Three Essays on Cryptocurrency Analytics and Forecasting by Using Deep Learning Models

Bahareh Amirshahi

A Thesis
In
The John Molson School of Business

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Business Administration) at
Concordia University
Montreal, Quebec, Canada

June 2024

CONCORDIA UNIVERSITY
SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Bahareh Amirshahi

Entitled: Three Essays on Cryptocurrency Analytics and Forecasting by Using Deep Learning Models

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy Business Administration

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Chair
Dr. Juliane Proelss

External Examiner
Dr. Roger Nkambou

Arm's Length Examiner
Dr. Rustam Vahidov

Examiner
Dr. Danielle Morin

Examiner
Dr. Warut Khern-am-nuai

Thesis Supervisor (s)
Dr. Salim Lahmiri

Approved by
Dr. Tracy Hecht Chair of Department or Graduate Program Director

June 3, 2024
Date of Defence

Dr. Anne-Marie Croteau Dean, John Molson School of Business

Abstract

Three Essays on Cryptocurrency Analytics and Forecasting by Using Deep Learning Models

Bahareh Amirshahi, Ph.D.
Concordia University, 2024

Since the emergence of Bitcoin in 2008 as the first cryptocurrency, digital assets have become favored investment options worldwide. Understanding and predicting the behavior of cryptocurrency markets are essential for effective risk management and investment decisions. However, the rapid fluctuations in these markets make accurate predictions a challenging task. In this thesis, key aspects of cryptocurrency market analytics are explored from three different angles. The recurring theme in each study involves proposing hybrid prediction models by combining a feature extractor component with deep learning models to enhance prediction performance.

The first study focuses on predicting cryptocurrency volatility, an underexplored area despite extensive studies in financial markets. By combining traditional econometrics methods with deep learning models, we forecast daily volatility with improved accuracies. The findings revealed that deep learning models not only enhance the accuracy of traditional models but also exhibit superior forecasting when combined with such models in a hybrid approach.

In the second study, we address the challenge of predicting cryptocurrency price values. Recognizing the impracticality of a universal model due to unique cryptocurrency characteristics, we propose a flexible architecture tailored for each cryptocurrency. Additionally, we explore the impact of sentiment data from Twitter posts on prediction accuracy, employing state-of-the-art pre-trained language models in an ensemble manner for more robust sentiment analysis. We show that sentiment data improves the prediction results for more than 70% of the cryptocurrencies studied. This indicates that social media posts, in particular tweets play a significant role in the cryptocurrency markets behavior.

The third study focuses on predicting the direction of cryptocurrency prices. We propose an innovative approach by combining data denoising techniques with machine learning methods, that generates high-quality data for prediction models and achieves significantly higher accuracies. Notably, we assess the proposed approach across distinct periods: before, during, and after the COVID-19 pandemic, filling a critical gap in research regarding predictive models during crisis periods. We show that the predictive performance of the proposed method is not affected by high values of volatility in challenging periods like the COVID-19 pandemic.

This research helps in developing highly accurate prediction models that can aid investors seize profitable opportunities and avoid potential losses. By analyzing over 25 cryptocurrencies, collectively representing 75% of the total market capitalization, this study offers a comprehensive perspective beyond the conventional Bitcoin-centric approach.

Keywords: Cryptocurrency, Forecasting, Machine Learning, Deep Learning, COVID-19

Acknowledgements

Studying for a PhD was not just about pursuing an academic degree for me. It was a journey of personal development, shaping me in ways that would not have happened otherwise. I would like to thank Concordia University for giving me the opportunity to pursue a doctorate. Without their belief in me and financial support, none of this would have been possible.

I would like to express my deepest gratitude to my supervisor, Dr. Salim Lahmiri, for his continuous and timely support, patience, and encouragement throughout this journey. His great flexibility along with his patience with my mistakes made this academic journey a joyful experience. He is truly the best supervisor I could have wished for. I would like to thank the other member of my thesis committee, Dr. Danielle Morin, and Dr. Warut Khern-am-nuai for their insightful comments, questions, and supports from the first steps of writing my PhD candidacy exam till now that I write my thesis.

I am extremely grateful to my parents, who taught me how to read and write when I was just two years old. I thank my brother for his constant love and support, attentively tracking my research and teaching endeavors with curiosity during our phone conversations. My gratitude extends to my friends and other family members for their understanding, encouragement, and support.

I reserve the final words of this acknowledgment for my love, Meyar, who has been by my side since the inception of this journey. Thank you for your sacrifices, support, and encouragements in pursuit of this great accomplishment. Without you, it was not possible to pass the moments of doubts and difficulties.

Contribution of Authors

This thesis has been prepared in “Manuscript-based” format. All the contents that are represented in this thesis either have been published or submitted for publication in the form of journal articles in which the student is the first author. All the papers presented in this thesis were co-authored and reviewed prior to submission for publication by the supervisor Dr. Salim Lahmiri. Dr. Lahmiri helped with the conceptual development of all three papers as well as providing feedback for data collection, analysis, and preparation of the dissertation.

Contents

List of Tables

List of Figures

List of Abbreviations

ADF	Augmented Dickey-Fuller
AE	Autoencoder
AI	Artificial Intelligence
AIC	Akaike Info Criterion
AM	Attention Mechanism
ANFIS	Adaptive Network Fuzzy Inference System
ANN	Artificial Neural Network
AP	Asymmetric Power
APARCH	Asymmetric Power ARCH
APGARCH	Asymmetric Power GARCH
AR	Autoregressive
ARCH	Autoregressive Conditional Heteroskedasticity
ARIMA	Autoregressive Integrated Moving Average
ARMA	Autoregressive Moving Average
BERT	Bidirectional Encoder Representations from Transformers
BRNN	Bayesian Regularization Neural Network
CGARCH	Component Generalized Autoregressive Conditional Heteroskedasticity
CNN	Convolutional Neural Network
COVID	Coronavirus Disease
CV	Cross Validation
DFFNN	Deep Feedforward Neural Network
DFNN	Feedforward Neural Network
DL	Deep Learning
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
EGARCH	Exponential GARCH
ELM	Extreme Learning Machine
EMA	Exponential Moving Average
EPU	Economic Policy Uncertainty
EWMA	Exponentially Weighted Moving Average
FCN	Fully Convolutional Network
FFNN	Feedforward Neural Network
FNN	Fully Connected Neural Network
GA	Genetic Algorithm
GARCH	Generalized Autoregressive Conditional Heteroskedasticity
GB	Gradient Boosting
GBM	Gradient Boosting Machine
GED	Generalized Error Distribution
GJR-GARCH	Glosten-Jagannathan-Runkle GARCH
GRP	Gaussian Poisson Regression
GRU	Gated Recurrent Unit

HONN	Higher Order Neural Network
HV	Historical Volatility
IGARCH	Integrated GARCH
JB	Jarque-Bera
KOPSI	Korean Stock Price Index
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Linear Discriminant Analysis
LGARCH	Logistic GARCH
LM	Lagrange-Multiplier
LR	Logistic Regression
LSTM	Long Short-Term Memory
MACD	Moving Average Convergence Divergence
MALSTM	Multivariate Attention Long Short-Term Memory
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NLP	Natural Language Processing
OHLC	Open, High, Low, and Close prices
OHLCV	Open, High, Low, and Close prices, trading Volume
OHLCVA	Open, High, Low, and Close prices, trading Volume, trading Amount
OLS	Ordinary Least Squares
PCA	Principal Component Analysis
PP	Philips-Perron
QDA	Quadratic Discriminant Analysis
RBFNN	Radial Basis Function Neural Network
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
RT	Regression Tree
SGD	Stochastic Gradient Descent
SMA	Simple Moving Average
SVM	Support Vector Machine
SVR	Support Vector Regression
TGARCH	Threshold GARCH
TI	Technical Indicator
VAD	Valence, Arousal, Dominance
VADER	Valence Aware Dictionary and sEntiment Reasoner
WAMC	Weighted and Attentive Memory Channels
WHO	World Health Organization
XGB	eXtreme Gradient Boosting

Thesis Introduction

According to Merriam Webster's definition, "Cryptocurrency" is defined as "any form of currency that only exists digitally, that usually has no central issuing or regulating authority but instead uses a decentralized system to record transactions and manage the issuance of new units"[1]. Since the Bitcoin's introduction by Nakamoto (2008) as the first cryptocurrency, these digital assets have been a favoured investment target of investors all over the world. As of November 2023, there exists over 8000 cryptocurrencies with the global market capitalization of over $1.4 trillion[2]. The cryptocurrency market is dynamic, and new cryptocurrencies are frequently introduced, while others may lose relevance.

The need to understand and predict the fundamental aspects of cryptocurrencies is vital as it offers implications for risk management and investment strategies. Cryptocurrency markets are known for their rapid fluctuations that makes accurate predictions a challenging task. These forecasts may involve predicting the directional movement of cryptocurrency prices (a classification task), forecasting specific price values (a regression task), and forecasting volatility levels (a regression task). In this thesis, we study these three interconnected facets of cryptocurrency market analytics, approaching the subject from three distinct angles. The goal is to create innovative models that provides highly accurate predictions that not only adds to the existing knowledge but also assists investors and decision-makers for better investment strategies and risk assessment.

Recently, with the increasing use of Artificial Intelligence (AI) in various fields, its application in cryptocurrency markets has been a developing stream of research. In contrast to traditional linear models that assume linear relationships between various factors influencing cryptocurrencies' prices, the AI approach demonstrates a unique capability to capture the non-linear patterns inherent in cryptocurrencies' behavior by automatically extracting relevant features from raw data. In this regard, deep learning (DL) methods, a subset of AI, have been widely used by researchers of this domain (Cavalli & Amoretti, 2021; Lahmiri & Bekiros, 2019, 2021; Ortu et al., 2022a; Oyedele et al., 2023a; Shintate & Pichl, 2019). Deep learning methods involve the use of neural networks, particularly deep neural networks, to model and solve complex problems.

[1] https://www.merriam-webster.com/dictionary/cryptocurrency
[2] https://www.statista.com/statistics/730876/cryptocurrency-maket-value/

Neural networks are computational models inspired by the structure and functioning of the human brain. These networks consist of layers of interconnected nodes (neurons) that process and transform input data to produce output. The term "deep" in deep learning refers to the use of multiple layers (deep architectures) in neural networks.

This thesis contains three essays as follows:

Essay 1: In the first essay, the main objective is to improve the prediction performance of traditional econometric methods in volatility forecasting for cryptocurrency markets. The key hypotheses investigated in this study are as follows:

a) Comparison of various Generalized Autoregressive Conditional Heteroskedasticity (GARCH) models, collectively referred to as GARCH-type models – widely used econometric models for volatility forecasting in financial markets. We investigate which model is the best at forecasting cryptocurrency volatility for out-of-sample data.
b) Investigation and comparison of the performance of two prominent deep learning models – the Deep Feedforward Neural Network (DFFNN) and Long Short-Term Memory (LSTM) models. The focus is on evaluating their efficacy in enhancing forecasts generated by GARCH-type models.
c) Examination of whether incorporating GARCH-type forecasts into the DFFNN and LSTM models improves forecasting accuracies. This involves utilizing GARCH-type models as feature extractors for deep learning models, referred to as hybrid models.

Essay 2: The second essay has been inspired by two streams of studies in the literature: 1) studies that utilize neural networks, especially the Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) (Patel et al., 2020; Chen et al., 2021; Zoumpekas et al., 2020; Ji et al., 2019; Zhang et al., 2021; Lahmiri & Bekiros, 2020; Oyedele et al., 2023), 2) studies that investigate the effectiveness of social media data in predicting cryptocurrency prices (Kraaijeveld & de Smedt, 2020; Ortu et al., 2022; Anbaee Farimani et al., 2022; Zou & Herremans, 2022). We chose to utilize Twitter data because this social media platform is the most popular one among traders and provides a combination of both news and investor sentiment simultaneously (Kraaijeveld & de Smedt, 2020). We propose a hybrid model based on two popular deep learning models, namely LSTM and CNN and the attention mechanism. The proposed model has a flexible architecture that can be customized for a specific cryptocurrency for higher prediction accuracies. The objective of this study is not to find one unique best model that performs well for all cryptocurrencies and various lengths of input sequences. Due to the highly volatile nature of the cryptocurrency markets, finding a model with consistently high performance for all cryptocurrencies is almost impossible. Instead, we aim to investigate the different architectures of our proposed models and find the optimal architecture and input layer configuration for each specific cryptocurrency.

The main hypotheses that are investigated in this study are as follows:

a) Does the inclusion of sentiment data impact the prediction performance of the models?
b) Is presence of a CNN layer helpful in improving the prediction performance of the models?
c) How does the incorporation of an attention layer affect the prediction performance of the models?
d) What is the optimal length of the input sequence for effective model performance?
e) How should the input layer be configured for each cryptocurrency to maximize the prediction performance?

Essay 3: The third essay focusses on predicting whether the next day's price of cryptocurrencies will rise or fall. We utilize historical price data and technical indicators as the initial feature sets. To generate more informative features, two feature extraction methods are employed in parallel: 1) an Autoencoder model with LSTM layers in place of traditional fully connected layers in the encoder and decoder parts (LSTM-Autoencoder), 2) a signal denoising technique called Discrete Wavelet Transform (DWT). While this technique has been proven to enhance predictions in stock markets (Ji et al., 2022; Wu et al., 2022a), we apply this technique to cryptocurrency data to remove noise or abrupt fluctuations from the initial features. The combined features from these methods create the final feature set for a Gated Recurrent Unit (GRU) model, which predicts daily price movement directions. The main hypothesis is to evaluate whether this feature set significantly improves predictions of the GRU or any other deep learning model compared to base cases in which no feature extraction method is utilized. The study also explores the approach's effectiveness during different periods, including volatile situations like the COVID-19 pandemic, which presents challenges for accurate price direction predictions.

By employing the state-of-the-art DL techniques, we address main existing gaps and make significant contributions to the growing body of knowledge about cryptocurrency markets:

i. Developing a reliable and accurate model that can perform well in forecasting all cryptocurrency markets is a challenging task due to the unique characteristics of each market. Previous research has often been limited to evaluating models primarily on Bitcoin, given its dominant market capitalization (Chen et al., 2021; Ji et al., 2019; Lahmiri & Bekiros, 2020; Zou & Herremans, 2022). In this regard, my approach stands out as I assess the performance of the models across a comprehensive dataset encompassing 27 cryptocurrencies. This dataset represents 75% of the total market cap as of August 2023.

ii. As another solution to the aforementioned issue, the second study introduces a new prediction model with a flexible architecture. This adaptable framework can be customized for each specific cryptocurrency, enhancing the accuracy of price predictions.

iii. While DL models can extract deep hidden patterns in raw data, there is substantial evidence that incorporating high-quality features extracted through alternative methods can elevate their predictive performance. Accordingly, we adopt a consistent strategy across all three studies by creating hybrid models that combine feature extraction components with DL prediction models. The main objective in all three studies is to improve the predictive capabilities of DL models by incorporating informative features that are derived from the feature extractor component of the proposed hybrid models. Across all three studies, historical data forms the foundational feature set. However, the first study augments this with features derived from traditional statistical models, the second study leverages social media data for feature generation, and the third study introduces a pioneering technique in which historical data and technical indicators undergo refinement through a process that filters out noise – a novel approach in the prediction of cryptocurrency market behavior.

iv. Financial markets experience extreme volatility during the crisis periods such as the recent one caused by COVID-19 pandemic. Cryptocurrency prices under high volatility exhibit rapid and unpredictable changes, making it challenging for models to adapt and capture these fleeting patterns accurately. In the third study, the effectiveness of the proposed approach is assessed across different challenging periods, including the periods before, during, and after the COVID-19 pandemic, which were marked by varying levels of cryptocurrency market volatility.

Essay 1: Hybrid Deep Learning and GARCH-Family Models for Forecasting Volatility of Cryptocurrencies

Abstract

The combination of deep learning and GARCH-type models has been proven to be superior to the single models in forecasting volatility across various markets, including energy, main metals, and especially stock markets. To verify this hypothesis for cryptocurrency markets, we constructed various deep learning models based on Feed Forward Neural Networks (DFFNNs) and Long Short-Term Memory (LSTM) networks, evaluating their performance in forecasting the volatility of 27 cryptocurrencies. Then, different hybrid models were built in which the outputs of three GARCH-type models, namely GARCH, EGARCH, and APGARCH, with three different assumptions for the residuals' distribution were fed into the DFFNN and LSTM networks. In other words, GARCH-type models were utilized as feature extractors and the deep learning models leveraged a sequence of extracted features as inputs to produce the volatility for the next day. Our findings revealed that not only do deep learning models improve the forecasts of GARCH-type models under any distribution assumption, but also the forecasts of GARCH-type models, when used as informative features, significantly increase the predictive power of the studied deep learning models, specifically DFFNN and LSTM models.

Keywords: Deep learning, GARCH-family models, Cryptocurrency, Volatility, Statistical distribution

1. Introduction and literature review

In quantitative finance, volatility refers to the conditional standard deviation (or conditional variance) of the underlying asset returns (Lahmiri et al., 2018). Among various financial markets, the rapid growth of the cryptocurrency market, its high volatility and its applications in different commercial transactions have attracted the attention of academics and investors (Tapia & Kristjanpoller, 2022). Like other financial time series, cryptocurrencies are heteroscedastic, and forecasting their volatility as a fundamental variable in risk management is a challenging task.

From a financial econometric point of view, the literature on volatility forecasting mostly consists of Autoregressive Conditional Heteroskedasticity (ARCH) and their generalization, known as GARCH-type models. Such times series models with heteroscedastic errors are specifically applicable to modeling financial market data which are highly volatile (Hajizadeh et al., 2012). The GARCH model, introduced by Bollerslev (1986) as a generalization of ARCH model (Engle, 1982), is one of the most popular models for forecasting time series volatility. The GARCH model is a symmetric model in which conditional variance is determined based on squared values of both residuals and conditional variances from previous periods. Volatility tends to increase more after a negative shock than after a positive shock of the same magnitude (Yu, 2019) — a phenomenon known as the "leverage effect". The GARCH model is not able to model leverage effect because its specifications assume that the variance depends on the shock's magnitude, and it is independent of the shock's sign (Khaldi et al., 2019). Later, several models with more flexibility were introduced being able to capture asymmetric behavior of the time series by incorporating new parameters into the GARCH models. Two such models employed in this study are the exponential GARCH (EGARCH) (Nelson, 1991) and Asymmetric Power GARCH (APGARCH) models (Ding et al., 1993).

Recently, with the increasing use of Artificial Intelligence (AI) in various fields, the application of such models has become a developing stream of research in financial markets, particularly in stocks and cryptocurrency markets. Numerous studies have demonstrated that Artificial Neural Network (ANN) Models outperform traditional GARCH-type models in volatility predictions since they can capture the non-linearity of the series and do not require the series to be stationary for modeling (Tapia & Kristjanpoller, 2022). More specifically, ANN models can approximate any continuous functions (Hornik et al., 1989) without requiring restrictive assumptions on the underlying process from which data are generated (D'Amato et al., 2022).

A comparative study of GARCH-type models as parametric models and deep learning models as non-parametric models for volatility forecasting was conducted by Khaldi et al. (2019). In their study, seven GARCH-type models, namely GARCH, EGARCH, APARCH, TGARCH (Zakoian, 1994), GJR-GARCH (Glosten et al., 1993), CGARCH (Engle & Lee, 1999), and IGARCH (Engle & Bollerslev, 1986) were analyzed in terms of their ability to forecast Bitcoin volatility. Assuming four different distributions for the standardized residuals, the TGARCH model was identified as the best parametric approach among all seven GARCH-type models. The authors evaluated two deep learning models, namely Multilayer Perceptron (MLP) and Recurrent

Neural Network (RNN). They demonstrated the superiority of the MLP, and its comparison with the TGARCH model revealed that it outperforms the TGARCH as well.

In line with this literature, hybrids of deep learning and GARCH-type models are typically found to exhibit better performance compared to single deep learning or time series models. As an example, in the context of the metal commodities market, Kristjanpoller and Hernández (2017) utilized forecasts from the best GARCH models along with some other financial variables as inputs for the ANN model to forecast the volatility of gold, silver, and copper. Their results revealed that the GARCH model incorporating exogenous variables, provided superior forecasting compared to the GARCH model that is fitted only with the studied variable returns. Moreover, they showed that the hybrid model significantly improves forecasts in comparison to the best forecasts of the GARCH models.

The hybridization of two deep learning models can be observed in the work of Vidal and Kristjanpoller (2020), where they employed a specially pretrained Convolutional Neural Network (CNN) model to extract a greater amount of information from gold return data. They combined these results with the outcomes of a Long Short-Term Memory (LSTM) network to generate volatility forecasts. Their hybrid model improved gold volatility forecasts compared to benchmark methods such as GARCH, Support Vector Regression (SVR), ANN, ANN-GARCH, LSTM and CNN.

Apart from the gold market, the effectiveness of hybrid models in forecasting coper volatility was demonstrated in the study by Hu et al. (2020). The authors found that not only Neural Network models can outperform the GARCH model, but also incorporating GARCH forecasts as inputs enhances the prediction power of those models. Additionally, they realized that incorporating the LSTM and Bidirectional LSTM models into the hybrid of ANN and GARCH models improves the volatility predictions for copper prices.

The application of hybrid models in forecasting the volatility of crude oil can be found in the work of Kristjanpoller and Minutolo (2016). In their study, forecasts from the GARCH model, along with financial time series data (exchange rate and the stock market index), were used as inputs in the ANN model. The authors concluded that the volatility forecasts obtained by the hybrid model showed improvement over traditional forecasting models.

In the context of the stock market, Ramos-Pérez et al. (2019) proposed a two-level stacked model based on various machine learning models to predict the volatility of the S&P500. They utilized Random Forest (RF), Gradient Boosting (GB) with regression trees, and Support Vector Machine (SVM) at the first level, and an ANN model at the second level. Their stacked model was found to be superior to hybrid models built on GARCH, EGARCH, and ANN models. In the same context, Liu (2019) proved that the LSTM and v-SVR model can outperform the GARCH model for a large time interval forecasting of volatility for S&P500 and AAPL.

Kim and Won (2018) constructed a hybrid model by combining the LSTM model with GARCH-type models to forecast the volatility of the Korean stock price index (KOPSI 200). The novelty of their work lies in using estimated parameters of two or more GARCH-type models as inputs for the LSTM model, instead of using GARCH-type forecasts. Their results showed that their GEW-LSTM model, which combines GARCH, EGARCH, and Exponentially Weighted Moving Average (EWMA) models with the LSTM model, has the lowest prediction errors. Kristjanpoller et al. (2014) tested the hybrid of ANN and GARCH model in forecasting three Latin-American stock exchange indexes from Brazil, Chile, and Mexico. They used a mean test and showed a significant improvement in forecast accuracy compared to the GARCH model.

