\beta}_j) \in R^{J+1}} \left[\sum_{t=1}^{T} (y_t - \hat{y}_t)^2\right] = argmin_{(\widehat{\beta}_0, \widehat{\beta}_j) \in R^{J+1}} (SSR), \tag{7}$$

By incorporating penalization techniques into the OLS framework, we gain an understanding of the Ridge and Lasso regression. Ridge regression employs ridge constraints to restrict the size of certain coefficients, thereby regulating the selection of variables [32]. Ridge regression augments the loss function with a penalty equal to the square of the magnitude of the coefficients. By penalizing the inclusion of large coefficients, this method effectively mitigates the complexity of the model and aids in addressing multicollinearity, which occurs when independent variables are significantly correlated. It is particularly useful when dealing with data where the number of predictors (variables) is close to or exceeds the number of observations. The penalty term is controlled by a hyperparameter, often denoted by $\lambda \geq 0$, which determines the extent of regularization. The higher the value of λ, the greater the amount of shrinkage of the coefficients towards zero. The model is developed through Equation (8) as follows:

$$\left(\hat{\beta}_0, \widehat{\beta}_j\right)_{Ridge} = argmin_{(\hat{\beta}_0, \hat{\beta}_j) \in R^{J+1}} \left[\sum_{t=1}^{T} (y_t - \widehat{y_t})^2 + \lambda \sum_{j=1}^{K} \beta_j^2\right], \tag{8}$$

We can notice that the difference between the estimations $\left(\hat{\beta}_0, \widehat{\beta}_j\right)_{OLS}$ and $\left(\hat{\beta}_0, \widehat{\beta}_j\right)_{Ridge}$ is the l_2 penalty of the form:

$$\left\|\beta_j\right\|_2^2 = \sum_{j=1}^{J} \beta_j^2 \leq M, \tag{9}$$

for M to be the upper bound for the sum of the coefficients. Lasso Regression also adds one more penalty (compared to the ridge regression) to the loss function, but unlike Ridge, the penalty is the absolute value of the magnitude of coefficients. This can lead to some coefficients being exactly zero when the penalty is large enough, effectively performing variable selection and producing models that are sparse [33]. This is particularly useful for models that benefit from variable reduction/selection. Similar to Ridge, the strength of the regularization is controlled by a hyperparameter, λ. Lasso regression satisfies the next optimization problem [34].

$$\left(\hat{\beta}_0, \widehat{\beta}_j\right)_{Lasso} = argmin_{(\hat{\beta}_0, \hat{\beta}_j) \in R^{J+1}} \left[\sum_{t=1}^{T} (y_t - \widehat{y_t})^2 + \lambda \sum_{j=1}^{J} |\beta_j|\right], \tag{10}$$

with l_1 penalty of the form

$$\left\|\beta_j\right\|_1 = \sum_{j=1}^{J}|\beta_j|, \tag{11}$$

Then, the combination of the two previous regression methods, Ridge and Lasso, yields the elastic net regression [4]. It adds both penalties (the square of the magnitude and the absolute value of the coefficients) to the loss function. This approach aims to leverage the benefits of both Ridge and Lasso regression. It is useful when there are multiple features correlated with each other. Elastic Net has two parameters to control the mix of Ridge and Lasso penalties, which controls the overall strength of the penalties [35]. The model for elastic net regression is then expressed through the following Equation (12) as follows:

$$\left(\hat{\beta}_0,\ \widehat{\beta}_j\right)_{EN} = argmin_{(\widehat{\beta}_0,\ \widehat{\beta}_j)\in R^{J+1}}\left[\sum_{t=1}^{T}(y_t-\widehat{y}_t)^2 + \frac{\lambda(1-\alpha)}{2}\sum_{j=1}^{J}\beta_j^2 + \lambda\alpha\sum_{j=1}^{J}|\beta_j|\right], \tag{12}$$

In order to optimize the selection of the Elastic Net Regression's (ENR) hyperparameters, α and λ, which dictate the balance between the L1 and L2 penalties, we employ the RandomizedSearchCV method using the Python programming language on the train set. This methodological approach is implemented using libraries such as scikit-learn for the RandomizedSearchCV functionality and NumPy for numerical operations. We randomly select combinations from a predefined grid of α and λ values and evaluate the performance of each combination using k-fold cross-validation. This stochastic sampling method is not only computationally less intensive than an exhaustive grid search, but also provides a practical compromise between thoroughness and efficiency.