The use of hybrid models in predicting cryptocurrencies' volatility was introduced by Kristjanpoller and Minutolo (2018). Analyzing GARCH, EGARCH, and APGARCH models in their study, EGARCH was identified as the best model. The authors evaluated 12 hybrid models using different combinations of EGARCH forecasts, Bitcoin's price lagged values, technical indicators, and the principal components derived from technical indicators as inputs for the ANN model. In all cases, the hybrid model improved the results of the best GARCH model. Seo and Kim (2020) adopted regular ANN and Higher Order Neural Network (HONN) to construct hybrid models for predicting Bitcoin's volatility. They used outputs of GARCH-type models along with lagged values of realized volatility and some other relevant variables as the inputs data for their ANN and HONN models. They showed that the hybrid models based on HONN provide more accurate forecasts than the other models.

Recently, Aras (2021) proposed a new stacking ensemble model for forecasting Bitcoin's volatility. This ensemble model was constructed with five base learners and an SVM model served as a meta learner, wherein the forecasts of the formers were used as inputs for the latter. The base learners include the standard GARCH model along with four hybrid models: SVM-GARCH, ANN-GARCH, K-Nearest Neighbor-GARCH, and RF-GARCH. Features obtained through feature selection methods such as LASSO (Tibshirani, 1996), Random Forest Selector, and Principal Component Analysis (PCA) were fed into the meta learner, i.e., the SVM model. The results indicated that the hybrid models outperformed the standard GARCH model, and among those hybrids, the SVM-GARCH was found to be superior to the others. Furthermore, among the five variants of the ensemble framework, the one with the LASSO feature selector was selected as the best model.

One interesting hybrid model was found in the work of Peng et al. (2018), where they use an SVR model to estimate the mean and volatility equations of conventional GARCH model. Their study demonstrated that for all exchange rates and cryptocurrencies considered in their study, across both high and low-frequency datasets, the SVR-GARCH model achieved the lowest prediction errors compared to other GARCH type models with various assumptions for the residual's distributions. The reason for this could be the high ability of kernel functions in capturing the non-linearities.

Regarding the leverage effect mentioned earlier, Yu (2019) investigated the effects of economic policy uncertainty (EPU) and leverage effect on Bitcoin's volatility using high-frequency data. The conclusion was that considering both the EPU and leverage effect can aid in forecasting Bitcoin's volatility. More specifically, their results show that the leverage effect has more power than jumps components in forecasting Bitcoin's volatility.

While there are many studies in volatility forecasting, only a few of them focus on predicting the volatility of cryptocurrencies. Furthermore, most studies related to cryptocurrency markets have analyzed Bitcoin's data, given its largest market capitalization. To the best of our knowledge, our study is the first one to utilize an extensive set of cryptocurrency data to demonstrate the ability of deep learning models in volatility forecasting.

Our contributions are summarized as follows:

i. We investigate which GARCH-type model is the best at forecasting cryptocurrencies' volatility for out-of-sample data.
ii. We investigate and compare the performance of deep learning models, namely the common DFFNN and LSTM models, in improving the forecasts of GARCH-type models.

iii. We investigate whether adding GARCH-type forecasts to the DFFNN and LSTM models improves forecasting accuracies.
iv. We use a large dataset composed of 27 cryptocurrencies to draw general and robust conclusions.
v. We apply formal statistical tests to check for differences between the performances of the models.

Table 1: Recent studies on volatility forecasting in various markets

Reference	Market / Data	Methodology	Outcome
Khaldi et al. (2019)	Bitcoin	1- GARCH, EGARCH, APARCH, TGARCH, GJR-GARCH, CGARCH, and IGARCH. 2- MLP and RNN	1- TGARCH model was found to be the best parametric approach. 2- MLP outperformed the TGARCH model.
Kristjanpoller and Hernández (2017)	Gold, silver, and copper	ANN model fed by forecasts of best GARCH models along with some other financial variables.	The hybrid model improved forecasts significantly in comparison to the best forecast of the GARCH models.
Vidal and Kristjanpoller (2020)	Gold	Hybrid of CNN and LSTM models.	The hybrid model improved the volatility forecasts obtained by benchmark methods such as GARCH, SVR, ANN, ANN-GARCH, LSTM and CNN.
Hu et al. (2020)	Copper	Hybrid of ANN and GARCH models.	Incorporating GARCH forecasts as inputs can enhance the prediction power of ANN models.
Kristjanpoller and Minutolo (2016)	Crude oil	The forecasts from the GARCH model, along with financial time series data (exchange rate and the stock market index) were used as inputs for the ANN model.	Volatility forecasts obtained by hybrid model were improved over traditional forecasting models.
Ramos-Pérez et al. (2019)	S&P500	A two-level stacked model: RF, GB with regression trees and SVM in the first level, and an ANN model within the second level.	The stacked model was found to be superior to the hybrid models built on GARCH, EGARCH, and ANN models.
Liu (2019)	S&P500 and AAPL	LSTM and v-SVR models.	LSTM and v-SVR can outperform the GARCH model.
Kim and Won (2018)	Korean stock price index (KOPSI 200)	A hybrid model by combining the LSTM model with GARCH-type models.	The GEW-LSTM model which combines GARCH, EGARCH, and exponentially weighted moving average (EWMA) models with LSTM model has the lowest prediction errors.
Kristjanpoller et al. (2014)	Three stock exchange indexes from Brazil, Chile, and Mexico.	Hybrid of ANN and GARCH model.	A significant improvement of the forecast accuracy was observed compared to the GARCH model.
Kristjanpoller and Minutolo (2018)	Bitcoin	12 hybrid models using different combinations of EGARCH forecasts, Bitcoin price lagged values, technical indicators and the principal components derived from technical indicators as inputs for the ANN model.	Analyzing GARCH, EGARCH, and APGARCH models in their study, EGARCH was found as the best model. In all cases, the hybrid model improved the results of the best GARCH model.

Seo and Kim (2020)	Bitcoin	ANN and HONN models fed by outputs of GARCH-type models along with lagged values of realized volatility and some other relevant variables.	The hybrid models based on HONN provided more accurate forecasts than the other models.
Aras (2021)	Bitcoin	A new stacking ensemble model built by five base learners and a SVM model serving as a meta learner in which the forecasts of the formers were used as inputs for the latter.	The hybrid models outperformed the standard GARCH model and among those hybrids, the SVM-GARCH was found to be superior to the other ones.
Peng et al. (2018)	Exchange rates and cryptocurrencies	A SVR model to estimate the mean and volatility equations of conventional GARCH model.	The SVR-GARCH model achieved the lowest prediction errors compared to other GARCH-type models with various assumptions for the residuals' distributions.

2. Data

In the first step of data collection, a list of 100 most traded cryptocurrencies was selected based on data available on the website "www.investing.com". The cryptocurrencies were then arranged according to their entrance date into the market, and the 30 oldest ones were chosen for analysis in this study. The 30 cryptocurrencies are as follows: Bitcoin, XRP (Ripple), Monero, Ethereum, Ethereum Classic, Litecoin, Zcash, Stellar, Dash, Tether, Dogecoin, IOTA, EOS, OMG Network, Bitcoin Cash, Waves, Neo, VeChain, Binance Coin, TRON, Cardano, Maker, Filecoin, Aave, Decentraland, Chainlink, Tezos, Dai, Enjin Coin, Theta.

To ensure an equal amount of data for all cryptocurrencies, we considered the market entrance date of the newest cryptocurrency (THETA), i.e., "2018-02-07" as the starting date for all cryptocurrency datasets. Using "2022-02-07" as the end date, we collected four years of data for each cryptocurrency from "www.investing.com", including values for "Open price", "Close price", "High price", "Low price", and "Volume". We selected the "Close price" variable for our analysis, as it is the most commonly used variable in almost all similar studies (see, e.g., Kristjanpoller & Minutolo, 2018; Khaldi et al., 2019; Seo & Kim, 2020; Peng et al., 2018; Hu et al., 2020; Hajizadeh et al., 2012).

Five of the datasets were found to contain a few missing values which in worst case, it was approximately 2% of the data. To address this, we replaced the missing values with the last available values, resulting in 1462 observations for each cryptocurrency. Among the 30 cryptocurrencies, three of them ("Aave", "Dogecoin", and "VeChain") exhibited behavior resembling white noise series, making them unpredictable by our models. Consequently, 27 cryptocurrencies were selected as the final datasets for analysis in this study.

Given that investors are typically more interested in price returns (i.e., price variation) rather than absolute price itself (see, e.g., Kristjanpoller & Minutolo, 2018; Khaldi et al., 2019; Seo & Kim, 2020; Peng et al., 2018; Hu et al., 2020; Hajizadeh et al., 2012), and considering that return data is often stationary, making it suitable for use in time series models, we decided to conduct our experiments with this variable. Therefore, price returns were calculated using Equation (1):

$$r_t = \log(P_t) - \log(P_{t-1}) \quad (1)$$

where P_t is Close price of the cryptocurrencies at time t.

Table 2 displays descriptive statistics, stationarity, normality, and heteroscedasticity test results for the price returns of each cryptocurrency. High values of Kurtosis indicate that fat-tailed distributions are necessary to accurately describe the conditional distribution of the returns (Kristjanpoller & Minutolo, 2015). Two-thirds of the cryptocurrencies exhibit negative skewness values for the price returns, suggesting a high probability of supplying negative values in return distributions (Khaldi et al., 2019). Both Augmented Dickey Fuller (ADF) and Philips-Perron (PP) test results indicate stationarity of price returns for all cryptocurrencies. The Jarque-Bera (JB) statistic is significant at the 1% level of significance, indicating that the price returns are not normally distributed for all cases. The results of the Lagrange-Multiplier test (LM (12)) show the presence of ARCH effects in the Ordinary Least Squares (OLS) residuals from the regression of the returns on a constant data. Specifically, the null hypothesis of LM (12) indicates no conditional heteroscedasticity until the twelfth lag. The rejection of this hypothesis at a significance level of 1% for almost all cryptocurrencies verifies the existence of heteroscedasticity until the twelfth lag. Therefore, volatility clustering is present, and GARCH-type models are deemed appropriate for use in this study. This implies that when volatility is high, this highly volatile state is maintained to some extent, and when volatility is low, this low volatility state is maintained up to a certain threshold (H. Y. Kim & Won, 2018).

Table 2: Descriptive statistics of price returns

Cryptocurrency	**Number of Observations**	**Mean**	**Standard Deviation**	**Kurtosis**	**Skewness**	**PP statistics**	**ADF statistics**	**JB statistics**	**LM statistics**
Binance Coin	1461	0.00272	0.05721	16.00	-0.29	-10.76***	-41.63***	15604***	98.74***
Bitcoin	1461	0.00120	0.03968	18.38	-1.32	-17.80***	-41.44***	20992***	20.35*
Bitcoin Cash	1461	-0.00070	0.06406	13.44	-0.40	-9.10***	-39.90***	11036***	36.49***
Cardano	1461	0.00088	0.06024	6.17	-0.23	-17.56***	-41.26***	2330***	43.62***
Chainlink	1461	0.00267	0.07289	7.25	-0.31	-17.80***	-41.95***	3224***	35.56***
Dai	1461	0.00000	0.00806	13.23	-0.22	-12.80***	-85.29***	10666***	278.62***
Dash	1461	-0.00103	0.06559	25.98	0.01	-44.17***	-43.82***	41102***	220.81***
Decentraland	1461	0.00242	0.07660	22.85	1.34	-18.71***	-38.70***	32223***	67.52***
Enjin Coin	1461	0.00190	0.11099	17.43	0.14	-25.40***	-51.55***	18489***	197.95***
EOS	1461	-0.00072	0.06422	10.42	-0.37	-27.14***	-43.17***	6645***	100.29***
Ethereum	1461	0.00098	0.05191	13.82	-1.26	-11.10***	-41.99***	12007***	35.63***
Ethereum Classic	1461	0.00040	0.06149	10.82	-0.27	-16.96***	-39.53***	7148***	101.64***
Filecoin	1461	0.00017	0.06356	29.45	2.02	-10.42***	-41.70***	53780***	79.21***
IOTA	1461	-0.00032	0.06250	9.33	-0.74	-25.50***	-41.14***	5434***	32.48***
Litecoin	1461	0.00000	0.05546	9.20	-0.71	-18.06***	-41.05***	5271***	63.21***
Maker	1461	0.00063	0.06991	27.61	-1.09	-14.75***	-45.01***	46707***	67.28***
Monero	1461	-0.00007	0.05405	14.52	-1.29	-17.40***	-44.54***	13242***	59.69***
Neo	1461	-0.00092	0.06444	8.65	-0.31	-18.50***	-42.16***	4584***	66.51***
OMG Network	1461	-0.00048	0.07002	8.65	-0.09	-16.90***	-40.84***	4562***	74.18***
Stellar	1461	-0.00022	0.05940	10.55	0.48	-40.20***	-40.15***	6836***	37.93***
Tether	1461	0.00000	0.00250	24.18	0.20	-12.06***	-53.08***	35612***	403.85***
Tezos	1461	0.00033	0.06857	7.81	-0.66	-12.46***	-39.50***	3818***	62.28***
Theta	1461	0.00231	0.07569	8.02	-0.18	-18.33***	-42.19***	3919***	30.53***
TRON	1461	0.00050	0.05984	8.79	-0.54	-26.95***	-40.80***	4773***	44.22***
Waves	1461	0.00054	0.06466	8.38	0.13	-14.12***	-40.84***	4282***	53.43***
XRP	1461	0.00010	0.06093	13.18	0.11	-39.33***	-39.34***	10579***	69.62***
Zcash	1461	-0.00069	0.06047	8.11	-0.80	-12.23***	-41.63***	4163***	70.01***

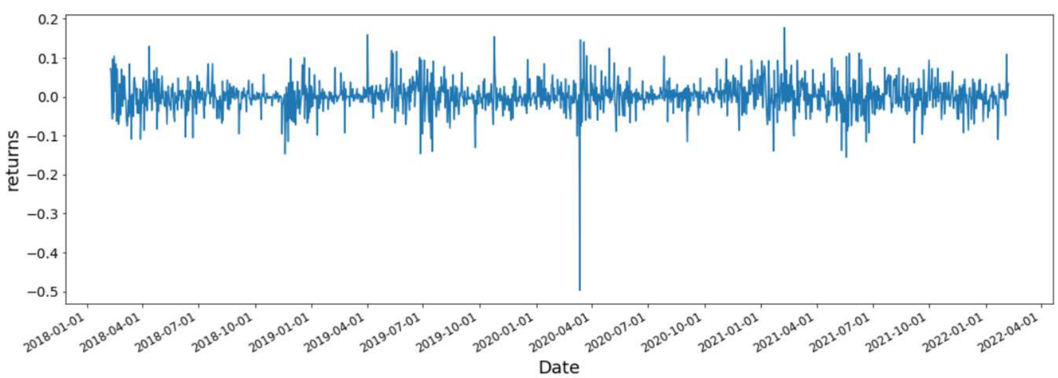
***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

To assess the performance of our models, a comparison between predicted and actual values of volatility is essential. However, volatility is an unobservable variable and cannot be directly measured. As a proxy for volatility, we utilize standard deviation since it indicates the extent of variation in a cryptocurrency's value during a specific period. Therefore, historical volatility at time t is calculated as the standard deviation of the returns' values using Equation (2).

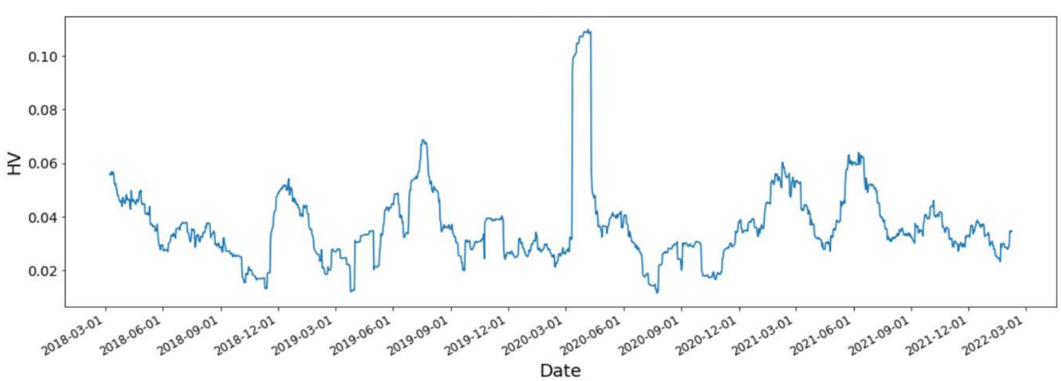
$$HV_t = \sqrt{\frac{1}{T}\sum_{i=t}^{t-T}(r_i - \overline{r_t})^2} \tag{2}$$

where r_i represents the price return of the cryptocurrency on day i, and $\overline{r_t}$ is the average return of the cryptocurrency over the past T trading days. In this study, $T = 30$ (approximately one month of transactions) will be considered. Figure 1 presents price returns and historical volatilities of Bitcoin as an example cryptocurrency over the period of the study.

Figure 1: a) Daily returns, b) Historical Volatilities (HVs) for Bitcoin's data as an example



(a)



(b)

Each dataset is divided into two parts: the initial 80%, from 2018-02-07 to 2021-04-23, as the training sample, while the remaining 20% from 2021-04-24 to 2022-02-07 as the testing sample. For deep learning models, where HV (historical volatility) is employed to train the models, the training sample begins from 2018-03-09, as the first 30 values of HV are lost due to the utilization of Equation (2). Moreover, for the purpose of hyper-parameter tuning for deep learning models, the last 20% of the training sample is used as a validation sample.

3. Methodology

3.1. Time Series models

We employ three GARCH-type models as our time series models, specifically GARCH(p, q) (Engle, 1982), EGARCH(p, q) (Nelson, 1991), and APGARCH(p, q) (Ding et al., 1993). The conditional variance equations for these models are described in Equations (3), (4), and (5). In each equation, σ_t^2 denotes the conditional variance of the return series. $\hat{r}_t = \hat{\mu}_t + \epsilon_t$ is called the mean equation with $\hat{\mu}_t$ as the conditional mean of the return series. $\epsilon_t = \hat{\sigma}_t z_t$ is the heteroscedastic error term, and z_t is a random variable with a mean of zero and a unit variance. Moreover, ω, α_i, β_i, and γ_i are the parameters to be estimated through the maximization of the sample log-likelihood function. The distribution of the return series depends on the distribution of the standardized residuals, i.e., z_t. Three different distributions that are considered in this study are Normal, Student's t, and Generalized Error Distribution (GED).

$$\hat{\sigma}_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2 \tag{3}$$

As Equation (3) indicates, the conditional variance of price returns depends on both the square of the errors (ϵ_{t-i}^2) as well as the lagged conditional variances (σ_{t-j}^2).

As mentioned in the introduction section, the GARCH model fails to capture the leverage effect of the volatility. The EGARCH model overcomes this limitation by incorporating a new parameter, γ_i that captures the asymmetric impact of negative shocks in the news:

$$\log \hat{\sigma}_t^2 = \omega + \sum_{i=1}^{p} (\alpha_i \epsilon_{t-1} + \gamma_i(|\epsilon_{t-1}| - E|\epsilon_{t-1}|)) + \sum_{j=1}^{q} \beta_j \log \hat{\sigma}_{t-j}^2 \tag{4}$$

The APGARCH model described in Equation (5) is a comprehensive model that encompasses other GARCH-type models. Although the parameter δ can be estimated from data, for the sake of simplicity in the analysis, we set $\delta = 1$.

$$\hat{\sigma}_t^{\delta} = \omega + \sum_{i=1}^{p} \alpha_i (|\epsilon_{t-1}| - \gamma_i \epsilon_{t-i})^{\delta} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^{\delta} \tag{5}$$

3.2. Deep Learning models

3.2.1. Deep Feed Forward Neural Network (DFFNN)

The Deep Feed Forward Neural Network (DFFNN) is one of the most widely used machine learning models in volatility prediction. A DFFNN model is illustrated in Figure 2, featuring one input layer with m predictors, x_i, four hidden layers, each having n neurons, where each neuron performs a non-linear transformation of outputs from the previous layer. Finally, one output layer is responsible for generating the value of y, corresponding to the predicted historical volatility. It is worth mentioning that in this study, various architectures of this model are explored, including variations in the number of hidden layers, the number of neurons in each layer, and more to determine the most effective configuration for each cryptocurrency's data. Equation (6) computes y:

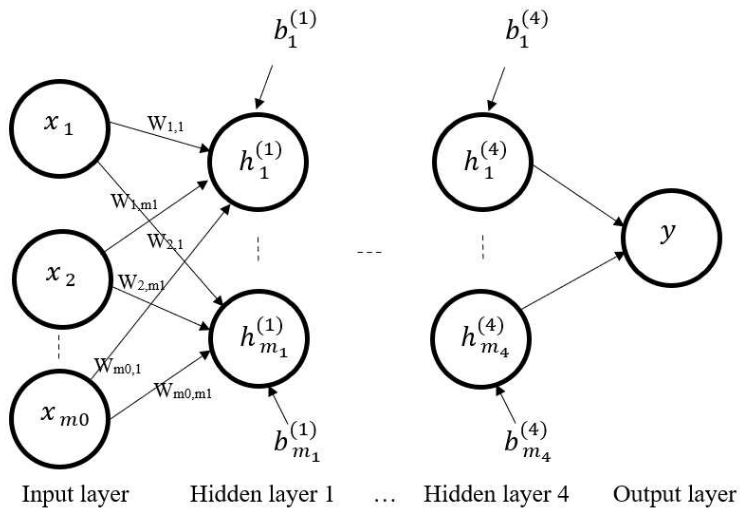
$$y = f_l(\dots f_1\left(\sum_{j=1}^{m_1} w_{jk}^{(1)} . f_0\left(\sum_{i=1}^{m_0} w_{ij}^{(0)} . x_i + b_j^{(0)}\right) + b_k^{(1)}\right) \tag{6}$$

where:

- w_{ij} represents the weight or parameter connecting neuron j in a layer to neuron i from the previous layer,
- $b_j^{(l)}$ denotes the bias term for neuron j in layer l,
- m_l is the number of neurons in layer l,
- $f_l(.)$ is the activation function in layer l and it controls the magnitude of the outputs from each neuron.

The training process of the DFFNN involves the optimization algorithm (backpropagation), which iteratively adjusts the unknown weights and biases to minimize the value of a loss function. For a more in-depth understanding, please refer to these papers: (Seo & Kim, 2020) and (Tapia & Kristjanpoller, 2022).