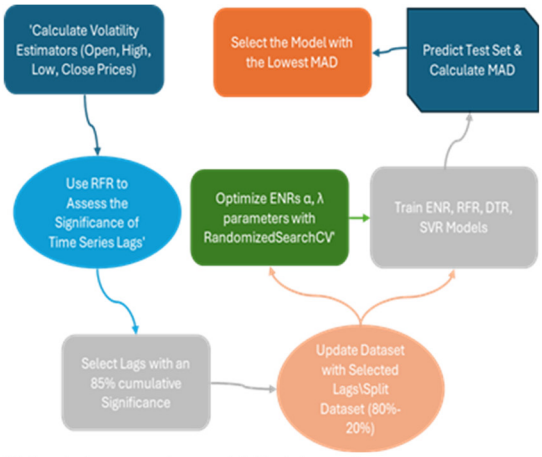
Table 2 provides the nomenclature Table of the examined elastic net model parameters.

Table 2. Nomenclature of Elastic net Regression.

Model Parameters	Nomenclature
β_0	y intercept
β_j	Vector of the independent variable coefficients
y_t	Actual values of the prices at time t
$\widehat{y}_t$	Predicted values of the prices at time t
T	Dataset time periods (either train or test)
λ	Regularization parameter. Higher λ means higher regularization
α	Takes values of 0–1. Values of α higher than 0.5, means a higher impact of the Lasso regression regularization on the prediction, else a higher impact of Ridge regression
$\left\|\beta_j\right\|_1$	Sum of absolute values of the coefficients
$\left\|\beta_j\right\|_2^2$	Sum of squares of the coefficients

To this end and for providing a comparative assessment of the ENR's predictive accuracy, we compare ENR with traditional machine learning regression models such as the DTR, the RFR, and the SVR. Considering the optimally selected number of lags for each volatility estimator, these models are fitted on the 80% train dataset for determining their optimal parameter values and employed on predicting the 20% test set dependent variable value. The criterion for assessing the examined machine learning regression models is the Mean Absolute Error (MAE). The model parameters optimized under the decision tree regression model involve the depth and the leaves of each tree, while for the random forest, the average depth, and leaves of the forest trees [6]. Finally, for the SVR, the parameters optimized involve the regularization parameter C, the error margin epsilon, and the number of support vector regressors [36].

The following Figure 1 summarizes the process map and pseudocode of the employed methodology.



Define calculate_volatility_estimators(open_prices, high_prices, low_prices, close_prices):
Calculate volatility using the open, high, low, and close prices
Return volatility_estimators
Define assess_lag_significance(volatility_estimators):
Use RFR to evaluate the significance of each time series lag
Return lag_significance_scores
Define select_significant_lags(lag_significance_scores, significance_threshold):
Filter and retain lags that cumulatively meet the significance threshold
Return selected_lags
Define update_and_split_dataset(data, selected_lags, train_ratio):
Update dataset to include only the selected lags
Split the dataset into training and testing sets based on the specified ratio
Return training_set, testing_set
Define optimize_ENR_parameters(training_set):
Configure and run RandomizedSearchCV with Elastic Net Regression to find optimal α, λ
Return optimized_parameters
Define train_regression_models(training_set):
Train ENR, RFR, DTR, and SVR models
Return trained_models
Define predict_and_calculate_MAD(models, testing_set):
Predict using each model on the testing set
Calculate and return the MAD for each model's predictions
Define select_best_model(models, mads):
Identify the model with the lowest MAD
Return best_model

Figure 1. Process map and pseudocode of the examined methodology.

4. Numerical Analysis and Results

Table 3 summarizes the descriptive statistics of the volatility estimators associated with the open, high, low, and close data from September 2014 to January 2024.

Table 3. Descriptive Statistics of Bitcoin Volatility Estimators.

	σ_p^2	σ_{GK}^2	σ_M^2	σ_{RS}^2
count	3420.0	3420.0	3420.0	3420.0
mean	0.001375	0.999884	0.999983	0.268848
std	0.004913	3.874017	3.041994	0.963543
min	4.34×10^{-15}	7.87×10^{-10}	5.19×10^{-8}	8.95×10^{-13}
25%	0.000033	0.028411	0.025095	0.006589
50%	0.000220	0.153595	0.156463	0.043649
75%	0.001053	0.661257	0.765213	0.209370
max	0.207560	125.595023	100.630430	42.454934

All estimators have a data count of 3421 daily observations, illustrating a robust dataset. The mean values indicate the average level of volatility, showing a progression from minimal at open to more substantial at close, suggesting varying volatility levels at different trading times. The standard deviations are quite broad, especially for the close prices, pointing to significant variability and thus potential unpredictability in Bitcoin's price movements. Minimum values near zero depict days of exceptionally low volatility, whereas the maximum values, which are drastically higher than the means, highlight days of extreme volatility, underscoring the erratic nature of cryptocurrency markets. The percentiles (25%, 50%, and 75%) suggest a right-skewed distribution, indicating that while most days witness lower volatility, a few experiences exceptionally high volatility. These comprehensive data underscore the substantial risk and the necessity for careful risk management in trading and investment strategies concerning Bitcoin.

With respect to the optimal lag selection decisions, the following Figure 2 depicts the lags that lead to a cumulative significance of 85% for each VE_1-4, using the random forest regression methodological approach, while Table 4 summarizes the optimal lags for each VE. As the volatility estimators are daily, we examined 30 lags as our initial lags number.

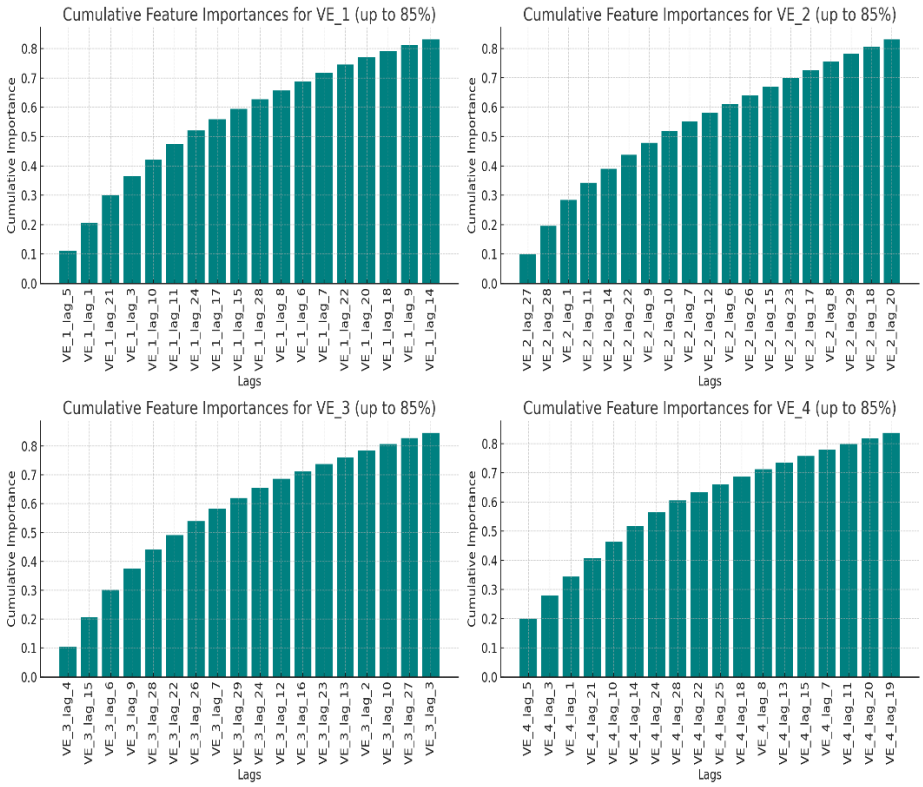


Figure 2. Lags with an 85% cumulative feature importance.

Table 4. Lags Selected per Volatility Estimator.