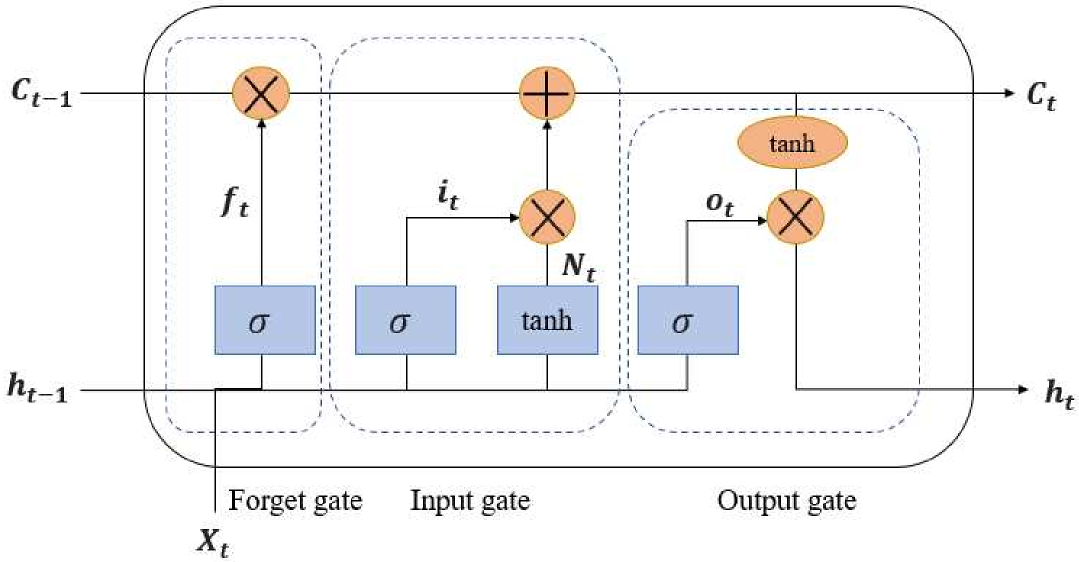
Figure 2: A four-layer DFFNN model



3.2.2. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks, a variant of Recurrent Neural Networks (RNN), were introduced by Hochreiter & Schmidhuber (1997). As explained by Bengio et al. (1994), a task displays long-term dependencies if predicting the desired output at time *t* depends on input presented at an earlier time *T*. Gradient-based learning algorithms such as RNNs face difficulties in performing such tasks and their parameters settle in sub-optimal solutions that take into account short-term dependencies but not long-term dependencies. LSTM networks allow the information to persist in the network and could learn both short and long-term dependencies in the input sequence. Figure 3 shows the components of a typical LSTM unit which includes the cell state (*C*), hidden state (*h*), and three gates: forget, input, and output gates. These gates control the flow of information within the LSTM.

Figure 3: LSTM unit structure



Forget gate: The forget gate in an LSTM network determines whether information from the previous timestamp should be retained or discarded. This is expressed by Equation (7).

$$f_t = \sigma(X_t . U_f + H_{t-1} . W_f) \tag{7}$$

where X_t represents the input at the current timestamp *t*, U_f is the weight associated with the input, $H_{t-1}$is the hidden state from the previous timestamp *t-1*, and W_f is the weight matrix associated with the hidden state. A sigmoid function is applied to f_t and its output ranges between 0 and 1. A value of 0 indicates that the network forgets everything from the previous timestamp, while a value of 1 means that it retains all the information from *t-1*.

Input gate: The input gate in LSTM quantifies the importance of new information entering the unit. It is represented by Equation (8).

$$i_t = \sigma(X_t . U_i + H_{t-1} . W_i) \tag{8}$$

U_i and W_i are the weight matrices associated with the input and hidden states. Another sigmoid function is applied to i_t and its value will be between 0 and 1. A value close to 1 dedicates more importance to the input information. The new information to be added or subtracted from the cell state of the LSTM is shown by Equation (9). This time, a *tanh* activation function is used to make the input information between -1 (for subtraction of information from the cell state) and 1 (for adding information to the cell state). However, the new information, N_t, will be updated by input and forget gates before entering the cell state (Equation (10)).

$$N_t = tanh(X_t.U_c + H_{t-1}.W_c) \tag{9}$$

$$C_t = f_t.C_{t-1} + i_t.N_t \tag{10}$$

C_{t-1} is the cell state in the previous timestamp *t-1*.

Output gate: This gate decides what information will be outputted as a hidden state at time *t*. This is determined by Equations (11) and (12).

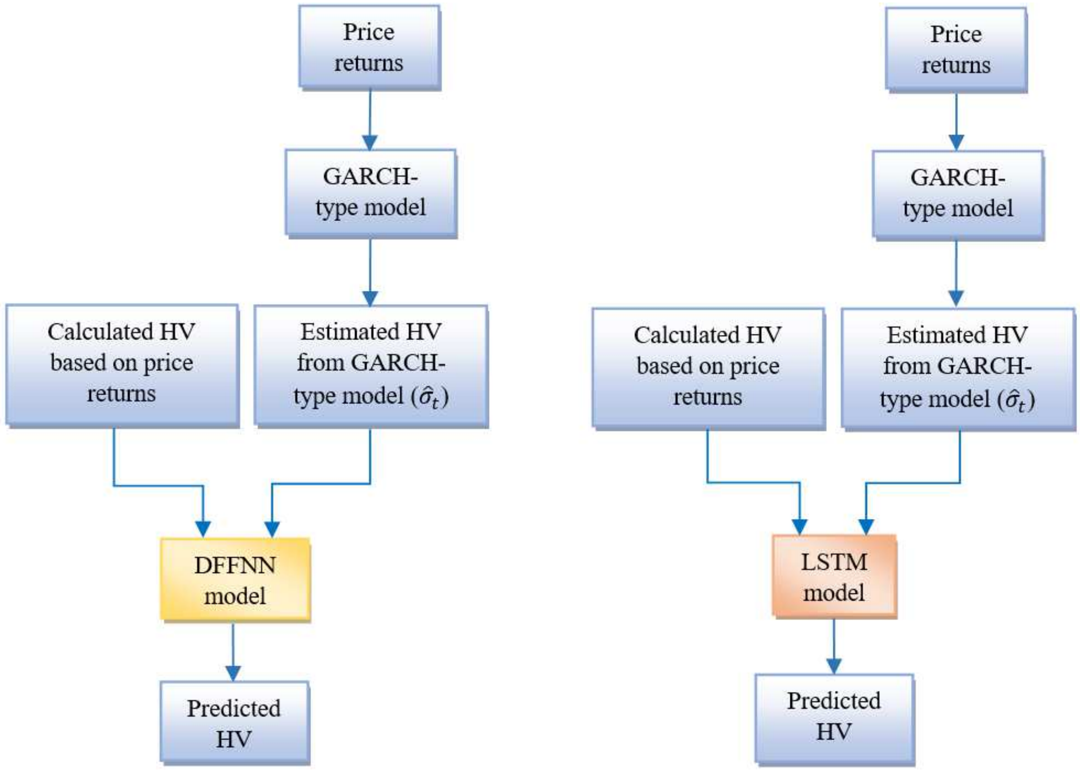
$$O_t = \sigma(X_t.U_o + H_{t-1}.W_o) \tag{11}$$

$$h_t = O_t.tanh\,(C_t) \tag{12}$$

3.3. Hybrid models

As mentioned in the introduction, the hybrid of Neural Networks and GARCH-type models is superior to single models of each. Therefore, our hybrid models are constructed by integrating the DFFNN and LSTM models with GARCH forecasts (See Figure 4). While the models combining GARCH, EGARCH, APGARCH, and DFFNN are referred to as G-DFFNN, E-DFFNN, and AP-DFFNN, respectively, the models combining GARCH, EGARCH, APGARCH, and LSTM are referred to as G-LSTM, E-LSTM, and AP-LSTM, respectively. The forecasts from each GARCH-type model, along with historical volatilities, are used as inputs for the DFFNN and LSTM models to improve the overall forecast.

Figure 4: Overall diagrams of the hybrid models used for volatility prediction



3.4. Performance metric

To evaluate each model's performance, we use the Root Mean Squared Error (RMSE) measure specified in Equation (13):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(HV_i - \widehat{HV}_i\right)^2} \tag{13}$$

where HV_i and $\widehat{HV}_i$ are the calculated (~ actual) and predicted historical volatilities, respectively; n is the number of predictions, which is equal to the size of testing samples, that is 290. The lower the RMSE, the better the accuracy of predictions.

4. Experiments and results

Three popular GARCH-type models — GARCH, EGARCH, and APGARCH — were selected to study their effectiveness in forecasting cryptocurrencies' volatility. Financial time series deviate from normality in real-life since they exhibit excessive skewness and kurtosis (Lahmiri &

Boukadoum, 2015). This is also evident in the descriptive statistics table (Table 2). Moreover, the results of Khaldi et al. (2019) revealed that EGARCH and APARCH models are highly sensitive to the type of distribution assumed for the residuals. Therefore, three different distributions were considered for the residuals of the GARCH-type models: Normal distribution to capture normality, Student's t distribution to capture skewness, and Generalized Error Distribution (GED) to capture the kurtosis of returns data (Lahmiri, 2017).

Regarding the ARCH and GARCH orders, represented by parameters p and q in the variance Equations (3), (4), and (5), we set $p = 1$ and $q = 1$, as many studies have shown that for financial time-series, the GARCH(1,1) is superior to other models with higher orders (Bollerslev, 1987; Hu et al., 2020). However, for the mean equation, i.e., $\hat{r}_t = \hat{\mu}_t + \epsilon_t$, various specifications were explored, including "zero mean", "constant mean", Autoregressive (AR) and Autoregressive Moving Average (ARMA) models such as "AR(1)", "AR(2)", and ARMA(1,1). For each cryptocurrency, the mean model was selected based on the lowest value for a penalized model selection criterion, i.e., the Akaike Info Criterion (AIC), and the highest value for the Log-likelihood function. It is worth mentioning that, in addition to these two criteria, the significance of parameters in the mean equation specification was also considered. Appendix A2 shows the results for Bitcoin data as an example.

After fitting the models to the whole dataset and estimating the unknown parameters like ω, α_i, β_i, and γ_i, conditional standard deviations (i.e., the $\hat{\sigma}_t$ in Equations (3), (4), and (5)) were computed for each cryptocurrency under three assumptions for residuals' distribution. This conditional standard deviation of returns data is referred to as estimated historical volatilities and will be used as inputs for the hybrid models, where we combine GARCH-type models with deep learning ones.

To assess the out-of-sample forecasting performance of GARCH-type models, each model was fitted to the training set. Using the best fitted model, the conditional standard deviations were forecasted for the testing sample. The nine sets of forecasted volatilities obtained from three GARCH-type models under three assumptions for residuals are presented in Appendix A3. The RMSE values were calculated by comparing the forecasted and calculated (~ actual) values of historical volatilities. As can be seen in Table 3 for each cryptocurrency, the lowest RMSE values are highlighted using three colors. While the green color shows the lowest RMSE value achieved for the corresponding cryptocurrency using a specific GARCH-type model under the best distribution for the residuals, the yellow and red colors indicate the lowest RMSE values achieved using the other two GARCH-type models with their respective distributions. For example, in the case of Binance coin, the lowest RMSE value is 0.01425 and it is achieved by APGARCH model when the Student's t distribution is assumed for the residuals. The best RMSE values achieved for the EGARCH and GARCH models are 0.01497 and 0.01654, respectively, assuming the Student's t distribution for the residuals.

A general overview of the results reported in Table 3 reveals that the green colors are mostly associated with the APGARCH model. This indicates the superiority of this model in forecasting the out-of-sample historical volatility compared to the EGARCH and GARCH models. The yellow color is primarily visible in the case of the EGARCH model, suggesting that this model performs better than the GARCH model, for which the best RMSE values are indicated by the red color. Regarding the choice of residuals' distribution, it is observed that the lowest RMSE values are mostly related to the Student's t distribution. This suggests that the Student's t distribution describes the residuals of most cryptocurrencies analyzed in this study very well.

We are interested in determining whether deep learning models significantly improve the forecasting errors of time series models. To answer this question, we will compare the RMSE values obtained for deep learning models with those of time series models shown in Table 3. The comparisons will be based on the average values of RMSE obtained for all 27 cryptocurrencies.

For deep learning models of this study, the objective is to train a model where h lagged values of HV_t are fed as an input vector for the model to predict HV_t. In order to assist the optimization algorithm of these models to converge faster, "min-max normalization" is used as described in Equation (14). Consequently, data is scaled into a range of 0 to 1.

$$\widetilde{HV}_t = \frac{HV_t - HV_{min}}{HV_{max} - HV_{min}} \tag{14}$$

where $\widetilde{HV}_t \in [0, 1]$, HV_{max}, and HV_{min} are the standardized, minimum, and maximum values of HV_t, respectively. We set the value of h to 6 so that given HV_{t-6}, …, HV_{t-1}, each model learns to predict the value of HV_t. Other values of h can be analyzed for robustness. Due to the highly volatile nature of cryptocurrencies, developing a model with the same sets of hyper-parameters that can perform well on multiple cryptocurrency datasets is not possible. To find the best DFFNN and LSTM models for each cryptocurrency, the most important hyper-parameters were selected to be tuned using the grid search method on a validation set. Consequently, our deep learning models' structures vary from one cryptocurrency to another, as both DFFNN and LSTM networks yield a specific set of optimal hyper-parameters for each cryptocurrency.

Table 3: RMSE of GARCH-type models obtained in HV forecasting for the testing set

	APGARCH			**EGARCH**			**GARCH**		
Cryptocurrency	**GED**	**normal**	**student's t**	**GED**	**normal**	**student's t**	**GED**	**normal**	**student's t**
Binance Coin	0.01472	0.01557	**0.01425**	0.01559	0.01683	**0.01497**	0.01698	0.01844	**0.01654**
Bitcoin	**0.00543**	0.00714	0.01165	**0.00543**	0.00732	0.01080	**0.00458**	0.00675	0.00968
Bitcoin Cash	**0.01821**	0.01910	0.02213	**0.01908**	0.01972	0.02188	**0.01876**	0.02051	0.02415
Cardano	0.01354	**0.01269**	0.01544	**0.01445**	0.02086	0.01596	0.01569	**0.01478**	0.01744
Chainlink	0.01295	0.01343	**0.01291**	0.01403	0.01426	**0.01400**	0.01348	**0.01301**	0.01384
Dai	0.00046	**0.00019**	0.00072	0.00044	**0.00019**	0.00060	0.00041	**0.00019**	0.00065
Dash	0.01908	**0.01879**	0.02060	0.02009	**0.01770**	0.02102	0.02278	**0.01915**	0.02423
Decentraland	0.03883	0.03952	**0.03741**	0.04150	0.04199	**0.04030**	0.04103	0.04351	**0.04064**
Enjin Coin	0.03047	**0.03038**	0.04500	0.02871	**0.02630**	0.04211	**0.02879**	0.03012	0.03854
EOS	**0.01749**	0.01869	0.03454	**0.02043**	0.02051	0.03302	0.01723	**0.01588**	0.04477
Ethereum	0.01115	**0.01105**	0.01300	0.01201	**0.01199**	0.01340	0.01268	**0.01211**	0.01371
Ethereum Classic	0.02809	**0.02788**	0.03036	0.02953	**0.02937**	0.03055	0.02965	**0.02770**	0.03307
Filecoin	0.02129	**0.01989**	0.02587	**0.02080**	0.02507	0.02523	**0.02442**	0.02533	0.03090
IOTA	0.01357	0.01305	**0.01275**	0.01492	0.01468	**0.01395**	0.01607	0.01515	**0.01514**
Litecoin	0.01503	0.01588	**0.01383**	0.01576	0.01651	**0.01465**	0.01407	0.01480	**0.01311**
Maker	0.02223	**0.02213**	0.02456	0.02243	**0.02207**	0.02457	**0.02340**	0.02371	0.02603
Monero	0.01894	**0.01805**	0.01810	0.01952	**0.01869**	0.01871	0.01755	0.01865	**0.01661**
Neo	0.01640	0.01994	**0.01552**	0.01744	0.02139	**0.01634**	0.01962	0.02470	**0.01659**
OMG Network	0.02295	0.02410	**0.02109**	0.02435	0.02564	**0.02242**	0.02584	0.02726	**0.02457**
Stellar	0.01659	0.01766	**0.01424**	0.01726	0.01830	**0.01593**	0.01847	0.01964	**0.01712**
Tether	0.00013	**0.00011**	0.00014	**0.00011**	0.00012	**0.00011**	0.00017	**0.00013**	0.00019
Tezos	0.02152	0.02199	**0.01987**	0.02278	0.02292	**0.02099**	0.02431	**0.02270**	0.02493
Theta	0.01616	0.01690	**0.01615**	0.01629	0.01750	**0.01627**	**0.01854**	0.01856	0.02028
TRON	0.01653	**0.01519**	0.01761	0.01729	**0.01639**	0.01792	0.01865	**0.01788**	0.02020
Waves	0.02057	0.02364	**0.01877**	0.02159	0.02460	**0.01963**	0.02193	0.02552	**0.02110**
XRP	**0.02126**	0.02155	0.02200	0.02126	0.02522	**0.02052**	**0.02285**	0.02835	0.02452
Zcash	0.01957	0.01928	**0.01868**	0.02139	0.02144	**0.02028**	0.02372	0.02430	**0.02232**
Average	**0.0175**	0.0179	0.0192	**0.0183**	0.0192	0.0195	**0.0190**	0.0196	0.0211

Table 4 shows the list of hyper-parameters and the a few possible values for them used in the optimization process. These hyper-parameters include the number of hidden layers, the number of neurons in each hidden layer, activation function, and dropout rate.

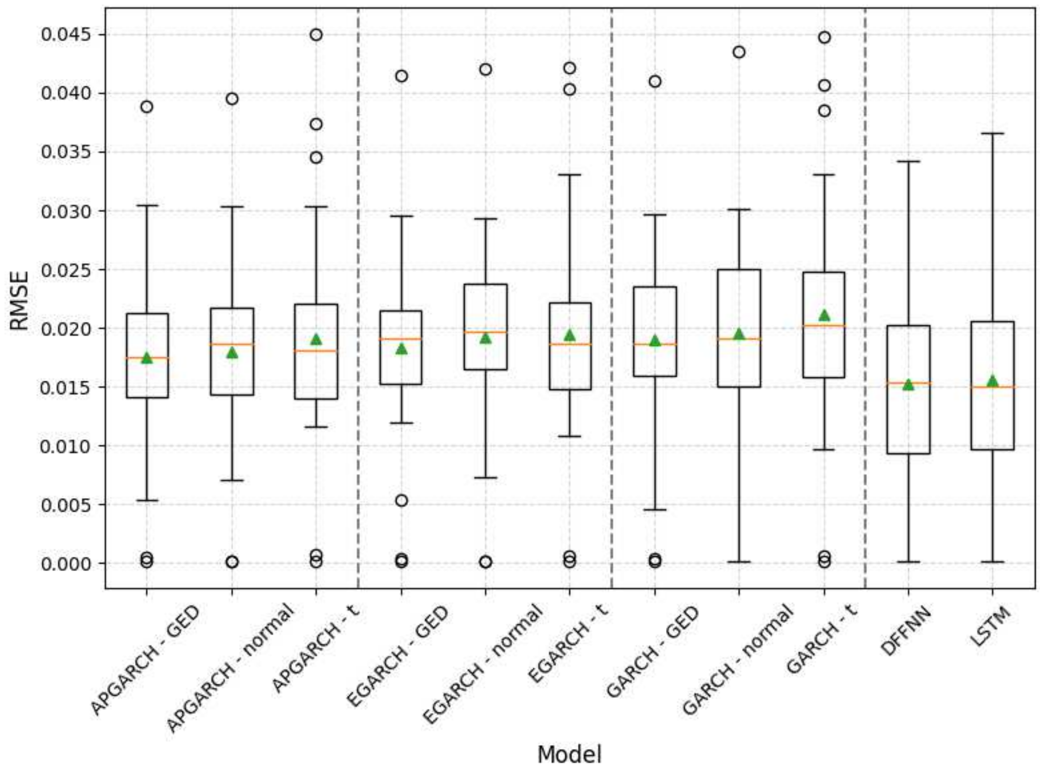
Table 4: Hyper-parameters of the DFFNN and LSTM models

Hyper-parameter	**Possible values**
Number of layers	[1, 2, 3, 4]
Number of units	[50, 100]
Activation functions	['relu', 'LeakyReLU', 'PReLU', ‘tanh’]
Dropout rates	[0.2, 0.1]

We used the Mean Squared Error (MSE) loss function, and the optimizer was set to “Adam” because other optimizers such as Stochastic Gradient Descent (SGD) did not show good results compared to “Adam”. Regarding the activation functions, the “sigmoid” was also tested, but its results were not satisfying. In addition, the use of dropout layers was found to be useful in reducing the overfitting problem. Therefore, two values of 0.1 and 0.2 were selected as possible dropout rates to be optimized in the hyper-parameter tuning process. It should be noted that our models were not overly complex, and we could achieve the best possible performances by training them in only 10 epochs. Therefore, we did not use other techniques to prevent the overfitting problem, such as early stopping, $L1$, or $L2$ regularization. Those techniques are mostly effective for more complex models that need to be simplified to avoid overfitting.

Using optimized parameters, each model was retrained on the entire training set (i.e., training + validation) to predict HV_t values for the testing set. Neural Networks produce different results when trained on the same data, making them unstable and unreliable due to factors such as random initialization of weights and biases, or randomness in regularization techniques like dropouts. To obtain more robust results, we reduced the effect of randomness by repeating the experiments 30 times. Consequently, we obtained 30 RMSE values for each cryptocurrency and each of the DFFNN and LSTM models. The average of these values was considered as a robust value for the RMSE. The results are shown in the first column of Table 6 and Table 7.The boxplots presented in Figure 5 provide a clearer comparison of the RMSE values of our deep learning and GARCH-type models. As seen in Figure 5, the DFFNN and LSTM models have the lowest RMSE values on average followed by the APGARCH model with GED distribution. On the other hand, the GARCH model with Students' t distribution has the highest average RMSE value, indicating its inefficiency for out-of-sample volatility forecasting. More detailed comparisons will be discussed in the later sections of this study.

Figure 5: Boxplots of the RMSE values: a comparison of deep learning and GARCH-type models



In the next step, we investigated whether including estimated HV_t from GARCH-type models as inputs for the deep learning models enhances the prediction performance of these models. In other words, we assessed the effectiveness of hybrid models, combining DFFNN with GARCH-type estimated volatilities and LSTM with GARCH-type estimated volatilities, in predicting HV_t. From the estimation outcomes of GARCH-type models, we obtained nine different estimated HV_t values, resulting from combinations of three GARCH-type models and three distributions. The correlation analysis results of all cryptocurrencies showed that all estimated HV_t values from GARCH-type models are correlated with the target variable (i.e., the HV_t calculated by Equation (2)). Figure 6 illustrates a heatmap of the correlation values for Bitcoin data as an example.