Estimator	Optimal No of Lags	Top Significant Lags (Up to 85% CI)
VE_1	18	5, 1, 21, 3, 10, 11, 24, 17, 15, 28, 8, 6, 7, 22, 20, 18, 9, 14
VE_2	19	27, 28, 1, 11, 14, 22, 9, 10, 7, 12, 6, 26, 15, 23, 17, 8, 29, 18, 20
VE_3	18	4, 15, 6, 9, 28, 22, 26, 7, 29, 24, 12, 16, 23, 13, 2, 10, 27, 3
VE_4	18	5, 3, 1, 21, 10, 14, 24, 28, 22, 25, 18, 8, 13, 15, 7, 11, 20, 19

Having determined the optimal number of lags using the Random Forest Regression model, we proceed with modelling each Bitcoin price volatility estimator using ENR. Initially, the dataset is divided into training and test sets, with 80% allocated for training and the remaining 20% for testing. The ENR, which incorporates both L1 and L2 regularization, is then configured. To optimize the ENR's hyperparameters efficiently, we employed RandomizedSearchCV. Once the optimal hyperparameters are identified, the next step involves the training of ENR on the training dataset. The trained model is then used to predict the volatility estimators in the test set, considering the identified lags for each volatility estimator as independent variables.

Table 5 provides the summary statistics of the ENR model.

The model results indicate that the intercepts and coefficients vary considerably across different estimators, suggesting unique underlying dynamics in each case. For VE1, the results show minimal coefficient shrinkage ($\lambda = 0.0511$) and a nearly pure ridge behavior ($\alpha \approx 0$), leading to a low mean absolute error (MAE), indicative of a robust model fit. In contrast, VE2 adopts a more LASSO-like approach ($\alpha = 0.9053$) with a moderate λ (0.0940), resulting in a higher MAE, which might imply less predictive accuracy. VE 3 exhibits

the highest regularization (λ = 0.1116) and a balanced α (0.7369), yet it scores the highest MAE among the models, possibly indicating issues with model suitability or data fit. Finally, VE4 demonstrates the highest degree of coefficient shrinkage (λ = 0.5613) with a predominance of ridge regression characteristics (α = 0.0762), which yields a relatively low MAE. These results highlight the critical importance of tuning the λ and α parameters in elastic net models to balance the trade-off between bias and variance effectively, particularly in financial datasets where volatility estimators can behave unpredictably.

Table 5. Elastic net regression statistics.

Coefficients	$VE1 = \sigma_p^2$	$VE2 = \sigma_{GK}^2$	$VE3 = \sigma_M^2$	$VE4 = \sigma_{RS}^2$
β_0 (Intercept)	0.00107	1.0167		0.2515
Lag1	0.0171	−0.0035		0.0000
Lag2				
Lag3	0.0095		−0.006025442	0.0000
Lag4			0.004396279	
Lag5	0.0051			0.0063
Lag6	0.0304	0.00000	0.007677287	
Lag7	0.0080	−0.003956239	0.000000000	0.0614
Lag8	0.0000	0.000000000		0.0000
Lag9	0.0114	−0.01242403	−0.001456905	
Lag10	0.0000	0.004415845		0.0000
Lag11	0.0000	−0.006322465		0.0122
Lag12		−0.003910096	0.000000000	
Lag13			0.000000000	0.0000
Lag14	0.0000	−0.009497958		0.0000
Lag15	0.0005	0.010847662	0.0114516667	0.0000
Lag16			−0.003135042	
Lag17	0.0001	−0.008438236		
Lag18	0.0046	0.000000000		0.0000
Lag19				0.0000
Lag20	0.0000	0.003699343		0.0000
Lag21	0.0000			0.0000
Lag22	0.0000	0.00000000	0.007914059	0.0000
Lag23		−0.011457625	−0.011388763	
Lag24	0.0000		0.004633000	0.0000
Lag25				0.0000
Lag26		0.000013700	−0.007468076	
Lag27		−0.002826818		
Lag28	0.0000	−0.003325371	0.00000000	0.0000
Lag29		0.008371966	0.002658218	
Lag30				
		Parameter Values		
λ (Lambda)	0.0511	0.09405	0.1116	0.5613
α (Alpha)	0.00000795	0.9053	0.7369	0.0762
MAE	0.001572	1.2230	1.2833	0.2076

To further evaluate the predictive power of elastic net regression compared to other regression models, we employed the same process for predicting the dependent target variable values of each bitcoin price volatility estimator considering DTR, RFR, and SVR. The derived optimized model parameter values along with their MAE values are summarized in the following Table 6.

Table 6. Model parameter and MAD values.

Volatility Estimators	Model	Parameters	Values
1	DTR	Depth	38
		Leaves	5407
		MAE	0.0020
	RFR	Average_Depth	40.64
		Average_Leaves	3414.1
		MAE	0.0018
	SVR	C (Regularization parameter)	1
		Epsilon (Error margin)	0.1
		Number_of_Support_Vectors	17
		MAE	0.0839
2	DTR	Depth	49
		Leaves	5423
		MAE	1.8767
	RFR	Average_Depth	43.75
		Average_Leaves	3425.3
		MAE	1.3853
	SVR	C (Regularization parameter)	1
		Epsilon (Error margin)	0.1
		Number_of_Support_Vectors	2136
		MAE	0.8495
3	DTR	Depth	56
		Leaves	5423
		MAE	2.0269
	RFR	Average_Depth	48.19
		Average_Leaves	3425.28
		MAE	1.5067
	SVR	C (Regularization parameter)	1
		Epsilon (Error margin)	0.1
		Number_of_Support_Vectors	2230
		MAE	0.9483
4	DTR	Depth	43
		Leaves	5425
		MAE	0.4224
	RFR	Average_Depth	43.13
		Average_Leaves	3426.6
		MAE	0.3359
	SVR	C (Regularization parameter)	1
		Epsilon (Error margin)	0.1
		Number_of_Support_Vectors	1351
		MAE	0.2492

The following Table 7 summarizes the MAD values of the examined regression methodologies on the train sets of each one of the examined volatility estimators

Table 7. MAE values per regression model and volatility estimator.

Model	VE1	VE2	VE3	VE4
ENR	**0.0016**	1.2366	1.2833	**0.2076**
DTR	0.0020	1.8767	2.0269	0.4224
RFR	0.0018	1.3853	1.5067	0.3359
SVR	0.0839	**0.8495**	**0.9483**	0.2492

The analysis of Table 7 reveals that different regression models exhibit varied effectiveness across the volatility estimators of Bitcoin prices, corresponding to open, high, low, and close values. ENR performs exceptionally well with open and close prices, where the predictive relationships may be more linear and less susceptible to sudden, non-linear shifts typically seen during trading hours. The regularization in ENR helps to prevent overfitting while effectively capturing the linear trends that might govern the open and close price movements. On the other hand, SVR shows superior performance in estimating the volatility of high and low prices as these prices often experience spikes and drops driven by transient news and intra-day market sentiments [37]. SVR, with its capability to handle non-linear relationships through kernel transformations, can capture these dynamics more accurately [38].

5. Discussion

The primary objective of this study is to develop and employ a four-step methodological approach for accurately predicting the volatility estimators of open, high, low, and closed bitcoin prices. The first step of our methodological approach is to transform these prices into volatility estimators using motion assumptions and log price transformations. The second step involves the optimal selection of the number of lags required for accurately capturing the volatile nature of bitcoin prices, while the third step is the formulation of the ENR model on the dataset. The final fourth step provided a comparative analysis of ENR's predictive accuracy with DTR, RFR, and SVR

Our findings reveal the paramount importance of immediate past volatility in predicting short-term fluctuations, highlighting the unpredictable nature of cryptocurrency markets that are prone to sudden shifts.

In comparison to other models tested in our study, the ENR model accuracy varied depending on the specific volatility estimator being analysed. More specifically, ENR exhibited exceptional performance with Open and Close prices, achieving the lowest Mean Absolute Error (MAE) values among the models for these price types. This performance highlights the model's ability to capture the more linear and predictable trends at the opening and closing of trading, where the market dynamics are potentially less volatile and more susceptible to long-term factors.

On the other hand, SVR demonstrated superior performance in estimating the volatility of High and Low prices. These price points often encompass abrupt changes due to intra-day news and market sentiments, resulting in complex, non-linear patterns. SVR's effectiveness in handling these dynamics, due to its capability to model non-linear relationships through kernel transformations, led to more accurate predictions of volatility under conditions of high unpredictability and transient market behaviours.

This analysis indicates that while the ENR effectively prevents overfitting and excellently captures linear relationships, its performance is not uniformly superior across all price volatility estimators. SVR's superior handling of non-linear dynamics suggests that a combination of models may be necessary to fully address the diverse characteristics of Bitcoin's price volatility. This comparative analysis underscores the necessity of employing tailored modelling approaches for different aspects of market behaviour, reflecting the multifaceted nature of cryptocurrency volatility.