To comprehend the individual impact of each GARCH-type feature, we systematically examined them one by one (see Table 5). This approach aids in mitigating the issue of multicollinearity that may arise when using multiple GARCH-type HV_t values in a feature set.

Figure 6: Correlation matrix of historical volatilities for Bitcoin data as an example

Table 5: Historical volatilities as feature sets

Feature set	**Definition**
[HV]	Historical volatility calculated by using the Equation (2)
[HV, HV_apgarch_ged]	HV, Estimated HV from APGARCH model using GED distribution
[HV, HV_apgarch_normal]	HV, Estimated HV from APGARCH model using normal distribution
[HV, HV_apgarch_t]	HV, Estimated HV from APGARCH model using student's t distribution
[HV, HV_egarch_ged]	HV, Estimated HV from EGARCH model using GED distribution
[HV, HV_egarch_normal]	HV, Estimated HV from EGARCH model using normal distribution
[HV, HV_egarch_t]	HV, Estimated HV from EGARCH model using student's t distribution
[HV, HV_garch_ged]	HV, Estimated HV from APGARCH model using GED distribution
[HV, HV_garch_normal]	HV, Estimated HV from APGARCH model using normal distribution
[HV, HV_garch_t]	HV, Estimated HV from APGARCH model using student's t distribution

By conducting each experiment 30 times, we derived the average RMSE values for each feature set and each cryptocurrency, along with their standard deviations, as presented in Table 6 and Table 7. Moreover, Figure 7 and Figure 8 represent the results in the form of boxplots. In both tables, the feature set yielding the lowest RMSE value for each cryptocurrency is highlighted in green. It is evident from the colors that the hybrid models of AP-DFFNN and AP-LSTM consistently exhibit the lowest RMSE values. Overall, all hybrid models demonstrated smaller average RMSE values compared to their individual counterparts (DFFNN or LSTM). While these differences may not be immediately apparent in Figure 7 and Figure 8, the p-values of RMSE differences between the deep learning models and their hybrid versions were found to be statistically significant (see Table 12). In summary, the hybrid of our deep learning models with the APGARCH model produced the lowest average RMSE values, irrespective of the assumed distribution for the residuals. More detailed comparisons will be discussed in later sections of this study.

Table 6: RMSE values for DFFNN and its hybrid models in forecasting HV for the testing set

Cryptocurrency \ Feature	DFFNN HV	AP-DFFNN [HV, HV_apgarch_ged]	AP-DFFNN [HV, HV_apgarch_normal]	AP-DFFNN [HV, HV_apgarch_t]	E-DFFNN [HV, HV_egarch_ged]	E-DFFNN [HV, HV_egarch_normal]	E-DFFNN [HV, HV_egarch_t]	G-DFFNN [HV, HV_garch_ged]	G-DFFNN [HV, HV_garch_normal]	G-DFFNN [HV, HV_garch_t]
Binance Coin	**0.01208** (0.00115)	**0.01133** (0.00102)	**0.01085** (0.00131)	**0.0114** (0.00108)	**0.01183** (0.00131)	**0.01186** (0.00113)	**0.01156** (0.00099)	**0.01141** (0.00104)	**0.01177** (0.00091)	**0.01193** (0.00106)
Bitcoin	**0.00747** (0.00024)	**0.00738** (0.00030)	**0.00738** (0.00032)	**0.0074** (0.00033)	**0.00739** (0.00023)	**0.00737** (0.00028)	**0.00742** (0.00028)	**0.00743** (0.00034)	**0.00743** (0.00028)	**0.00744** (0.00021)
Bitcoin Cash	**0.01712** (0.00041)	**0.01685** (0.00043)	**0.01687** (0.00061)	**0.01705** (0.00051)	**0.01693** (0.00047)	**0.0171** (0.00046)	**0.01701** (0.00063)	**0.01720** (0.00061)	**0.01710** (0.00048)	**0.01718** (0.00044)
Cardano	**0.00986** (0.00034)	**0.00965** (0.00054)	**0.00971** (0.00054)	**0.0097** (0.00048)	**0.00978** (0.00049)	**0.00973** (0.00044)	**0.00975** (0.00043)	**0.00980** (0.00040)	**0.00982** (0.00035)	**0.00980** (0.00052)
Chainlink	**0.01472** (0.00039)	**0.01442** (0.00073)	**0.01451** (0.00065)	**0.01452** (0.00052)	**0.01464** (0.00040)	**0.01457** (0.00048)	**0.01457** (0.00046)	**0.01474** (0.00049)	**0.01476** (0.00052)	**0.01471** (0.00051)
Dai	**0.00033** (0.00001)	**0.00034** (0.00001)	**0.00034** (0.00001)	**0.00034** (0.00001)	**0.00034** (0.00001)	**0.00035** (0.00001)	**0.00034** (0.00001)	**0.00034** (0.00001)	**0.00035** (0.00001)	**0.00034** (0.00001)
Dash	**0.01839** (0.00075)	**0.01792** (0.00088)	**0.01808** (0.00124)	**0.01772** (0.00132)	**0.01865** (0.00082)	**0.01832** (0.00082)	**0.01831** (0.00081)	**0.01862** (0.00099)	**0.01856** (0.00071)	**0.01880** (0.00072)
Decentraland	**0.03581** (0.00221)	**0.03287** (0.00171)	**0.0333** (0.00177)	**0.03287** (0.00238)	**0.03326** (0.00206)	**0.03278** (0.00203)	**0.0326** (0.00197)	**0.03234** (0.00178)	**0.03281** (0.00167)	**0.03331** (0.00161)
Enjin Coin	**0.01577** (0.00039)	**0.0159** (0.00051)	**0.01578** (0.00050)	**0.016** (0.00048)	**0.0159** (0.00048)	**0.01581** (0.00051)	**0.01588** (0.00031)	**0.01593** (0.00039)	**0.01591** (0.00039)	**0.01594** (0.00041)
EOS	**0.02242** (0.00098)	**0.02151** (0.00068)	**0.02148** (0.00067)	**0.02158** (0.00082)	**0.02193** (0.00072)	**0.02177** (0.00089)	**0.02204** (0.00084)	**0.02148** (0.00078)	**0.02164** (0.00065)	**0.02180** (0.00059)
Ethereum	**0.01418** (0.00034)	**0.01387** (0.00037)	**0.01391** (0.00036)	**0.0139** (0.00040)	**0.01396** (0.00027)	**0.01398** (0.00029)	**0.01397** (0.00040)	**0.01407** (0.00038)	**0.01412** (0.00033)	**0.01409** (0.00031)
Ethereum Classic	**0.01297** (0.00078)	**0.01192** (0.00089)	**0.01193** (0.00093)	**0.01216** (0.00062)	**0.01213** (0.00078)	**0.01204** (0.00080)	**0.01225** (0.00066)	**0.01241** (0.00066)	**0.01207** (0.00070)	**0.01216** (0.00077)
Filecoin	**0.01007** (0.00179)	**0.00955** (0.00191)	**0.009** (0.00210)	**0.00935** (0.00203)	**0.0097** (0.00132)	**0.01001** (0.00134)	**0.01022** (0.00150)	**0.00988** (0.00178)	**0.00956** (0.00155)	**0.01059** (0.00198)
IOTA	**0.01173**	**0.01129**	**0.01132**	**0.01149**	**0.01146**	**0.01145**	**0.01148**	**0.01166**	**0.01146**	**0.01144**

	(0.00045)	(0.00068)	(0.00074)	(0.00046)	(0.00058)	(0.00066)	(0.00054)	(0.00055)	(0.00058)	(0.00063)
Litecoin	**0.01683**	**0.0164**	**0.01648**	**0.01647**	**0.01658**	**0.01657**	**0.01653**	**0.01665**	**0.01656**	**0.01670**
	(0.00030)	(0.00045)	(0.00054)	(0.00035)	(0.00049)	(0.00048)	(0.00049)	(0.00031)	(0.00039)	(0.00045)
Maker	**0.01035**	**0.01016**	**0.01008**	**0.01015**	**0.01021**	**0.00997**	**0.01016**	**0.01013**	**0.01001**	**0.01008**
	(0.00048)	(0.00048)	(0.00064)	(0.00051)	(0.00051)	(0.00059)	(0.00052)	(0.00044)	(0.00053)	(0.00050)
Monero	**0.01912**	**0.01912**	**0.01904**	**0.01908**	**0.01909**	**0.01912**	**0.01917**	**0.01912**	**0.01904**	**0.01903**
	(0.00048)	(0.00032)	(0.00048)	(0.00041)	(0.00039)	(0.00041)	(0.00036)	(0.00043)	(0.00044)	(0.00041)
Neo	**0.01526**	**0.01499**	**0.01502**	**0.01505**	**0.01504**	**0.01503**	**0.01505**	**0.01513**	**0.01516**	**0.01526**
	(0.00052)	(0.00046)	(0.00062)	(0.00051)	(0.00048)	(0.00043)	(0.00042)	(0.00039)	(0.00041)	(0.00048)
OMG Network	**0.0134**	**0.01334**	**0.01338**	**0.0133**	**0.01352**	**0.01349**	**0.01349**	**0.01337**	**0.01335**	**0.01339**
	(0.00045)	(0.00048)	(0.00061)	(0.00042)	(0.00059)	(0.00048)	(0.00049)	(0.00038)	(0.00040)	(0.00051)
Stellar	**0.02677**	**0.02756**	**0.02719**	**0.02662**	**0.02803**	**0.02872**	**0.02724**	**0.02743**	**0.02668**	**0.02682**
	(0.00106)	(0.00100)	(0.00095)	(0.00165)	(0.00117)	(0.00118)	(0.00115)	(0.00116)	(0.00118)	(0.00121)
Tether	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**
	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)
Tezos	**0.01918**	**0.01887**	**0.01875**	**0.01891**	**0.01899**	**0.01898**	**0.01897**	**0.01895**	**0.01898**	**0.01887**
	(0.00048)	(0.00036)	(0.00045)	(0.00050)	(0.00038)	(0.00046)	(0.00044)	(0.00048)	(0.00040)	(0.00039)
Theta	**0.01365**	**0.01346**	**0.01358**	**0.0134**	**0.01351**	**0.01358**	**0.01355**	**0.01366**	**0.01366**	**0.01362**
	(0.00078)	(0.00061)	(0.00065)	(0.00089)	(0.00076)	(0.00078)	(0.00059)	(0.00074)	(0.00070)	(0.00058)
TRON	**0.00879**	**0.00873**	**0.00864**	**0.00871**	**0.00874**	**0.00872**	**0.00875**	**0.00869**	**0.00868**	**0.00870**
	(0.00048)	(0.00056)	(0.00062)	(0.00049)	(0.00058)	(0.00063)	(0.00061)	(0.00071)	(0.00053)	(0.00050)
Waves	**0.01026**	**0.01026**	**0.01026**	**0.01033**	**0.01033**	**0.01026**	**0.01021**	**0.01035**	**0.01038**	**0.01042**
	(0.00081)	(0.00078)	(0.00070)	(0.00098)	(0.00080)	(0.00098)	(0.00068)	(0.00074)	(0.00063)	(0.00097)
XRP	**0.03199**	**0.03192**	**0.0317**	**0.03167**	**0.03167**	**0.03159**	**0.03029**	**0.03172**	**0.03110**	**0.03155**
	(0.00213)	(0.00212)	(0.00239)	(0.00223)	(0.00243)	(0.00195)	(0.00268)	(0.00246)	(0.00284)	(0.00248)
Zcash	**0.01371**	**0.01367**	**0.0137**	**0.01364**	**0.01375**	**0.01373**	**0.01371**	**0.01374**	**0.01371**	**0.01368**
	(0.00042)	(0.00036)	(0.00041)	(0.00040)	(0.00048)	(0.00037)	(0.00037)	(0.00038)	(0.00042)	(0.00050)
Average	0.0149	0.0146	0.0145	0.0146	0.0147	0.0147	0.0146	0.0147	0.0146	0.0147

Table 7: RMSE values for LSTM and its hybrid models in forecasting HV for the testing set

	LSTM	AP-LSTM			E-LSTM			G-LSTM		
Feature / **Cryptocurrency**	HV	[HV, HV_apgarch_ged]	[HV, HV_apgarch_normal]	[HV, HV_apgarch_t]	[HV, HV_egarch_ged]	[HV, HV_egarch_normal]	[HV, HV_egarch_t]	[HV, HV_garch_ged]	[HV, HV_garch_normal]	[HV, HV_garch_t]
Binance Coin	**0.00681** (0.00065)	**0.00714** (0.00055)	**0.00730** (0.00042)	**0.00689** (0.00048)	**0.00680** (0.00043)	**0.00687** (0.00057)	**0.00683** (0.00061)	**0.00696** (0.00037)	**0.00704** (0.00049)	**0.00699** (0.00056)
Bitcoin	**0.01164** (0.00013)	**0.01119** (0.00017)	**0.01115** (0.00017)	**0.01132** (0.00021)	**0.01132** (0.00018)	**0.01121** (0.00016)	**0.01138** (0.00017)	**0.01151** (0.00014)	**0.01147** (0.00019)	**0.01151** (0.00016)
Bitcoin Cash	**0.01958** (0.00033)	**0.01915** (0.00031)	**0.01917** (0.00044)	**0.01919** (0.00040)	**0.01942** (0.00030)	**0.01931** (0.00031)	**0.01941** (0.00049)	**0.01962** (0.00040)	**0.01956** (0.00035)	**0.01968** (0.00037)
Cardano	**0.01106** (0.00021)	**0.00999** (0.00028)	**0.00992** (0.00027)	**0.01009** (0.00033)	**0.01030** (0.00026)	**0.01032** (0.00027)	**0.01051** (0.00027)	**0.01063** (0.00025)	**0.01060** (0.00031)	**0.01066** (0.00031)
Chainlink	**0.01734** (0.00018)	**0.01570** (0.00018)	**0.01543** (0.00014)	**0.01561** (0.00020)	**0.01598** (0.00021)	**0.01596** (0.00016)	**0.01605** (0.00015)	**0.01685** (0.00018)	**0.01691** (0.00020)	**0.01694** (0.00020)
Dai	**0.00037** (0.00001)	**0.00038** (0.00001)	**0.00038** (0.00001)	**0.00037** (0.00001)	**0.00038** (0.00001)	**0.00039** (0.00001)	**0.00038** (0.00001)	**0.00037** (0.00001)	**0.00039** (0.00001)	**0.00038** (0.00001)
Dash	**0.00904** (0.00061)	**0.01022** (0.00074)	**0.01124** (0.00077)	**0.00987** (0.00084)	**0.00929** (0.00078)	**0.01053** (0.00067)	**0.00891** (0.00054)	**0.00879** (0.00058)	**0.00994** (0.00056)	**0.00845** (0.00061)
Decentraland	**0.02142** (0.00053)	**0.02020** (0.00042)	**0.02028** (0.00042)	**0.02033** (0.00044)	**0.02090** (0.00046)	**0.02106** (0.00041)	**0.02085** (0.00047)	**0.02045** (0.00056)	**0.02039** (0.00037)	**0.02050** (0.00061)
Enjin Coin	**0.03654** (0.00098)	**0.03623** (0.00061)	**0.03584** (0.00050)	**0.03627** (0.00080)	**0.03598** (0.00073)	**0.03593** (0.00054)	**0.03602** (0.00070)	**0.03616** (0.00082)	**0.03624** (0.00063)	**0.03626** (0.00061)
EOS	**0.02883** (0.00031)	**0.02574** (0.00036)	**0.02516** (0.00040)	**0.02600** (0.00040)	**0.02593** (0.00040)	**0.02524** (0.00048)	**0.02645** (0.00044)	**0.02814** (0.00046)	**0.02803** (0.00042)	**0.02788** (0.00037)
Ethereum	**0.01640** (0.00018)	**0.01539** (0.00027)	**0.01565** (0.00017)	**0.01543** (0.00021)	**0.01578** (0.00021)	**0.01596** (0.00022)	**0.01580** (0.00017)	**0.01623** (0.00019)	**0.01621** (0.00016)	**0.01616** (0.00025)
Ethereum Classic	**0.02241** (0.00029)	**0.01986** (0.00038)	**0.01938** (0.00062)	**0.02011** (0.00035)	**0.02036** (0.00033)	**0.01942** (0.00059)	**0.02065** (0.00029)	**0.02122** (0.00030)	**0.02134** (0.00030)	**0.02107** (0.00030)
Filecoin	**0.00916** (0.00093)	**0.00928** (0.00097)	**0.00902** (0.00075)	**0.00942** (0.00094)	**0.00962** (0.00087)	**0.00966** (0.00080)	**0.00977** (0.00109)	**0.00906** (0.00099)	**0.00914** (0.00114)	**0.00955** (0.00105)
IOTA	**0.01544**	**0.01339**	**0.01324**	**0.01368**	**0.01375**	**0.01386**	**0.01386**	**0.01451**	**0.01454**	**0.01463**

	(0.00037)	(0.00035)	(0.00042)	(0.00047)	(0.00041)	(0.00040)	(0.00042)	(0.00035)	(0.00035)	(0.00037)
Litecoin	**0.01772**	**0.01631**	**0.01653**	**0.01640**	**0.01659**	**0.01680**	**0.01663**	**0.01708**	**0.01711**	**0.01712**
	(0.00025)	(0.00020)	(0.00023)	(0.00030)	(0.00031)	(0.00024)	(0.00022)	(0.00025)	(0.00028)	(0.00024)
Maker	**0.02072**	**0.02050**	**0.02034**	**0.02053**	**0.02076**	**0.02089**	**0.02085**	**0.02068**	**0.02047**	**0.02082**
	(0.00032)	(0.00036)	(0.00025)	(0.00033)	(0.00040)	(0.00032)	(0.00033)	(0.00036)	(0.00029)	(0.00026)
Monero	**0.02050**	**0.01938**	**0.01933**	**0.01950**	**0.01961**	**0.01955**	**0.01970**	**0.01999**	**0.02008**	**0.02005**
	(0.00017)	(0.00030)	(0.00028)	(0.00028)	(0.00027)	(0.00024)	(0.00027)	(0.00023)	(0.00021)	(0.00024)
Neo	**0.02480**	**0.02385**	**0.02379**	**0.02398**	**0.02391**	**0.02371**	**0.02398**	**0.02418**	**0.02410**	**0.02412**
	(0.00027)	(0.00026)	(0.00028)	(0.00035)	(0.00036)	(0.00034)	(0.00032)	(0.00022)	(0.00027)	(0.00032)
OMG Network	**0.01219**	**0.01271**	**0.01267**	**0.01272**	**0.01245**	**0.01245**	**0.01260**	**0.01252**	**0.01269**	**0.01248**
	(0.00051)	(0.00064)	(0.00055)	(0.00059)	(0.00057)	(0.00064)	(0.00069)	(0.00050)	(0.00066)	(0.00080)
Stellar	**0.00596**	**0.00634**	**0.00624**	**0.00572**	**0.00638**	**0.00646**	**0.00606**	**0.00647**	**0.00664**	**0.00639**
	(0.00027)	(0.00035)	(0.00030)	(0.00038)	(0.00030)	(0.00046)	(0.00030)	(0.00035)	(0.00039)	(0.00036)
Tether	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**	**0.00013**
	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)	(0.00000)
Tezos	**0.03170**	**0.03108**	**0.03112**	**0.03107**	**0.03164**	**0.03166**	**0.03156**	**0.03157**	**0.03137**	**0.03160**
	(0.00025)	(0.00042)	(0.00041)	(0.00045)	(0.00035)	(0.00035)	(0.00038)	(0.00031)	(0.00026)	(0.00038)
Theta	**0.01393**	**0.01351**	**0.01343**	**0.01349**	**0.01364**	**0.01348**	**0.01369**	**0.01356**	**0.01353**	**0.01362**
	(0.00020)	(0.00016)	(0.00020)	(0.00023)	(0.00021)	(0.00015)	(0.00023)	(0.00021)	(0.00018)	(0.00018)
TRON	**0.00956**	**0.00907**	**0.00880**	**0.00903**	**0.00932**	**0.00919**	**0.00942**	**0.00923**	**0.00932**	**0.00942**
	(0.00036)	(0.00029)	(0.00041)	(0.00040)	(0.00040)	(0.00035)	(0.00028)	(0.00035)	(0.00036)	(0.00035)
Waves	**0.01194**	**0.01163**	**0.01163**	**0.01168**	**0.01156**	**0.01141**	**0.01168**	**0.01144**	**0.01144**	**0.01152**
	(0.00036)	(0.00022)	(0.00029)	(0.00028)	(0.00028)	(0.00023)	(0.00029)	(0.00020)	(0.00027)	(0.00027)
XRP	**0.00991**	**0.01333**	**0.01340**	**0.01264**	**0.01263**	**0.01306**	**0.01273**	**0.01183**	**0.01282**	**0.01197**
	(0.00096)	(0.00116)	(0.00082)	(0.00092)	(0.00097)	(0.00095)	(0.00098)	(0.00113)	(0.00111)	(0.00095)
Zcash	**0.01498**	**0.01414**	**0.01403**	**0.01421**	**0.01427**	**0.01417**	**0.01430**	**0.01450**	**0.01442**	**0.01449**
	(0.00015)	(0.00012)	(0.00016)	(0.00011)	(0.00012)	(0.00014)	(0.00012)	(0.00011)	(0.00012)	(0.00012)
Average	0.0156	0.0150	0.0150	0.0150	0.0151	0.0151	0.0152	0.0153	0.0154	0.0153

Figure 7: Boxplots of the RMSE values: a comparison of DFFNN model with its hybrid models