6. Conclusions

This research significantly contributes to the field of cryptocurrency analysis by highlighting the efficacy of different regression models, including Elastic Net, in forecasting Bitcoin price volatility. Utilizing a robust dataset spanning from September 2014 to January 2024, this study emphasizes the importance of a methodologically sound approach to lag selection. Such methodologies enhance the model's accuracy and relevance, as demonstrated by the meticulous selection of optimal lags using Random Forest, which is crucial in financial time series analysis. The ENR model excels in modelling the open and close price volatilities, showcasing its strength in capturing more linear and predictable market

behaviours. However, the analysis also reveals that none of the examined regression models uniformly prevailed in terms of predictive accuracy.

Future research could involve assessing a broader range of cryptocurrencies to assess the generalizability of the proposed models across different digital currencies. Furthermore, the inclusion of additional financial and economic indicators could improve the models' predictive capabilities, offering deeper insights into market behaviours.

Finally, the assessment of advanced machine learning and deep learning approaches could also provide valuable perspectives on nonlinear patterns in cryptocurrency volatility, potentially enhancing prediction accuracy.

In conclusion, while this study provides methodological advancements in predicting Bitcoin volatility, it also paves the way for further research to deepen our understanding of cryptocurrency markets and develop more sophisticated predictive models.

Author Contributions: Conceptualization, G.Z. and I.M.; Methodology, G.Z., I.M. and D.F.; Validation, I.M.; Resources, G.Z.; Writing—original draft, G.Z., I.M. and D.F.; Writing—review & editing, G.Z., I.M. and C.F.; Supervision, G.Z. and C.F.; Project administration, G.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

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

References

1. Sinha, N.; Yang, Y. Strategic Diversification for Asynchronous Asset Trading: Insights from Generalized Coherence Analysis of Cryptocurrency Price Movements. *Ledger* **2021**, *6*, 102–125. [CrossRef]
2. Tsay, R.S. *Analysis of Financial Time Series*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2010.
3. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2005**, *67*, 301–320. [CrossRef]
4. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009.
5. Friedman, J.; Hastie, T.; Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J. Stat. Softw.* **2010**, *33*, 1–22. [CrossRef]
6. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning with Applications in R*; Springer: Berlin/Heidelberg, Germany, 2013.
7. Bühlmann, P.; van de Geer, S. *Statistics for High-Dimensional Data: Methods, Theory and Applications*; Springer: Berlin/Heidelberg, Germany, 2011.
8. Hastie, T.; Tibshirani, R.; Wainwright, M. *Statistical Learning with Sparsity: The Lasso and Generalizations*; CRC Press: Boca Raton, FL, USA, 2015.
9. James, B.; Bengio, Y. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* **2012**, *13*, 281–305.
10. Polyzos, E.; Siriopoulos, C. Autoregressive Random Forests: Machine Learning and Lag Selection for Financial Research. *Comput. Econ.* **2023**, 1–38. [CrossRef]
11. Loh, W.Y. Classification and regression trees. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* **2011**, *1*, 14–23. [CrossRef]
12. Liaw, A.; Wiener, M. Classification and regression by random Forest. *R News* **2002**, *2*, 18–22.
13. Drucker, H.; Burges, C.J.C.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *9*, 155–161.
14. de Prado Marcos, L. *Advances in Financial Machine Learning*; Wiley: Hoboken, NJ, USA, 2018.
15. Kumar, A.S.; Gopirajan Pv, G.; Jackson, B. Machine Learning-Based Timeseries Analysis for Cryptocurrency Price Prediction A Systematic Review and Research. In Proceedings of the 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 5–6 April 2023; IEEE: Piscataway, NJ, USA, 2023.
16. Jethani, L.; Patil, R.; Sanghvi, S.; Singh, R. Analysis of Machine Learning Models for Stock Market Prediction. In Proceedings of the 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 24–25 November 2023; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2023; pp. 1–8.
17. Lumoring, N.; Chandra, D.; Gunawan, A.A.S. *A Systematic Literature Review: Forecasting Stock Price Using Machine Learning Approach*; IEEE: Piscataway, NJ, USA, 2023.
18. Liu, L.A. Comparative Examination of Stock Market Prediction: Evaluating Traditional Time Series Analysis Against Deep Learning Approaches. *Adv. Econ. Manag. Political Sci.* **2023**, *55*, 196–204. [CrossRef]

19. Jay, P.; Kalariya, V.; Parmar, P.; Tanwar, S.; Kumar, N.; Alazab, M. Stochastic neural networks for cryptocurrency price prediction. *IEEE Access* **2020**, *8*, 82804–82818. [CrossRef]
20. Hemant; Parida, A.K.; Kumari, R.; Singh, A.R.; Bandyopadhyay, A.; Swain, S. Stock market prediction under a deep learning approach using Variational Autoencoder, and kernel extreme learning machine. In Proceedings of the 2023 OITS International Conference on Information Technology (OCIT), Raipur, India, 13–15 December 2023; pp. 156–161.
21. Charandabi, S.E.; Kamyar, K. Prediction of Cryptocurrency Price Index Using Artificial Neural Networks: A Survey of the Literature. *Eur. J. Bus. Manag. Res.* **2021**, *6*, 17–20. [CrossRef]
22. Pathak, S.; Kakkar, A. Cryptocurrency Price Prediction Based on Historical Data and Social Media Sentiment Analysis. *Lect. Notes Netw. Syst.* **2020**, *103*, 47–55.
23. Usmani, S.; Shamsi, J.A. News sensitive stock market prediction: Literature review and suggestions. *PeerJ Comput. Sci.* **2021**, *7*, 1–36. [CrossRef]
24. Dopi, G.Y.; Hartanto, R.; Fauziati, S. Systematic Literature Review: Stock Price Prediction Using Machine Learning and Deep Learning. In Proceedings of the International Conference on Management, Business, and Technology (ICOMBEST 2021), Jember, Indonesia, 12 October 2021; Volume 194, pp. 52–61.
25. Parkinson, M. The Extreme Value Method for Estimating the Variance of the Rate of Return. *J. Bus.* **1980**, *53*, 61. [CrossRef]
26. Robert, K. Opening and Closing Asymmetry: Empirical Analysis from ISE Xetra. *Econ. Soc. Rev.* **2008**, *39*, 55–78.
27. Meilijson, I. The Garman–Klass Volatility Estimator Revisited. *Revstat. Stat. J.* **2008**, *9*, 199–212. [CrossRef]
28. Robert, Ś.; Grzegorz, Z. High-Frequency and Model-Free Volatility Estimators. Available online: https://ssrn.com/abstract=2508648 (accessed on 19 October 2009). [CrossRef]
29. Yang, D.; Zhang, Q. Drift-Independent Volatility Estimation Based on High, Low, Open, and Close Prices. *J. Bus.* **2000**, *73*, 477–492. [CrossRef]
30. Hamilton, J.D. *Time Series Analysis*; Princeton University Press: Princeton, NJ, USA, 1994.
31. Tay, J.K.; Narasimhan, B.; Hastie, T. Elastic Net Regularization Paths for All Generalized Linear Models. *J. Stat. Softw. March* **2023**, *106*, 1–13. [CrossRef] [PubMed]
32. Xing, Y.; Li, D.; Li, C. Time series prediction via elastic net regularization integrating partial autocorrelation. *Appl. Soft Comput.* **2022**, *129*, 109640. [CrossRef]
33. Roozbeh, M.; Arashi, M. Shrinkage ridge regression in partial linear models. *Commun. Stat.—Theory Methods* **2016**, *45*, 6022–6044. [CrossRef]
34. Ghadeer, M.; Nadia, J.M.; Zahraa, A.-S. Regression shrinkage and selection variables via an adaptive elastic net model. *J. Phys. Conf. Ser.* **2021**, *1879*, 032014. [CrossRef]
35. Tibshirani, R. *Regression Shrinkage and Selection*; Oxford University Press: Oxford, UK, 1996; Volume 58, pp. 267–288.
36. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
37. Schmidt, A.B. *Quantitative Finance for Physicists: An Introduction*; Academic Press: Cambridge, MA, USA, 2005.
38. Thenmozhi, M. Support Vector Machines for Prediction of Futures Prices in Indian Stock Market. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, NV, USA, 5–7 April 2004; IEEE: Piscataway, NJ, USA, 2004; Volume 2, pp. 219–223.