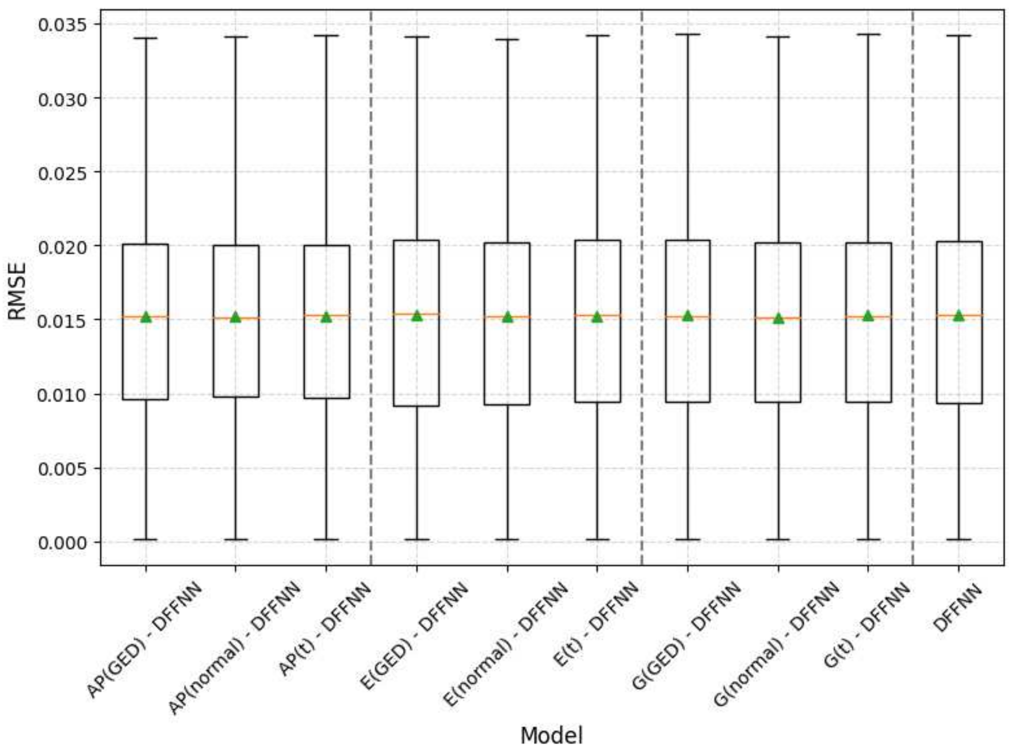
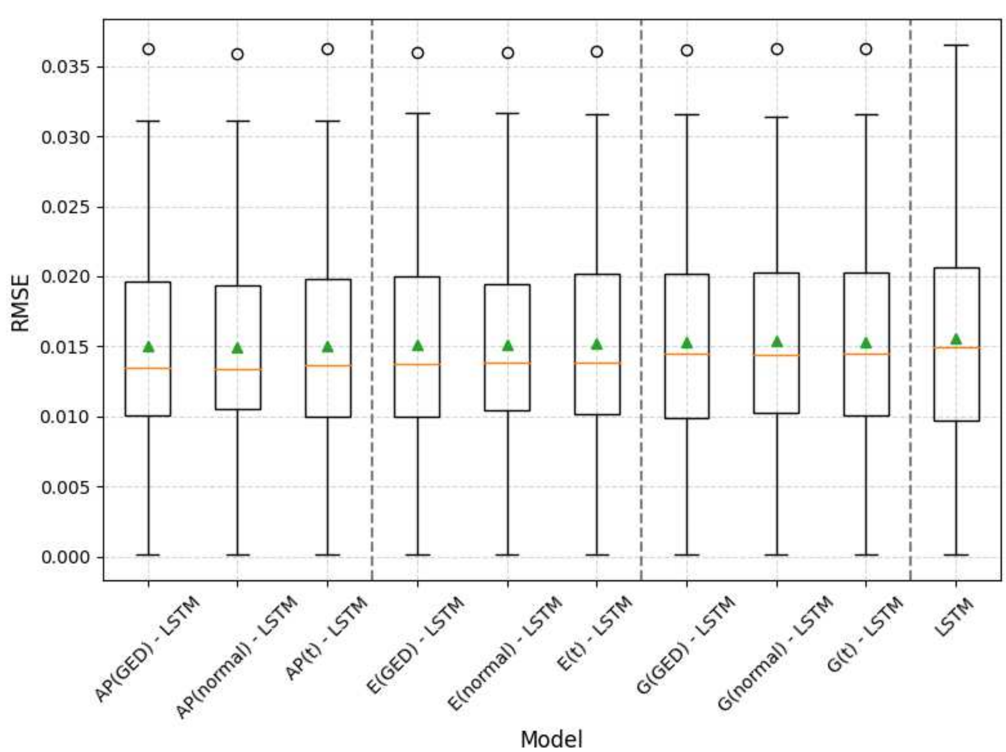


Figure 8: Boxplots of the RMSE values: a comparison of LSTM model with its hybrid models



5. Discussion

In this study, we investigated (a) which GARCH-type model is the best for forecasting cryptocurrencies' volatility in out-of-sample data, (b) the performance of deep learning models namely the common DFFNN and LSTM models in improving the forecasts of GARCH-type models, and (c) whether adding GARCH-type estimations to the DFFNN and LSTM models improve forecasting accuracies; in addition, (d) we used a large dataset composed of 27 cryptocurrencies to draw general and robust conclusions, and (e) we employed formal statistical tests to check differences in model performances.

In this regard, to perform the aforementioned investigations and identify statistically significant differences in out of-sample forecast accuracy, we utilized a paired sample t-test where each row in the two samples corresponds to the same cryptocurrency, with two RMSE values for comparison. In addition, to increase the power of the test, one-tailed version of the test was employed. It is crucial to clarify that the term "RMSE" denotes the average RMSE value calculated across all 27 cryptocurrencies.

The null hypotheses and discussion on each research question are as follows:

$$H_o: \overline{RMSE}_{GARCH-type(i)} - \overline{RMSE}_{GARCH-type(j)} \geq 0 \tag{15}$$
$$H_a: \overline{RMSE}_{GARCH-type(i)} - \overline{RMSE}_{GARCH-type(j)} < 0$$

According to the results reported in Table 8, the APGARCH model with the GED distribution for residuals is the best GARCH-type model in forecasting the historical volatility of cryptocurrencies. This conclusion is supported by the significant reduction in RMSE values compared to all other model-distribution combinations. When comparing the RMSE of this model with others, the null hypothesis (15) is rejected at a 5% level of significance. The APGARCH-GED model demonstrated the lowest and highest RMSE reduction in the case of APGARCH-normal and GARCH-t, improving RMSE values by 2.19% and 17.12%, respectively. The second-best model is APGARCH-normal with RMSE reduction values between 2.17% and 15.26%. However, the APGARCH model with a Student's t distribution did not outperform EGARCH and GARCH models with GED and normal distributions. It is worth mentioning that this conclusion does not contradict the observation from Table 3 where we stated that the Student's t distribution is the best one to be assumed for residuals. In fact, Table 3 reflects cryptocurrency-level results, allowing for the selection of the best distribution for each cryptocurrency whereas Table 8 represents the average results from a t-test perspective. The first approach indicates the effectiveness where we select the best distribution for each cryptocurrency, while the second approach shows the efficiency where we find the most efficient distribution that results in lowest RMSE values for all cryptocurrencies on average.

Analyzing the RMSE reduction values by the EGARCH-GED model reveals that it significantly improved the prediction performance compared to all other EGARCH and GARCH models. Consequently, in response to the first research question, we assert that the top three GARCH-type models for forecasting cryptocurrencies' out-of-sample volatility are APGARCH-GED, APGARCH-normal, and EGARCH-GED. To draw more robust conclusions, we compared three GARCH-type models, considering their best RMSE values (shown by green, yellow, and red colors in each row of Table 3). According to the results reported in Table 9, the APGARCH model was found to be the best model followed by the EGARCH model. This indicates that the leverage effect was captured efficiently by these two models and the GARCH model is not an efficient model for forecasting cryptocurrencies' volatility.

$$H_o: \overline{RMSE}_{Deep\ Learning(i)} - \overline{RMSE}_{GARCH-type(j)} \geq 0 \tag{16}$$
$$H_a: \overline{RMSE}_{Deep\ Learning(i)} - \overline{RMSE}_{GARCH-type(j)} < 0$$

To address the second research question, we compared the average RMSE values obtained by our deep learning models with the average RMSE value of each GARCH-type model. The RMSE reduction values and their significance are presented in Table 10. In almost all cases, the null hypothesis of (16) is rejected at the 5% or even 1% level of significance. The lowest and highest RMSE reductions by the DFFNN model are observed in cases of APGARCH-GED and GARCH-t, with values of 14.96% and 29,52%, respectively. Similar cases are observed for the LSTM model, where this model reduced the RMSE value of the APGARCH-GED and GARCH-t by 11.22% and 26.42%, respectively. This indicates that both DFFNN and LSTM models are superior to any GARCH-type model in forecasting cryptocurrencies' volatility. It is worth mentioning that we applied the paired t-test to compare the results achieved by the LSTM and DFFNN models, and the RMSE differences were not statistically significant, meaning that these two models were equal in terms of average prediction performance.

To make more robust conclusions, the comparisons were made considering the best RMSE values of GARCH-type models (shown by green, yellow, and red colors in each row of Table 3). As seen in Table 11, the DFFNN model exhibited significantly improved prediction performance even when the best RMSE values were considered for the GARCH-type models. However, in case of the LTSM model, the RMSE reduction value from this model to the APGARCH model with its best results was not statistically significant.

$$H_o: \overline{RMSE}_{hybrid(i)} - \overline{RMSE}_{DL(j)} \geq 0 \tag{17}$$
$$H_a: \overline{RMSE}_{hybrid(i)} - \overline{RMSE}_{DL(j)} < 0$$

The answer to the third research question can be found in the results shown in Table 12 where the null hypothesis (17) is rejected in all paired comparisons. Adding APGARCH model's estimated volatilities to the inputs of the DFFNN and LSTM models can reduce the RMSE value by more than 2% and 3%, respectively. Adding EGARCH and GARCH forecasts can also improve the RMSE values of our deep learning models by 1% to 3 %. Therefore, the hybrid of GARCH-type and DFFNN and LSTM models can improve the performance in forecasting of cryptocurrencies' volatility. This conclusion is consistent with previous studies in this domain which proved the effectiveness of hybrid models in volatility forecasting.

Table 8: RMSE value reduction (%) from GARCH-type(i) model to GARCH-type(j) model (i: index of rows, j: index of columns)

		APGARCH			EGARCH			GARCH		
		GED	**normal**	**student's t**	**GED**	**normal**	**student's t**	**GED**	**normal**	**student's t**
APGARCH	**GED**	-	-2.19**	-8.51**	-4.31***	-8.58***	-10.06***	-7.53***	-10.52***	-17.12***
	normal		-	-6.46*	-2.17*	-6.53***	-8.05**	-5.46***	-8.52***	-15.26***
	student's t			-	4.60	-0.07	-1.69*	1.08	-2.20	-9.40***
EGARCH	**GED**				-	-4.46**	-6.01*	-3.37**	-2.20***	-13.38***
	normal					-	-1.62	1.15	-6.49	-9.34**
	student's t						-	2.82	-2.13	-7.84***
GARCH	**GED**							-	-0.51*	-10.37**
	normal								-	-7.37
	student's t									-

***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

Table 9: RMSE value reduction (%) from GARCH-type(i) model to GARCH-type(j) model considering the lowest RMSEs achieved for each model (i: index of rows, j: index of columns)

	APGARCH	**EGARCH**	**GARCH**
APGARCH	-	-4.02***	-7.04***
EGARCH	-	-	-3.15**
GARCH	-	-	-

***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

Table 10: RMSE value reduction (%) from deep learning (i) to GARCH-type(j) model (i: index of rows, j: index of columns)

	APGARCH			EGARCH			GARCH		
	GED	**normal**	**student's t**	**GED**	**normal**	**student's t**	**GED**	**normal**	**student's t**
DFFNN	-14.96**	-16.83**	-22.20***	-18.63***	-22.26***	-23.52***	-21.37***	-23.91***	-29.52***
LSTM	-11.22*	-13.16*	-18.78***	-15.04**	-18.84**	-20.15***	-17.90**	-20.56**	-26.42***

***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

Table 11: RMSE value reduction (%) from deep learning (i) to GARCH-type(j) model considering the lowest RMSEs achieved for each model (i: index of rows, j: index of columns)

	APGARCH	**EGARCH**	**GARCH**
DFFNN	-11.42*	-14.98**	-17.65**
LSTM	-7.51	-11.23*	-14.03*

***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

Table 12: RMSE value reduction (%) from hybrid model (i) to deep learning model (j) (i: index of rows, j: index of columns)

	Feature set	**DFFNN**	**LSTM**
AP-DFFNN	[HV, HV_apgarch_ged]	-2.22***	-
	[HV, HV_apgarch_normal]	-2.47***	-
	[HV, HV_apgarch_t]	-2.34***	-
E-DFFNN	[HV, HV_egarch_ged]	-1.21*	-
	[HV, HV_egarch_normal]	-1.32*	-
	[HV, HV_egarch_t]	-1.92**	-
G-DFFNN	[HV, HV_garch_ged]	-1.49*	-
	[HV, HV_garch_normal]	-1.88**	-
	[HV, HV_garch_t]	-1.14*	-
AP-LSTM	[HV, HV_apgarch_ged]	-	-3.39**
	[HV, HV_apgarch_normal]	-	-3.69**
	[HV, HV_apgarch_t]	-	-3.43***
E-LSTM	[HV, HV_egarch_ged]	-	-2.71**
	[HV, HV_egarch_normal]	-	-2.71**
	[HV, HV_egarch_t]	-	-2.35**
G-LSTM	[HV, HV_garch_ged]	-	-1.52**
	[HV, HV_garch_normal]	-	-0.99
	[HV, HV_garch_t]	-	-1.35**

***, **, and * indicate a rejection of null hypothesis at the 1%, 5%, and 10% level of significance

6. Conclusion

This study proposed an extensive comparison of two categories of models in the context of volatility forecasting for cryptocurrency market data: (1) GARCH-type as time series models, (2) DFFNN and LSTM models as two of the most popular deep learning models. Our experimental results on 27 cryptocurrencies' data showed that our deep learning models outperform the GARCH-type models regardless of any distribution assumption for the residuals. In addition, we studied various hybrid models where the outputs of GARCH-type models are used as inputs for DFFNN and LSTM models to improve the prediction performance of these models. The conclusion was that the hybrid models perform better than single models at volatility forecasting. Although many studies have explored volatility forecasting using hybrid models, the application of these models to cryptocurrencies' volatility is rare. Even in studies related to the cryptocurrency domain, most of the researchers used Bitcoin's data to prove the effectiveness of hybrid models. Our study contributes to the existing literature by conducting experiments on 27 cryptocurrencies with available data from February 2018 to February 2022. Our second contribution is that we evaluated the predictive performance of hybrid models using hypothesis tests and not just comparing error metrics. Both contributions, i.e., the use of large set of cryptocurrencies and hypothesis tests allowed us to evaluate the strength of our claims and prove that the superiority of the neural networks over the traditional time series models has not occurred by chance.

In summary, our findings align with the results of other studies in the literature, concluding that Neural Networks, and in particular, their combination with GARCH-type forecasts, improve the volatility forecasts of GARCH-type models in cryptocurrency markets.

Essay 2: Investigating the Effectiveness of Twitter Sentiment in Cryptocurrency Close Price Prediction by Using Deep Learning

Abstract

In recent years, the prediction of cryptocurrency prices has attracted the interest of many, including investors, researchers, and practitioners. In this study, we propose a hybrid model for predicting the daily close prices of cryptocurrencies based on various neural networks such as Long Short-Term Memory (LSTM), Convolutional Neural Network (CNN), and Attention mechanism. Using an ensemble of three pre-trained language models, we extract sentiment from cryptocurrency-related tweets posted between January 1, 2021, and December 31, 2021. We construct 20 different versions of our model and evaluate their performance on data from the 27 most traded cryptocurrencies, using a history of previous days' sentiment data along with close prices as input. The flexible input layer of our model enables different ways of feeding data into the model to adjust it for different cryptocurrencies, yielding better predictions. Our analysis reveals several important findings. We show that longer sequences of input data achieve more accurate predictions on average. Specifically, using a history of 14- and 21-days' data results in the lowest RMSE values on average compared to using a history of 7 days. However, there is no significant difference between the results related to the input sequences with lengths of 14 and 21. In addition, our findings suggest that sentiment data can be useful in predicting prices for more than 70% of the studied cryptocurrencies. Thus, people's emotions, opinions, and sentiment expressed through their posts on Twitter platform play a significant role in predicting cryptocurrency prices.

Keywords: Cryptocurrencies, Price prediction, Deep learning, Sentiment analysis, Hybrid model

1. Introduction

Since the introduction of blockchain by Nakamoto (2008) as an underlying technology for the first cryptocurrency, i.e., Bitcoin, the application of this technology has been raised in many sectors, particularly, in the financial sector. According to Chang et al., 2020, blockchain has significant impacts on financial services by introducing decentralization, transparency, security, and innovation. "Finance is the natural scenario of the blockchain, and cryptocurrencies are also by far one of its most successful applications", Chang et al., 2020. Cryptocurrencies operate on decentralized networks of blockchain, which eliminates the central control of banks over financial transactions. The transparency of blockchain increases trust among participants in the cryptocurrency ecosystem, consequently positioning cryptocurrencies as favored investment targets for investors all over the world. Cryptocurrencies offer diversification and hedging benefits for investors by considerably reducing portfolio risk (Sun et al., 2020a). Due to cryptocurrencies' decentralization, immutability, and security, they have become a global phenomenon, attracting a significant number of users (Oyedele et al., 2023b). Accurate price forecasts are important for asset allocation and gaining profits in cryptocurrency markets. In this regard, a wide variety of techniques have been attempted and utilized by researchers of this domain. However, developing a reliable and accurate model that can perform well in forecasting all cryptocurrencies' prices is a challenging task due to the different characteristics of each market. That is why researchers evaluate the effectiveness of their proposed models on a limited number of cryptocurrencies. In this study, we aim to address such difficulty by proposing a prediction model with a flexible architecture that can be adaptable to different cryptocurrencies.

Our study has been significantly inspired by two streams of studies in the literature: 1) studies that utilize neural networks especially the Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) (Patel et al., 2020; Chen et al., 2021; Zoumpekas et al., 2020; Ji et al., 2019; Zhang et al., 2021; Lahmiri & Bekiros, 2020; Oyedele et al., 2023), 2) studies that investigate the effectiveness of social media data in predicting cryptocurrency prices (Kraaijeveld & de Smedt, 2020; Ortu et al., 2022; Anbaee Farimani et al., 2022; Zou & Herremans, 2022).

The novelty of our work can be seen from both experimental and methodological aspects:

- From the experimental point of view:
  i. Most previous studies have primarily focused on Bitcoin due to its largest market capitalization (e.g., Chen et al., 2021; Ji et al., 2019; Lahmiri & Bekiros, 2020; Zou & Herremans, 2022). In contrast, we assess the performance of our price prediction models on a comprehensive set consisting of 27 cryptocurrencies. Although other cryptocurrencies have not attracted much attention, it is worth investigating the robustness of the models on less famous cryptocurrencies to offer a suitable strategy for their price prediction and understand their overall price dynamics (Oyedele et al., 2023b).

ii. We perform sentiment analysis on an extensive text dataset comprising over 16 million tweets related to cryptocurrencies posted between 2021-01-01 and 2021-12-31. We show the effectiveness of using sentiment data in predicting the close price of more than 70% of the cryptocurrencies analyzed in this study. The reason that we use Twitter data is that this social media platform is the most popular one among traders and provides a combination of both news and investor sentiment at the same time (Kraaijeveld & de Smedt, 2020).

- From the methodological point of view:

i. We propose a hybrid prediction model based on two popular deep learning models, namely LSTM and CNN with a flexible input layer that can be customized for different cryptocurrencies. Due to the highly volatile nature of cryptocurrencies, developing a single model that performs well across multiple cryptocurrency datasets is not feasible. The flexibility of our model enables the selection of the most suitable architecture for each cryptocurrency. It is worth mentioning that the rationale behind employing both CNN and LSTM models is that the former performs well in identifying local dependencies that are independent from time, while the latter can capture long-term dependencies in the data.

ii. The novelty of our sentiment analysis compared to other similar studies such as Kraaijeveld & de Smedt (2020), lies in the utilization of a different approach. While they employ a lexicon-based method to calculate polarity scores for each tweet, we employ three pre-trained language models in an ensemble way to calculate sentiment scores for the tweets. Lexicon-based sentiment analysis relies on limited pre-defined sentiment dictionaries, whereas pre-trained language models have been trained on vast amounts of text data and can capture the meaning of words within their surrounding context. Moreover, using pre-trained language models that have been fine-tuned specifically for the finance and social media data, will increase the accuracy of predicted sentiments. Furthermore, compared to the work of Zou & Herremans (2022) in which only one pre-trained language model was employed for sentiment analysis, we use three of them in an ensemble way, reducing the risk associated with relying on the predictions of a single model.

To the best of our knowledge, the combination of ensemble sentiment classifiers and a hybrid price prediction model with a flexible input layer has not been proposed by other researchers, making our framework novel and innovative.

2. Literature review

2.1. Neural networks and cryptocurrency price prediction

The high performance of both LSTM and Gated Recurrent Unit (GRU) models in time series forecasting has been demonstrated in numerous studies, as they can handle the problem of gradient vanishing (Bengio et al., 1994) in long sequences. Patel et al. (2020) developed a hybrid model that concatenates the results of GRU and LSTM networks to predict the prices of Litecoin and Monero using a dense layer. For training data preparation, the authors created multiple input-output samples where the inputs include the average price and its 30 previous values, and the output is the next day's average price. After training the model, the last 30 observations in the training set were used to predict the next day's price (i.e., the first price in the testing set). The predicted price was then incorporated into the next input sequence along with its last 30 values to predict the subsequent value. This process continues for k times (where $k = 1$, 3, and 7 days), resulting in k forecasts that can be compared with their real values. The results showed that the proposed hybrid model outperformed an LSTM baseline model for all k values. However, the authors found that their hybrid model was particularly effective in short-term predictions, i.e., for $k = 1$.

Chen et al. (2021) collected a diverse range of technical and economic features across four distinct periods for Bitcoin. Using two feature selection approaches, they identified the most significant factors affecting Bitcoin's price during each period. The selected features were used as inputs for an LSTM network. To find out whether these factors play the major role in price forecasting, the results were compared to those of four baseline models trained solely on historical price data. The baseline models comprised the Autoregressive Integrated Moving Average (ARIMA), Support Vector Regression (SVR), Adaptive Network Fuzzy Inference System (ANFIS), and another LSTM, all trained exclusively with historical price data. Their findings indicated that not only did the economic and technological features used in the LSTM model yield higher prediction performances, but their effects also varied over time. In other words, the factors determining the price of Bitcoin in different periods were not the same.

Zoumpekas et al. (2020) conducted a comparative analysis of six machine learning models, assessing both prediction performance and complexity in forecasting the closing price of Ethereum at half-hour intervals. The CNN models in this study included a two-layer and three-layer networks, both performing one-dimensional convolutions, as the authors were dealing with one-dimensional time series data. In addition to regular LSTM and GRU networks, the stacked LSTM and Bi-Directional LSTM (BiLSTM) were also investigated in this work. After validating all six networks, the best-performing model for each case was selected and evaluated on six randomly selected test sets. Indeed, for each model, the authors used the confidential intervals of performance measures obtained on these test sets for robustness assessment. The findings indicated that the LSTM model was superior to the other networks, with the GRU model being the second-best performer. Regarding complexity, Recurrent Neural Networks (RNNs) were observed to be more complex than CNNs, as they are more memory intensive and require more training time. An interesting aspect of the authors' work was the development of a web-based application. In the back-end system, data from the previous 30 minutes (six observations at 5-minute intervals) were collected and used by a pre-trained LSTM network to predict the next 5-minute closing price of Ethereum. The results were displayed in the front end as live graphs.

Ji et al. (2019) addressed both classification and regression by comparing various state-of-the-art deep learning methods. They utilized a sequence of previous m days (where m = 5, 10, 20, 50, and 100) for each of the 18 Bitcoin blockchain features. For the classification problem, they predicted whether the price would go up or down the next day, while the regression problem was defined as predicting the value of the price for the next day. Training all models with different values of m, the authors found that for regression, shorter sequences yielded better performance, whereas for classification, longer sequences were required. Comparing the results of six different models (Deep Neural Network, LSTM, CNN, a combination of CNN and RNN, etc.) with a few baseline models, they found that LSTM had the best regression performance, and the Deep Neural Network (DNN) was selected as the best classifier.

Another study that covers both trend and price predictions is the study of Zhang et al. (2021) in which they proposed a new model named Weighted and Attentive Memory Channels (WAMC) consisting of three modules. Apart from proposing a new model, the novelty of this work is in the use of data from different cryptocurrencies to predict the closing price of a specific cryptocurrency. In fact, four cryptocurrencies with a high degree of correlation and similar trends were selected for this purpose. Using a moving window of length 7 (representing historical prices) and the closing prices of the four cryptocurrencies as channels, the authors constructed input data samples as 3-dimensional tensors (7, 1, 4). One of the four cryptocurrencies was then selected as the target for prediction. In the attentive memory module of the WAMC, a self-attention mechanism over the outputs of a GRU network was employed to capture important information at different time steps. The outputs of this module were fed into the channel-wise weighting module, where the importance of each channel was determined using a GRU model. The last module performed convolution and pooling operations on the outputs from the previous module, predicting either the price or its movement direction for the target cryptocurrency. The results revealed that the WAMC outperformed all baseline models considered in this study. Moreover, the WAMC was trained and tested three times, each time excluding one of the three modules. The results showed that the performance of the model was worse when any of the modules was omitted, emphasizing the importance of all three modules for optimal predictive performance.

In an extensive investigation conducted by Lahmiri & Bekiros (2020), three different sets of models were analyzed for prediction of the next 5-minute's Bitcoin price using its five previous observations. The authors utilized several complexity measures to quantify the nonlinear dynamics of the Bitcoin series data. For instance, the sample entropy was used to measure the randomness of data. The results of this complexity analysis required the authors to employ advanced machine learning methods that can capture noisy and nonlinear behaviors in data. Out of seven different models examined, the Bayesian Regularization Neural Network (BRNN) was found to have the best performance in predicting Bitcoin's high-frequency price data. The reason mentioned by the authors is that the objective function of the Bayesian optimization in this network penalizes large weights and hence, the network is less likely to overfit.

Kim et al. (2021) utilized various blockchain information related to Ethereum, including transaction volume, mining difficulty, and network activity as features for predicting Ethereum's price. Conducting a stepwise analysis for two machine learning models, namely the Artificial Neural Network (ANN) and Support Vector Machine (SVM), they showed that the ANN model, which incorporates Ethereum- and Bitcoin-specific blockchain information along with macro-economic factors, exhibits the lowest prediction error among all other combinations of features. The authors noted in their study that future research should consider incorporating social media data, such as user sentiment, to further improve the accuracy of Ethereum price predictions.

In a recent study, Oyedele et al. (2023) evaluated the prediction performance of deep learning and boosted tree-based techniques using six cryptocurrency datasets obtained from three different data sources. The use of multiple data sources aimed to investigate the robustness of prediction models in terms of their response to patterns in diverse data sources. Another distinctive aspect of their work was the design of two training sets — one containing peaks and drops of prices, and the other without higher spikes as part of the training set. The authors showed that their models performed well and yielded more accurate results on datasets with more training data, particularly those that included peaks and drops in prices. Furthermore, they concluded that the CNN model was more reliable with limited training data compared to other deep learning and tree-based techniques and it is easily generalizable for predicting the daily close price of several cryptocurrencies.

2.2. Social media and cryptocurrency price prediction

One of the first researchers that studied the predictive power of Twitter sentiment for a large set of cryptocurrencies was Kraaijeveld & de Smedt (2020). The authors investigated the extent to which public Twitter sentiment could be used to predict price returns for the nine largest cryptocurrencies. They collected more than 24 million tweets and implemented several measures to detect and remove tweets generated by bots. The Valence Aware Dictionary and Sentiment Reasoner (VADER) algorithm (Hutto & Gilbert, 2014), a lexicon-based approach, was employed to calculate polarity scores for each tweet. These scores were then aggregated into daily and hourly intervals. By using the bivariate Granger-causality test, the authors concluded that Twitter sentiment had predictive power for three out of the nine cryptocurrencies included in their study.

Ortu et al. (2022) used three sets of input data in their study to predict the price movements of Bitcoin and Ethereum. They trained the pre-trained BERT-base-case model (Devlin et al., 2018) on some labeled benchmark datasets and subsequently used the trained model to extract emotions and sentiments from GitHub and Reddit comments. They built hourly and daily time series data by aggregating sentiments and emotions from multiple comments for each hour and day, respectively. These social media indicators, along with other types of data, were fed into four different models. The study demonstrated that, particularly at daily frequency, the models fed with social media indicators outperformed other models in predicting the price movements of Bitcoin and Ethereum.

Anbaee Farimani et al. (2022) evaluated the effectiveness of using both news sentiment and informative indicators for price regression in Forex and Cryptocurrency markets. They built news sentiment time series by analyzing the probability distribution of news title embedding generated through the FinBERT (Araci, 2019) classifier layer. They showed that incorporating both market data and news sentiment significantly reduced the error for price regression.

Zou & Herremans (2022) employed FinBERT (Araci, 2019) to generate actual word embeddings. They stacked the embeddings of tweets from the same day into 20-dimensional vectors. These stacked embeddings were then fed into a CNN model, while technical indicators and other data were fed into an SVM model. The researchers combined these two models in both sequential and parallel formats to predict extreme price movement. Their results indicate that the sequential model outperformed the SVM model, which used only price and technical analysis data. This suggests that incorporating predictions based on Twitter content improves the overall performance of the model.

Table 13: Recent studies on cryptocurrency price prediction

Reference	Market / Data	Data period / Frequency	Input features	Methodology
Patel et al. (2020)	Litecoin and Monero	Litecoin: August 24, 2016, to February 23, 2020 Monero: January 30, 2015, to February 23, 2020 / Daily	Average price of cryptocurrency for the day	LSTM-GRU hybrid
Chen et al. (2021)	Bitcoin	four important periods for Bitcoin: August 1, 2011, to December 31, 2013 - August 1, 2013, to December 31, 2014 - July 1, 2014, to December 31, 2017 - July 1, 2015, to July 31, 2018 / Daily	1- Technology category: Blockchain information, Public attention 2- Economic category: Macroeconomic indicators, Global currency ratios	ANN and RF for feature selection LSTM for price prediction
Zoumpekas et al. (2020)	Ethereum	Training data: August 8, 2015, to May 28, 2018 Test data were collected in six randomly selected timeframes / 5-minute	Historical price data, transaction and blockchain data, user comments, economic factors, and other community data. Google Trends and Wikipedia	2-layer CNN, 3-layer CNN, LSTM, BiLSTM, Stacked LSTM, GRU
Ji et al. (2019)	Bitcoin	November 29, 2011 to December 31, 2018 / Daily	A sequence of 18 daily value of Bitcoin blockchain features such as block size, the total number of transactions, etc.	DNN, LSTM, CNN, a combination of CNNs and RNNs (CRNN), Residual Network (ResNet), Ensemble of LSTM, DNN and CNN
Zhang et al. (2021)	BTC, Bitcoin Cash, Litecoin, ETH, EOS, and XRP	July 23, 2017 to July 15, 2020 / Daily	A window length (7, 11, 15) of different cryptocurrencies closing prices	A Weighted and Attentive Memory Channels (WAMC) model consisting of three modules
Lahmiri & Bekiros (2020)	Bitcoin	January 1, 2016 to March 16, 2018 / 5-minute	Previous five observations of prices	SVR, Gaussian Poisson Regression (GRP), Regression Tree (RT), kNN, Feedforward Neural Network (FFNN), Bayesian Regularization Neural Network (BRNN), and Radial Basis Function Neural Network (RBFNN)

H. M. Kim et al., 2021	Ethereum	August 11, 2015 to November 28, 2018 / Daily	Ethereum and other cryptocurrencies (Bitcoin, Litecoin, Dashcoin) blockchain information, global currency ratio, and macro-economic development index	ANN, SVM
Oyedele et al. (2023)	Six cryptocurrencies (BTC, ETH, BNBUSD, LTC, XLM, and DOGE)	January 1, 2018 to December 31, 2021	OHLCV and technical indicators	Boosted tree based (AdaBoost, GBM, XGB) and Deep Learning (DFNN, CNN, GRU) techniques
Kraaijeveld & de Smedt (2020)	9 cryptocurrencies	June 4, 2018 to August 4, 2018 / Daily and hourly	Tweets	lexicon based sentiment analysis
Ortu et al. (2022)	Bitcoin, Ethereum	2017-01 to 2021-01 / Daily and hourly	Technical and trading indicators, GitHub, and Reddit comments	BERT for extracting the sentiment, emotion, VAD, Multivariate Attention Long Short-Term Memory Fully Convolutional Network (MALSTM-FCN) for the classification task
Anbaee Farimani et al. (2022)	Forex and Cryptocurrency markets	September 2018 to May 2021 / Hourly	Technical indicators and news sentiment	FinBERT for extracting news sentiment, LSTM for price prediction part
Zou & Herremans (2022)	Bitcoin	2015 to 2021 / Daily	Tweets, OHLCV data and 13 technical indicators, with correlated asset prices such as Ethereum and Gold	FinBERT for extracting tweets sentiment, combination of SVM and CNN for price movement prediction

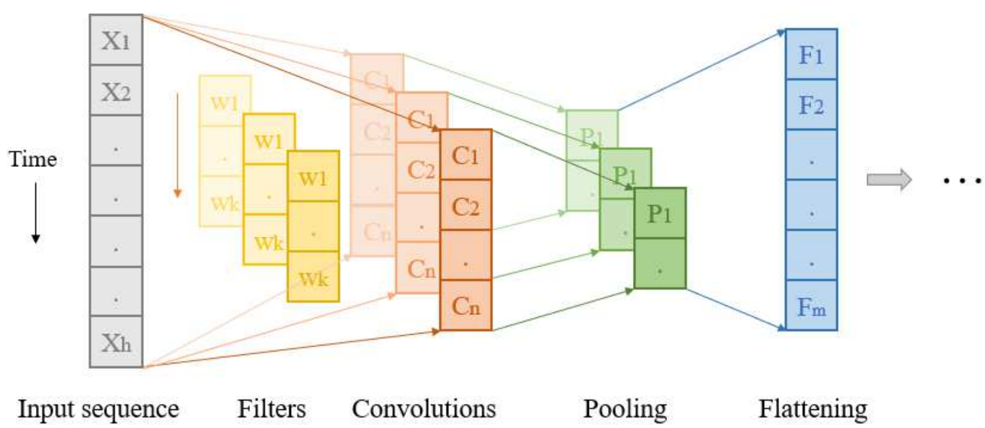
3. Methodology

In this section, we present CNN, sentiment classifiers, and various architectures of our hybrid model. We also explain the process of selecting sentiment classifiers from available pre-trained language models. Subsequently, we describe the hyper-parameter tuning procedure. Finally, we present the performance measure used to assess the performance of each model. To maintain brevity and avoid redundancy, we omit the detailed explanation of LSTM, as it has been comprehensively covered in essay 1.

3.1. Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN), introduced by Lecun et al. (1998), have many applications in image analysis and classification problems. Recently, it has been shown to be also effective for sequence data analysis (Ji et al., 2019). Figure 9 shows a simple one-dimensional CNN architecture.

Figure 9: one-dimensional CNN architecture



In this architecture, the input layer passes a fixed-length sequence from the full time series to the convolution layer. In our context, the input layer is fed with a sequence of h time steps from the Close price and/or sentiment scores. The convolution layer then slides one or multiple filters, each with the size of k, over the input sequence, one step at a time, where $k < h$. Mathematically, a one-dimensional (1-d) convolution operation can be described as the inner product of two vectors: a subsequence from the input sequence (X) with length of k, and the filter, $w \epsilon R^k$. Equation (18) shows the 1-d convolution operation.

$$C_j = \sum_{i=1}^{k} w_i . X_i \tag{18}$$

where i and j are indices of the filter and output sequence, respectively. X_i represents the i-th element in the input sequence X, w_i is the i-th element in the filter (or kernel) of length k, and C_j is the j-th element of the new one-dimensional output sequence (i.e., the convolution). Consequently, the length of the output vectors (shown by n in Figure 9) is not equal to the length of the input sequence. The resulting vectors are somehow the smoothed version of the input sequence that include the most important features and information from that sequence. It is worth mentioning that the 1-d convolution can be applied to more than one input sequence, as the filter(s) only move in one dimension along the axis of time, and the convolution operations are performed independently for each input sequence.

After the convolution operation, an activation function is applied to produce the output (although not explicitly shown in Figure 9). The pooling layer comes after that, combining the output of the previous layer at certain locations into a single neuron by taking the mean, maximum, or minimum of all values at those positions (Pérez-Enciso & Zingaretti, 2019).

Finally, the outputs from the pooling operation are flattened to create a unique vector, which can then be followed by other layers such as a Dense layer to generate predictions. In our case, this flattened layer vector will be concatenated with the output from the LSTM model as part of our hybrid model. That is why we represent the last layer of the CNN with three dots in Figure 9, indicating that it continues beyond what is explicitly shown.

For more in-depth details on the architecture of CNNs, please refer to (Kiranyaz et al., 2019) and (Pérez-Enciso & Zingaretti, 2019). As mentioned by Kiranyaz et al. (2019), one advantage of one-dimensional CNNs compared to two-dimensional CNNs is that one-dimensional CNNs with relatively shallow architectures (i.e., small number of hidden layers and neurons) can effectively learn challenging tasks involving 1-d signals. Therefore, the 1-d CNN part of our proposed model will be shallow and consist of only one layer. On the other hand, using more layers may lead to the loss of important information from the input sequence due to the increased number of convolution operations which will decrease the prediction performance of the proposed model.

3.2. Proposed Hybrid Model

Our proposed model is a hybrid model, and it consists of different layers including LSTM, CNN, Attention, and Dense layers. The overall diagram of the model is exhibited in Figure 10. In the input layer, a history of previous h days of close prices $(C_{t-h}, \ldots, C_{t-1})$ and sentiment data $(S_{t-h}, \ldots, S_{t-1})$ is fed into the LSTM and CNN layers. Subsequently, the output layer generates the close price at day t, denoted as C_t. The rationale behind using both CNN and LSTM layers in parallel is that the former performs well in finding local dependencies, while the latter is effective in capturing long-term dependencies in the data. This approach allows the generation of more informative representations from input features, and having these layers in parallel provides two distinctive representations from these two layers. It is important to remark that while there is only one CNN layer in the model, the number of LSTM layers may vary from one cryptocurrency to another based on hyper-parameter tuning results. The decision to include only one CNN layer is driven by the fact that the convolution operation of the CNN reduces the dimension of data, and the goal is to avoid losing information due to an excessive number of convolutions.

The problem with LSTM arises when dealing with long sequences, as there is a high probability that the initial context is lost by the end of the sequence. The “Attention” technique (Vaswani et al., 2017) addresses this issue by enabling the network to focus on different parts of the input sequence at each stage of the output sequence, thereby, preserving context from the beginning to end. In our hybrid model, there is an optional attention layer after the LSTM layer, allowing the model to learn where to pay attention in the input sequence by relating it to the items in the output sequence. Both our input and output sequences from the LSTM layer have a length of h, and the attention layer guides the LSTM layer to emphasize the most informative items in the input sequence for better representing and relating them to the output sequence.

The output of the attention layer (or LSTM layer if no attention layer is present) and the CNN layer are concatenated. The resulting vector is then passed through a dense layer, followed by a dropout layer to mitigate any potential overfitting issues. In the output layer, a dense layer with a linear activation function generates the predicted value for the close price at day t.

We formulated several architectures for this hybrid model, aiming not only to identify the optimal architecture for each cryptocurrency but also to determine the most effective way of inputting data into the model. Table 14 presents 20 different models that are examined in this study.

For instance, CNN(C)_LSTM_At(S) denotes an architecture featuring an attention layer (At) in the model, where close price (C) and sentiment (S) vectors are fed into the CNN and LSTM layers, respectively.

Figure 10: Overview of the hybrid model with different ways of feeding input data

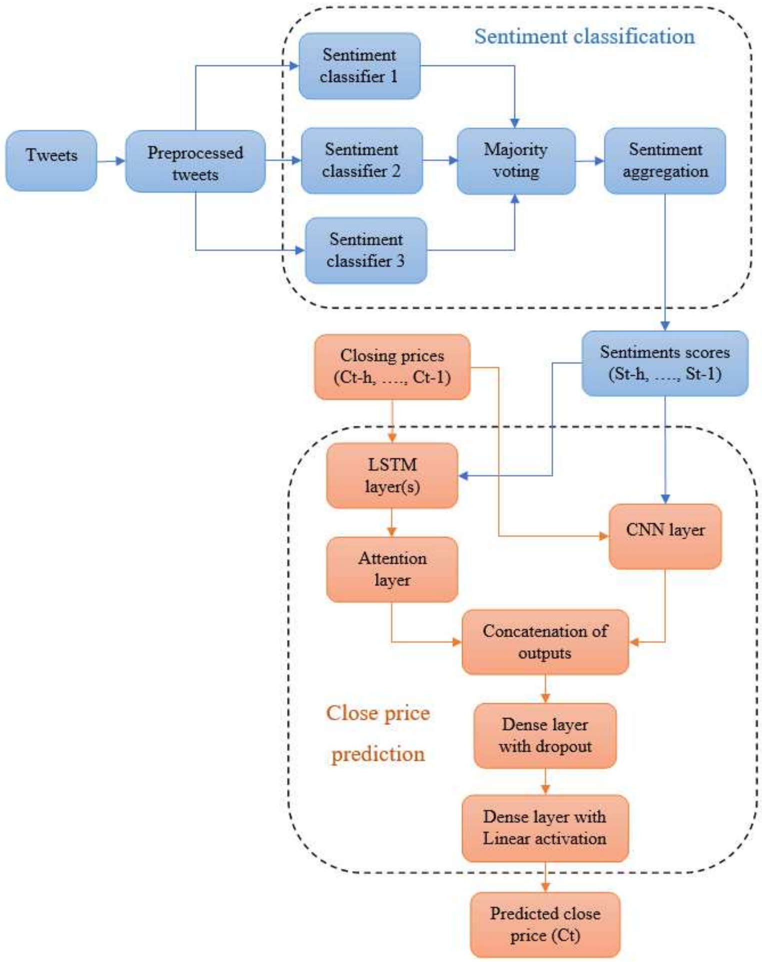


Table 14: Various architectures of the close price prediction model

1.	LSTM(C)
2.	LSTM_At(C)
3.	LSTM(C,S)
4.	LSTM_At(C,S)
5.	CNN(C)_LSTM(C)
6.	CNN(C)_LSTM_At(C)
7.	CNN(C)_LSTM(S)
8.	CNN(C)_LSTM_At(S)
9.	CNN(S)_LSTM(C)
10.	CNN(S)_LSTM_At(C)
11.	CNN(C)_LSTM(C,S)
12.	CNN(C)_LSTM_At(C,S)
13.	CNN(C,S)_LSTM(C)
14.	CNN(C,S)_LSTM_At(C)
15.	CNN(S)_LSTM(C,S)
16.	CNN(S)_LSTM_At(C,S)
17.	CNN(C,S)_LSTM(S)
18.	CNN(C,S)_LSTM_At(S)
19.	CNN(C,S)_LSTM(C,S)
20.	CNN(C,S)_LSTM_At(C,S)

3.3. Hyper-parameter tuning

We decided to perform hyperparameter tuning for only one version of our hybrid model, and we selected the CNN(C)_LSTM_At(C) model as the reference because it has all components and is fed with only close price data. The intention was to identify the optimal version of this reference model, and then, by removing components and adding sentiment data in different ways, we obtained the other 19 versions of our model. Therefore, for each cryptocurrency, the close price data was divided into 80% and 20% for training-validation and testing sets, respectively. The last 20% of the training-validation set was considered as the validation set for hyper-parameter tuning.

The CNN(C)_LSTM_At(C) model was trained on the training set, and the trained model was used to predict close prices in the validation set. More precisely, the last *h* values of the close price from the training set were used to predict the first close price in the validation set. The last *h-1* values of the close price from the training set, along with the first value of the close price in the validation set, were used to predict the second close price in the validation set, and so on. This process generated 58 predicted values, equal to the size of the validation set. To obtain more robust results and mitigate the randomness effect of neural networks, we repeated each training and validation process five times. The optimal set of hyperparameters for each cryptocurrency was found by comparing the average RMSE value calculated over five runs.

The list of hyper-parameters and their respective search interval are presented in Table 15. Using a grid search method and fixing the value of 21 for *h* (history of input data), "MSE" for the loss function, "Adam" as the optimizer, and training for 25 epochs, we tuned the CNN(C)_LSTM_At(C) model 27 times. This is because there is no unique set of hyper-parameters that is optimal for all cryptocurrencies, and the optimization results vary from one cryptocurrency to another. Similar approaches can be seen in the work of Oyedele et al. (2023b). Once the optimal hyper-parameters are found, the entire training-validation set is used to train the models and generate predicted values for the testing set.

Table 15: Hyperparameters and parameters in the tuning process

Hyper-parameter	**Searching interval**
Number of LSTM layers	[1, 2, 3]
Number of units in LSTM hidden layers	[50, 100]
Number of units in dense layers	[64, 128]
Number of filters in the CNN layer	[32, 64]
Filter size in the CNN layer (k)	[2, 3, 4]
Activation functions	['tanh', 'relu']
Batch size	[16, 32]
Fixed parameters	
h = 21 Epoch = 25	Loss = 'MSE' Optimizer = 'Adam' with the default learning rate of 0.001

3.4. Sentiment classifiers

Over the past few years, transformer-based language models have shown remarkable advancements in Natural Language Processing (NLP) tasks and have become the state-of-the-art in this field. Among these models, Bidirectional Encoder Representations from Transformers so-called BERT (Devlin et al., 2018) is the most well-known one. Due to space constraints in this thesis, the details of the BERT architecture are not included here; readers are encouraged to consult the original paper for a comprehensive understanding. BERT was initially pre-trained on BooksCorpus (800M words) (Zhu et al., 2015) and English Wikipedia (2,500M words). The resulting parameters can then be fine-tuned for specific downstream tasks, such as sentiment classification. Fine-tuning involves augmenting the pre-trained BERT model with an additional output layer designed for the target classification task, followed by training the model on a sentiment-labeled text dataset. Devlin et al. (2018) fine-tuned the pre-trained BERT model on the benchmark dataset of Stanford Sentiment Treebank (Socher et al., 2013). This dataset involves a binary single-sentence classification task, consisting of sentences extracted from movie reviews with human annotations of their sentiment. Their results outperformed other state-of-the-art results at the time of BERT's introduction.

The problem with general-purpose language models like BERT is that they have not been trained on domain-specific corpus and consequently, making them less ideal for down-stream tasks in especial domains. Recently, researchers have addressed this limitation by creating domain-specific BERT models. In these models, BERT is pre-trained from scratch using large corpora specific to certain domains. Some examples are SciBERT (Beltagy et al., 2019) trained on a large multi-domain corpus of scientific publications and has shown statistically significant improvements over BERT for several NLP tasks, ClinicalBERT (Huang et al., 2019) pre-trained on clinical notes corpus for the purpose of predicting hospital readmission.

FinBERT (Araci, 2019) is the first finance domain-specific BERT model, pre-trained on a large financial corpus called TRC2-financial by the author. This corpus, a subset of Reuters' TRC2, consists of 1.8 million news articles published by Reuters between 2008 and 2010. The pre-trained FinBERT model was also fine-tuned on Financial PhraseBank from (Malo et al., 2013) for sentiment classification tasks within the financial domain. According to the findings reported by Araci (2019), FinBERT exhibited a 15% improvement over generic BERT models in financial sentiment analysis tasks. This success, coupled with its application in similar studies such as (Zou & Herremans, 2022) and (Anbaee Farimani et al., 2022), motivated the use of the fine-tuned version of FinBERT as one of the sentiment classifiers in the current study. It is important to note that this model is not further fine-tuned, and the original fine-tuned version is applied directly to the Twitter data. Given that the tweets in our study are unlabeled, and FinBERT has been already fine-tuned on the Financial PhraseBank dataset specifically for sentiment classification, no additional fine-tuning is performed by us. The absence of true sentiment labels for the tweets prevents the assessment of FinBERT's performance. Consequently, we cannot rely on the sentiment labels generated by only one model, i.e., the FinBERT. To address this limitation, we decided to employ two additional fine-tuned language models as our second and third sentiment classifiers. Then, by taking a majority vote approach, we can find the final sentiment label for each tweet. This ensemble approach enhances reliability and mitigates dependence on a single model prediction (see the sentiment classification part in Figure 10).

In selecting the other two sentiment classifiers, our criterion was to utilize language models specifically fine-tuned for social media data, particularly Twitter data. This approach allowed us to address the sentiment classification problem not only from the financial domain perspective, covered by FinBERT, but also from the social media context. In this regard, one of the well-known language models is a RoBERTa-base model, trained on 124 million tweets from January 2018 to December 2021 (Loureiro et al., 2022). This model was also fine-tuned for sentiment analysis using the TweetEval benchmark (Barbieri et al., 2020), a common evaluation benchmark for language models on Twitter data. According to the findings in (Loureiro et al., 2022), this model achieved the highest accuracy (73.7%) in the sentiment classification task on the TweetEval dataset compared to other existing state-of-the-art methods. We employ this fine-tuned model as our second sentiment classifier and will use the term RoBERTa_tweet for it. It is worth mentioning that there is another state-of-the-art language model called BERTweet (Quoc Nguyen et al., 2020), which has been trained on over 900 million tweets posted between 2013 and 2019. However, we decided to employ RoBERTa_tweet since it was pre-trained with most recent tweets posted between 2018 and 2021.

The third sentiment classifier employed in our study is RoBERTuito (Pérez et al., 2021), a pre-trained language model designed for user-generated text in Spanish. This model was trained on an extensive dataset comprising over 500 million tweets. Evaluation results of RoBERTuito on English data, specifically on the SemEval 2017 Task-4 dataset (Rosenthal et al., 2017), demonstrated competitive performance when compared to monolingual models such as BERT, BERTweet, and RoBERTa. Therefore, we decided to use RoBERTuito as our third sentiment classifier in this study.

Table 16: Sentiment classifiers employed in the close price prediction study

Sentiment Classifier	Pre-trained dataset	Sentiment analysis dataset
FinBERT (Araci, 2019)	TRC2-financial (a subset of Reuters' TRC2)	Financial PhraseBank (Malo et al., 2013)
RoBERTa_tweet (Loureiro et al., 2022)	123.86 million tweets until the end of 2021	TweetEval benchmark (Barbieri et al., 2020)
RoBERTuito (Pérez et al., 2021)	Over 500 million Spanish tweets	SemEval 2017 Task-4 dataset (Rosenthal et al., 2017)

3.5. Performance metric

To evaluate prediction performance of the models, we used the Root Mean Squared Error (RMSE) measure specified in Equation (19):

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(C_i - \hat{C}_i\right)^2} \tag{19}$$

where C_i and $\hat{C}_i$ are the real and predicted values of the close price at time *i*, respectively. *n* is the number of predictions which is equal to the size of testing sample for each cryptocurrency, that is 73 (20% of one year's data). Lower RMSE values indicate better predictive performance, as they suggest that the predicted values are closer to the actual values.

4. Data and results

Previous studies have primarily focussed on Bitcoin, Ethereum and a few other cryptocurrencies since they are the most traded cryptocurrencies in the market. Aiming to do experiments on a large set of cryptocurrencies, we collected Twitter and price data for 27 different cryptocurrencies. The details of the data collection are outlined below.

4.1. Price data

In the first step, we prepared a list of 100 most traded cryptocurrencies with available data on the website "www.investing.com". Subsequently, these cryptocurrencies were sorted based on their market entrance dates. For this study, we selected the 27 oldest cryptocurrencies from this list, as listed below:

Bitcoin, XRP (Ripple), Monero, Ethereum, Ethereum Classic, Litecoin, Zcash, Stellar, Dash, Tether, IOTA, EOS, OMG Network, Bitcoin Cash, Waves, Neo, Binance Coin, TRON, Cardano, Maker, Filecoin, Decentraland, Chainlink, Tezos, Dai, Enjin Coin, Theta.

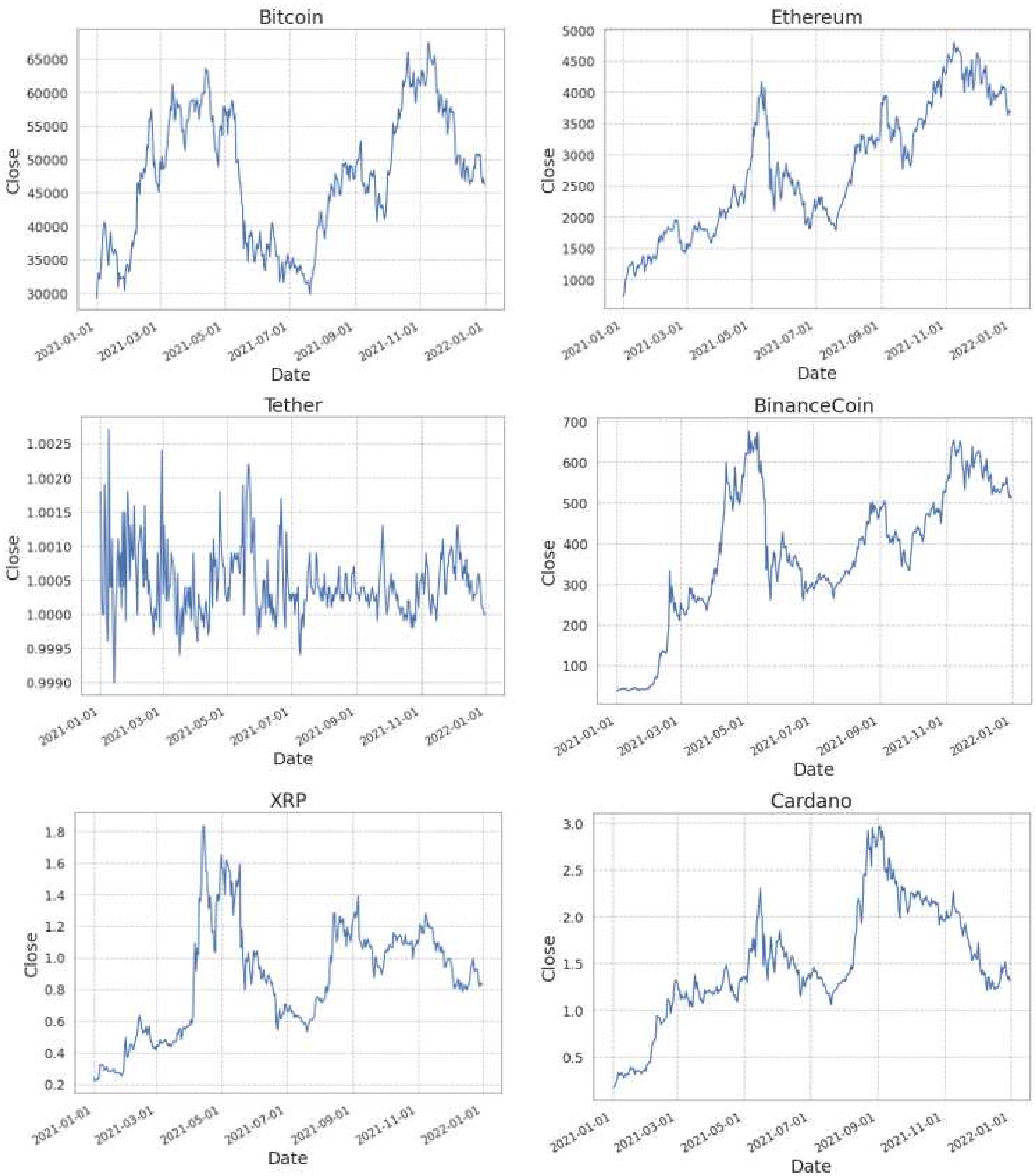
We collected daily "Close price", "High price", "Low price", "Open price", and "Volume" of trades for each cryptocurrency over a one-year period from 2021-01-01 to 2021-12-31. The "Close price" was selected as the target variable for prediction in alignment with similar studies (e.g., Patel et al., 2020; Chen et al. 2021; Ji et al., 2019). Summary statistics of the Close price data are available in Table 17. Figure 11 presents the Close price trends of the top six cryptocurrencies over the entire data period. As can be seen in this figure, cryptocurrencies are highly volatile and predicting their future behavior with a high accuracy is a difficult task. After close look at the Close price data for all 27 cryptocurrencies, no missing values were identified. The only data pre-processing applied was scaling, ensuring that Close prices fall within the range of 0 to 1 (See Equations (21) and (22)).

Table 17: Summary statistics of close prices for all 27 cryptocurrencies

Cryptocurrency	Number of Observations	Minimum	Maximum	Mean	Standard Deviation
Binance Coin	365	37.72	676.6	378.3	169.0
Bitcoin	365	29359.9	67528	47411	9761.8
Bitcoin Cash	365	341.53	1547	605.7	187.4
Cardano	365	0.175	2.965	1.499	0.6151
Chainlink	365	11.84	52.26	26.63	7.0149
Dai	365	0.9946	1.005	1.000	0.0013
Dash	365	86.92	443.6	194.0	65.49
Decentraland	365	0.0788	5.195	1.212	1.152
Enjin Coin	365	0.1320	4.490	1.758	0.9088
EOS	365	2.503	14.47	4.664	1.598
Ethereum	365	729.1	4808	2777	1024
Ethereum Classic	365	5.692	133.8	42.15	23.81
Filecoin	365	21.05	190.9	69.63	39.24
IOTA	365	0.2856	2.515	1.205	0.4462
Litecoin	365	107.3	386.8	185.7	48.09
Maker	365	581.0	5985	2757	856
Monero	365	125.5	483.7	247.7	64.50
Neo	365	14.42	122.8	46.24	20.40
OMG Network	365	2.369	19.10	7.247	3.490
Stellar	365	0.1275	0.7317	0.3631	0.1028
Tether	365	0.9990	1.003	1.000	0.0005
Tezos	365	1.997	8.699	4.535	1.487
Theta	365	1.743	14.19	6.563	2.795

Tron	365	0.02686	0.1639	0.0802	0.0293
Waves	365	5.395	36.49	17.85	7.539
XRP	365	0.2210	1.836	0.8670	0.3485
Zcash	365	56.78	319.6	150.8	50.48

Figure 11: Close price trends of cryptocurrencies over the entire period of study



4.2. Twitter data

We employed the open-source Python library snscrape[3] for web scraping to collect tweets related to cryptocurrencies. For each cryptocurrency, we defined search key parameters using a set of hashtags based on their names and symbols. For instance, the search key for Bitcoin included *"#BTC OR #btc OR #BITCOIN OR #Bitcoin"*. Given that our price dataset covered the interval from January 1, 2021, to December 31, 2021, we used the same time frame for Twitter data collection. To manage computational constraints, we set a limit on the number of tweets collected for each cryptocurrency. Specifically, we collected a maximum of 5000 tweets per day for each cryptocurrency to avoid extended processing times, especially for highly discussed cryptocurrencies like Bitcoin and Ethereum. For each tweet, we collected features such as the date and time of posting (datetime), the text of the tweet, tweet ID, username, URL of the tweet, and various other attributes. While this study primarily utilized the text of the tweet along with its posting datetime, we plan to leverage the other features in our future work. To ensure clean and less noisy text data input for our models, we performed several pre-processing (cleaning) operations which are available in Table 18 with their results on an example tweet.

[3] https://github.com/JustAnotherArchivist/snscrape

Table 18: Results of tweets pre-processing operations with an example

tweet: "@Rookie b Yaaaaay you're lucky. this will totally explode 🤯 June 3rd https://t.co/0l via @YouTube #Bitcoin $BTC"	
Operation	**Pre-processed tweet**
Convert all English alphabet characters to lower case	"@rookie b yaaaaay you're lucky. this will totally explode 🤯 june 30th https://t.co/0lp via @youtube #bitcoin $btc"
Remove links and mentions	" b yaaaaay you're lucky. this will totally explode 🤯 june 30th via #bitcoin $btc"
Remove ticker symbols ($)	" b yaaaaay you're lucky. this will totally explode 🤯 june 30th via #bitcoin "
Remove hashtag symbols (#)	" b yaaaaay you're lucky. this will totally explode 🤯 june 30th via bitcoin "
Replace any emojis with the text they represent	" b yaaaaay you're lucky. this will totally explode :exploding_head: june 30th via bitcoin "
Expand contraction words	'b yaaaaay you are lucky. this will totally explode :exploding_head: june 30th via bitcoin'
Reduce three or more repetitions of any character to two characters	'b yaay you are lucky. this will totally explode :exploding_head: june 30th via bitcoin'
Remove words that contain numbers	'b yaay you are lucky. this will totally explode :exploding_head: june via bitcoin'
Remove words with one character	'yaay you are lucky. this will totally explode :exploding_head: june via bitcoin'
Remove multiple spaces	'yaay you are lucky. this will totally explode :exploding_head: june via bitcoin'

Table 19: Tweets count pre-cleaning, post-cleaning, and post-majority voting in sentiment analysis

Cryptocurrency	Number of tweets before cleaning	Number of tweets after cleaning	Data loss due to cleaning	Number of tweets after majority voting	Data loss due to majority voting
Binance Coin	1,759,614	1,745,908	0.78%	1,737,513	0.48%
Bitcoin	1,783,500	1,781,118	0.13%	1,766,382	0.83%
Bitcoin Cash	453,194	452,454	0.16%	449,936	0.56%
Cardano	1,328,099	1,327,521	0.04%	1,316,923	0.80%
Chainlink	1,005,240	970,112	3.49%	960,764	0.96%
Dai	28,048	27,857	0.68%	27,687	0.61%
Dash	74,455	73,763	0.93%	73,118	0.87%
Decentraland	297,174	295,351	0.61%	292,541	0.95%
EOS	109,743	109,100	0.59%	108,122	0.90%
Enjin Coin	100,388	98,553	1.83%	97,586	0.98%
Ethereum	1,813,200	1,808,472	0.26%	1,792,384	0.89%
Ethereum Classic	1,808,520	1,808,202	0.02%	1,793,220	0.83%
Filecoin	123,793	122,986	0.65%	122,082	0.74%
IOTA	432,607	432,236	0.09%	428,595	0.84%
Litecoin	1,194,347	1,191,249	0.26%	1,181,627	0.81%
Maker	45,930	45,482	0.98%	45,110	0.82%
Monero	238,864	237,856	0.42%	236,445	0.59%
Neo	72,337	71,437	1.24%	70,897	0.76%
OMG Network	67,426	66,556	1.29%	65,600	1.44%
Stellar	188,307	186,560	0.93%	184,843	0.92%
Tether	1,642,775	1,641,160	0.10%	1,635,580	0.34%
Tezos	10,493	10,475	0.17%	10,418	0.54%
Theta	163,307	160,940	1.45%	159,336	1.00%
Tron	17,907	17,903	0.02%	17,810	0.52%
Waves	78,217	76,308	2.44%	75,906	0.53%
XRP	1,819,418	1,817,155	0.12%	1,800,384	0.92%
Zcash	130,429	130,115	0.24%	129,239	0.67%
Total (average)	**16,787,332**	**16,706,829**	**(0.74%)**	**16,580,048**	**(0.78%)**

5. Results

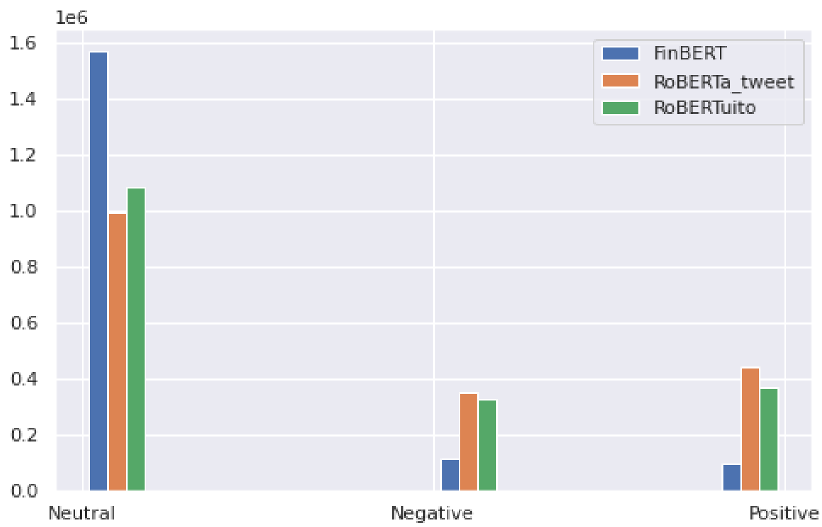
In this section, we assess the prediction performance of 20 different versions of our proposed models introduced earlier. The simplest model in Table 14, i.e., LSTM(C) is considered as our base model, and we are interested in comparing and ranking the prediction performance of other 19 models with respect to this model.

Our research questions are defined as follows:

- Does the inclusion of sentiment data impact the prediction performance of the models?
- Is presence of a CNN layer helpful in improving the prediction performance of the models?
- How does the incorporation of an attention layer affect the prediction performance of the models?
- What is the optimal length of the input sequence (history parameter, *h*) for effective model performance?
- How should the input layer be configured for each cryptocurrency to optimize prediction performance?

As explained earlier, three fine-tuned language models were employed to categorize the sentiment of each tweet into Negative, Neutral, and Positive. Given that our tweets were unlabeled, we could not fine-tune these language models on our specific dataset. Instead, we relied on the available models fine-tuned on similar datasets to ours and we employed them directly to predict sentiment labels for our tweets. However, by applying a majority voting technique, we only kept tweets that were given the same label by at least two of the classifiers. This way, we obtained more robust and reliable sentiment labels. Figure 12 exhibits the results for Bitcoin tweets. Across all cryptocurrencies, the data loss due to majority voting was minimal, consistently less than 1.5% (See Table 19).

Figure 12: Frequency plot of predicted sentiments by each sentiment classifier – Bitcoin's Twitter data as an example



Our model accepts time series data as its input, and we need to convert sentiment data into numerical values interpretable by the deep learning models. Following the work of Hiew et al. (2019), we used Equation (20) to aggregate sentiments from multiple tweets on each day and calculated a sentiment score at day t. Accordingly, our price and sentiment data have the same granularity.

$$Sentiment\ score_t = \frac{Pos_t - Neg_t}{Pos_t + Neu_t\ + Neg_t} \tag{20}$$

Pos_t, Neu_t, and Neg_t indicate the number of tweets classified as positive, neutral, and negative sentiments, respectively, on day t. Appendix B1 shows the sentiment aggregation results for Bitcoin data as an example.

As presented in Table 14, we have 20 different versions of our model, ranging from the simplest (LSTM(C)) to the most complex (CNN(C,S)_LSTM_At(C,S)). For each cryptocurrency, we used its optimal set of hyper-parameters and utilized close prices and/or sentiment scores as input data to evaluate the prediction performance of all 20 models on their respective testing set. Since sentiment and price data are in different scales, we employed the "min-max normalization" method to scale them into a range of 0 and 1. This also assists the optimization algorithm of our models to converge faster. Thus, the following equations were used:

$$\tilde{C}_t = \frac{C_t - C_{min}}{C_{max} - C_{min}} \tag{21}$$

$$\tilde{S}_t = \frac{S_t - S_{min}}{S_{max} - S_{min}} \tag{22}$$

where:

- $\tilde{C}_t \in [0, 1]$, C_{min}, and C_{max} are the standardized, minimum, and maximum values of close prices, respectively.
- $\tilde{S}_t \in [0, 1]$, S_{min}, and S_{max} are the standardized, minimum, and maximum values of sentiment scores, respectively.

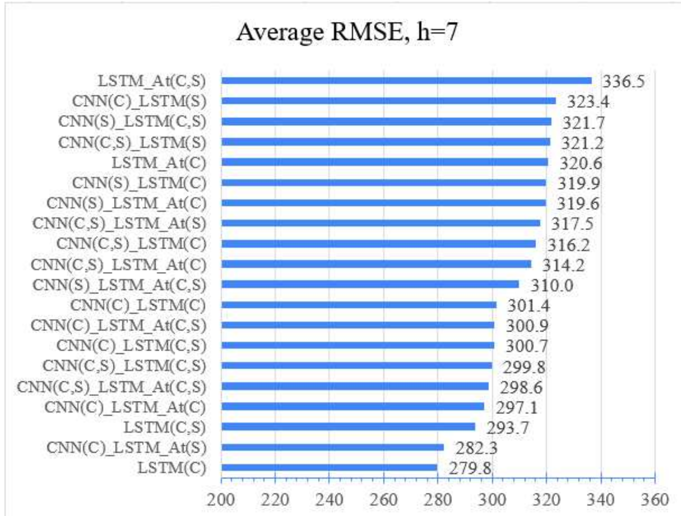
To examine the effect of input data length, we trained and evaluated models using three different values for the history parameter, h. More specifically, we set $h = 7$, $h = 14$, and $h = 21$. These values correspond to using data from one, two, and three weeks before, respectively, to predict today's close price. Training parameters, including Epoch = 25, Loss = "MSE", and Optimizer = "Adam" were fixed. To ensure robust RMSE values and reduce the randomness effects in neural networks training, we repeated the training and evaluation of each model 30 times. The average RMSE values over these 30 runs were recorded as the final RMSE values for each model. After obtaining RMSE values for each cryptocurrency, we calculated the average RMSE values across all cryptocurrencies. The results were then ranked and presented as bar plots in Figure 13. In addition, Figure 14 displays the standard deviations of the RMSE values.

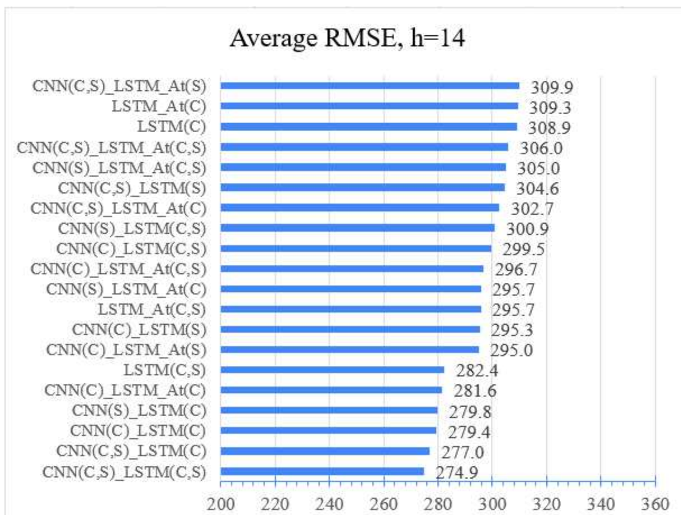
Comparing the average RMSE values in Figure 13 reveals that using data from the past 14 days for predicting today's close price results in the lowest RMSE values on average. Moreover, by comparing the standard deviation of the RMSE values in Figure 14, we can see that the models with $h = 14$ are more stable because they show lowest values for the standard deviation on average. Therefore, for both accuracy and stability considerations, using a history of 14 days of data is the best compared to using shorter ($h = 7$) and longer ($h = 21$) histories. Looking at the bar plot of the average RMSE values for the case of $h = 7$, the LSTM(C) as the base model of this study outperforms all other 19 models. This means that our models perform less effectively with short sequences of input data. However, with $h = 14$, 17 out of 19 models outperform the LSTM(C) model.

Looking at the best five models in the case of $h = 14$, the effectiveness of adding the CNN layer in reducing the RMSE values is detectable. The effectiveness of the attention layer becomes apparent in the case of $h = 21$, where 4 out of the 5 best-performing models incorporate the attention layer in their structure. This suggests that the attention layer enhances model performance, particularly when applied to longer sequences of input data. While no general conclusions can be drawn about the overall effectiveness of using sentiment scores to improve RMSE values, we will delve into this aspect in the discussion part of the study, where we analyze the results at the cryptocurrency level rather than considering the results averaged across all cryptocurrencies.

Despite the effectiveness of $h = 14$ in yielding more accurate and stable prediction values on average, the lowest RMSE value was achieved in the case of $h = 21$. Specifically, the CNN(C,S)_LSTM_At(S) model achieved the lowest average RMSE value of 268.5. While this model is the most stable one with a standard deviation of 1176.5 (See Figure 14, for $h = 21$), its performance diminishes with shorter input sequences, particularly for $h = 14$, where it records the highest RMSE value on average. It is worth mentioning that the objective of this study is not to find one unique best model that performs well for all cryptocurrencies and various lengths of input sequences. Due to the highly volatile nature of the cryptocurrency markets, finding a model with consistently high performance for all cryptocurrencies is almost impossible. Instead, we aim to investigate the different architectures of our proposed models and find the optimal architecture and input layer configuration for each specific cryptocurrency. Therefore, the conclusions drawn above are based on average RMSE values calculated over 27 cryptocurrencies. This implies that the CNN(C,S)_LSTM_At(S) model may not be the best-performing model for all 27 cryptocurrencies. Further details are discussed in the next section.

Figure 13: Average RMSE values for each model under h = 7, h = 14, and h = 21





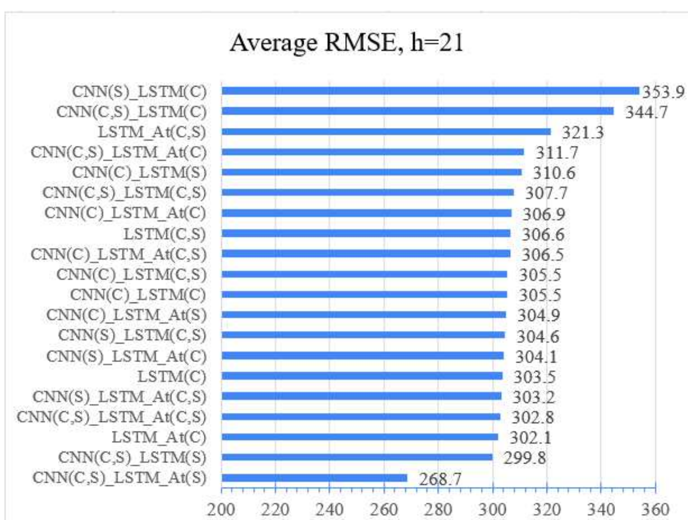
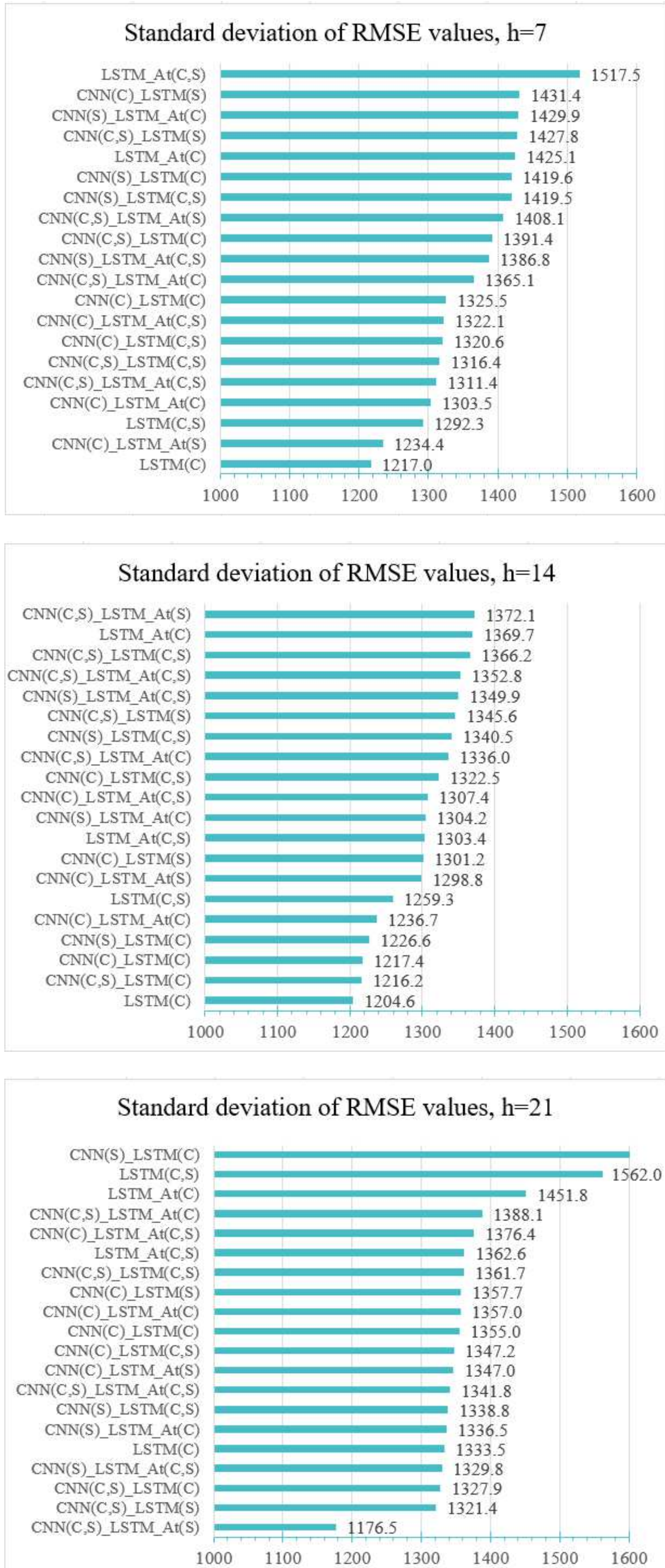


Figure 14: Standard deviation of RMSE values for each model under h = 7, h = 14, and h = 21



6. Discussion

Table 20, Table 21, and Table 22 present the best model and its corresponding RMSE value for each cryptocurrency under different settings of h. In addition, considering the LSTM(C) as the base model, the RMSE reduction value from the base model to the best model is reported in these tables. The results are sorted based on the RMSE reduction values in each table. Comparing the RMSE reduction values for $h = 7$, $h = 14$, and $h = 21$ reveals that in cases of $h = 21$ and $h = 14$, our models could improve the RMSE value of the base model more significantly compared to the case of $h = 7$. This aligns with our earlier findings where we observed that our models underperform the base model when fed with short sequences of input data.

In each table, models that incorporate sentiment scores (denoted by "S") are highlighted in green. Between 70% to 78% of the best models utilize the sentiment data as part of their input, feeding it into either the CNN or LSTM layer. This underscores the potential of extracting emotions and sentiments from Twitter posts to enhance the accuracy of cryptocurrency price forecasts.

Table 20: The best and base models for each cryptocurrency along with their corresponding RMSE values (h = 7)

Cryptocurrency	**Best model**	**RMSE**	**RMSE of base model**	**RMSE reduction**
OMG Network	CNN(S)_LSTM(C,S)	2.821	3.657	22.84%
Decentraland	CNN(S)_LSTM(C,S)	4.104	5.251	21.84%
Litecoin	CNN(C,S)_LSTM_At(C)	20.25	23.29	13.05%
Tron	CNN(S)_LSTM(C)	0.0092	0.0102	10.09%
Enjin Coin	CNN(S)_LSTM_At(C,S)	0.9544	1.058	9.76%
Binance Coin	LSTM_At(C,S)	56.79	62.65	9.35%
Theta	CNN(S)_LSTM(C)	1.031	1.134	9.04%
Cardano	CNN(S)_LSTM(C)	0.2217	0.2436	8.99%
Dash	CNN(C,S)_LSTM(C,S)	29.54	32.37	8.75%
Stellar	CNN(S)_LSTM(C)	0.0399	0.04359	8.53%
Chainlink	CNN(S)_LSTM(C)	4.198	4.586	8.47%
Waves	CNN(C)_LSTM(S)	2.682	2.879	6.86%
Tether	CNN(C)_LSTM_At(C)	0.00030	0.00032	6.62%
Bitcoin	LSTM(C,S)	6337	6731	5.86%
Ethereum Classic	CNN(C,S)_LSTM(C)	7.365	7.790	5.45%
Maker	CNN(C)_LSTM(S)	257.6	272.4	5.45%
XRP	CNN(C,S)_LSTM(S)	0.1023	0.1078	5.16%
Ethereum	LSTM_At(C)	561.89	591.4	4.99%
IOTA	CNN(C)_LSTM(C)	0.0997	0.1046	4.67%
Neo	LSTM_At(C)	8.584	8.908	3.65%

Dai	CNN(S)_LSTM_At(C)	0.0014	0.0014	3.14%
Bitcoin Cash	CNN(S)_LSTM(C,S)	93.69	96.22	2.63%
Zcash	CNN(C)_LSTM_At(S)	37.83	38.81	2.50%
Filecoin	CNN(S)_LSTM(C)	13.75	13.99	1.75%
Monero	LSTM_At(C)	30.90	31.36	1.49%
Tezos	CNN(C)_LSTM(C)	0.5710	0.5777	1.16%
EOS	LSTM(C)	0.8688	0.8688	0.00%
Average		**276.8**	**293.7**	

Table 21: The best and base models for each cryptocurrency along with their corresponding RMSE values (h = 14)

Cryptocurrency	**Best model**	**RMSE**	**RMSE of base model**	**RMSE reduction**
Decentraland	CNN(S)_LSTM(C,S)	3.275	5.413	39.49%
OMG Network	LSTM(C,S)	2.594	3.379	23.24%
Enjin Coin	LSTM_At(C,S)	0.8526	1.054	19.11%
Cardano	CNN(S)_LSTM(C)	0.2132	0.2487	14.25%
Ethereum	LSTM(C,S)	469.3	546.9	14.19%
Binance Coin	CNN(S)_LSTM_At(C)	54.42	62.86	13.43%
Bitcoin	CNN(S)_LSTM(C,S)	6275	6985	10.16%
Zcash	CNN(C,S)_LSTM(C)	34.80	38.59	9.81%
Ethereum Classic	CNN(C)_LSTM(C)	7.270	7.934	8.37%
XRP	CNN(S)_LSTM(C)	0.0988	0.1064	7.15%
Theta	CNN(C,S)_LSTM(C,S)	1.090	1.1734	7.08%
Tron	CNN(S)_LSTM(C)	0.0085	0.0091	6.57%
Dai	CNN(S)_LSTM_At(C)	0.0015	0.0016	5.60%
Chainlink	CNN(S)_LSTM(C)	4.155	4.327	3.97%
Maker	CNN(S)_LSTM(C)	248.2	257.8	3.72%
Bitcoin Cash	LSTM_At(C,S)	89.46	92.22	2.99%
Neo	CNN(S)_LSTM_At(C,S)	8.660	8.911	2.82%
Stellar	LSTM_At(C)	0.0412	0.0422	2.28%
EOS	LSTM_At(C)	0.8197	0.8384	2.23%
Filecoin	CNN(C)_LSTM_At(S)	14.13	14.41	1.96%
Tether	CNN(C)_LSTM_At(C,S)	0.00031	0.00032	1.71%
Dash	LSTM(C,S)	31.37	31.84	1.46%
Waves	CNN(S)_LSTM(C)	2.835	2.873	1.34%
Monero	LSTM_At(C,S)	33.27	33.63	1.07%
IOTA	LSTM(C)	0.1064	0.1064	0.00%
Litecoin	LSTM(C)	23.96	23.96	0.00%
Tezos	LSTM(C)	0.5771	0.5771	0.00%
Average		**270.6**	**300.9**	

Table 22: The best and base models for each cryptocurrency along with their corresponding RMSE values (h = 21)

Cryptocurrency	Best model	RMSE	RMSE of base model	RMSE reduction
Decentraland	LSTM_At(C,S)	3.778	5.805	34.91%
OMG Network	CNN(S)_LSTM(C)	2.292	3.125	26.68%
EnjinCoin	LSTM_At(C)	0.8953	1.109	19.28%
Bitcoin	LSTM_At(C,S)	6131	7566	18.97%
Bitcoin Cash	CNN(S)_LSTM(C,S)	87.53	101.7	13.96%
Cardano	CNN(S)_LSTM(C)	0.2208	0.2564	13.87%
Zcash	CNN(C)_LSTM_At(S)	33.73	39.02	13.56%
Waves	CNN(C)_LSTM_At(C)	2.812	3.239	13.20%
Binance Coin	CNN(C)_LSTM(S)	56.90	64.97	12.43%
Neo	LSTM_At(C,S)	7.866	8.744	10.04%
Tron	CNN(C)_LSTM_At(C,S)	0.0086	0.0095	9.49%
Ethereum	LSTM(C,S)	465.7	512.9	9.20%
Dai	CNN(S)_LSTM_At(C)	0.0014	0.0016	8.99%
EOS	CNN(S)_LSTM_At(C)	0.8487	0.9166	7.41%
Tether	CNN(C,S)_LSTM_At(C)	0.00032	0.00035	7.07%
Monero	LSTM_At(C)	29.00	31.04	6.57%
IOTA	CNN(C,S)_LSTM(C,S)	0.1003	0.1056	5.05%
Dash	LSTM_At(C)	29.68	31.04	4.36%
Stellar	CNN(S)_LSTM_At(C)	0.0371	0.0384	3.44%
XRP	CNN(C,S)_LSTM(S)	0.1019	0.1055	3.40%
Filecoin	LSTM_At(C)	13.82	14.20	2.69%
Tezos	CNN(C)_LSTM(S)	0.5600	0.5743	2.49%
Ethereum Classic	LSTM(C,S)	7.321	7.497	2.35%
Theta	LSTM(C,S)	0.9804	1.002	2.20%
Maker	LSTM(C,S)	253.7	256.9	1.25%
Chainlink	LSTM(C)	3.870	3.870	0.00%
Litecoin	LSTM(C)	21.15	21.15	0.00%
Average		**264.9**	**321.3**	

For robust comparisons, we employed a paired sample t-test on the results reported in Table 20, Table 21, and Table 22. In a paired sample t-test, each row of the two samples corresponds to the same subject; in our case, two RMSE values correspond to the same cryptocurrency. To increase the test's power, a one-tailed version of the test was used. Before interpreting the test results, it should be noted that wherever the RMSE term is mentioned, it refers to the average of RMSE values calculated across all 27 cryptocurrencies. We define the following hypotheses:

$$H_o: \overline{RMSE}_{best_{model}} - \overline{RMSE}_{LSTM(C)} \geq 0 \quad (23)$$

$$H_a: \overline{RMSE}_{best_{model}} - \overline{RMSE}_{LSTM(C)} < 0$$

where $\overline{RMSE}_{best_{model}}$ is the average of RMSE value calculated over the RMSE values of the best models, and $\overline{RMSE}_{LSTM(C)}$ is the average RMSE value calculated over the RMSE values of the LSTM(C) model obtained for 27 cryptocurrencies.

The p-values reported in the second column of Table 23 are greater than 5%, suggesting that the null hypothesis is not rejected. However, a closer examination of the RMSE values revealed that the Bitcoin and Ethereum are like outliers, influencing the t-test results. Therefore, we excluded the RMSE values of these cryptocurrencies from our t-test. The third and fourth columns of Table 23 show improved p-values, indicating that we can now reject the null hypothesis at the 5% level of significance for all h cases. Consequently, the best models have significantly reduced the RMSE value of the base model (LSTM(C)).

Table 23: P-values from t-tests comparing average RMSE of the best models with the base model (LSTM(C))

Sequence length	p-value with all cryptocurrencies	p-value after removing Bitcoin	p-value after removing Bitcoin and Ethereum
h = 7	0.1272	0.0292**	0.0190**
h = 14	0.1300	0.0883*	0.0132**
h = 21	0.1490	0.0432**	0.0121**

**, and * indicate a rejection of null hypothesis at the 5%, and 10% level of significance

We also defined the following hypotheses to compare the average RMSE values of the best models that we obtained for three different lengths of input sequences:

$$H_o: \overline{RMSE}_{h=14} - \overline{RMSE}_{h=7} \geq 0 \tag{24}$$
$$H_a: \overline{RMSE}_{h=14} - \overline{RMSE}_{h=7} < 0$$

$$H_o: \overline{RMSE}_{h=21} - \overline{RMSE}_{h=7} \geq 0 \tag{25}$$
$$H_a: \overline{RMSE}_{h=21} - \overline{RMSE}_{h=7} < 0$$

$$H_o: \overline{RMSE}_{h=21} - \overline{RMSE}_{h=14} \geq 0 \tag{26}$$
$$H_a: \overline{RMSE}_{h=21} - \overline{RMSE}_{h=14} < 0$$

The p-values reported in Table 24 indicate that the null hypotheses in (24) and (25) are rejected at the significance level of 10%. Therefore, using input sequences with lengths of $h =$

14 and $h = 21$ reduces the RMSE values of the best models on average compared to input data with a length of $h = 7$. The null hypothesis in (26) is not rejected and hence, there is no significant difference between the average RMSE value achieved for the best models in cases of $h = 14$ and $h = 21$.

Table 24: P-values from t-tests comparing average RMSE of the best models across varied *h* values

	RMSE_7	**RMSE_14**	**RMSE_21**
RMSE_7	-	0.0703*	0.0826*
RMSE_14		-	0.1369
RMSE_21			-

* indicates a rejection of null hypothesis at the 10% level of significance

7. Conclusion

We fill a significant knowledge gap and make two main contributions to the literature of cryptocurrency price prediction: 1) from the methodological point of view, we introduced a new hybrid model with a flexible architecture that can be customized for different cryptocurrencies, 2) from the experimental aspect, we analyzed a large set of cryptocurrencies along with their related tweets posted between 2021-01-01 and 2021-12-31. We proposed a hybrid deep learning-based model for predicting cryptocurrencies daily close prices. Our model can have 20 different versions and each cryptocurrency benefits from the best-suited model structure. Through comprehensive evaluations, considering input sequences of 7, 14, and 21 previous days, we find that longer sequences result in more accurate predictions. Specifically, a history of 14 days proves to be optimal for achieving higher prediction accuracy on average.

While incorporating the CNN layer was found to be more useful for the case of $h = 14$, the attention layer improved the prediction performance when longer sequences of input data were fed into the models, i.e., $h = 21$. We concluded that for more than 70% of the cryptocurrencies analyzed in this study, the use of sentiment data results in higher prediction performances. This indicates that social media posts and in particular, tweets, play a significant role in people's decision-making for cryptocurrency market trading. Thus, this study can be a guide for researchers and academics to take into consideration the effect of appropriate social media data in cryptocurrency market prediction.

There are certain limitations in our study. The sentiment analysis methods employed may struggle to understand irony and sarcasm present in tweets. This limits their availability in producing very accurate sentiments. In addition, there are unrelated contents in tweets such as promotions and spams, preventing us to extract more meaningful sentiments from tweets. Some researchers believe that tweets generated by bots should be removed from sentiment analysis. For the future research, we would like to develop a bot detection classifier and apply it to our Twitter data to remove such tweets. Exploring the impact of removing bot-generated tweets on the prediction performance of our models will be considered for further investigation.

Essay 3: Cryptocurrency Price Movement Prediction by Integrating Wavelets with Hybrid Deep Learning Systems: Evidence from a Large Dataset and the Effect of the COVID-19 Pandemic

Abstract

This study investigates the importance of employing high-quality features to enhance the forecasting performance of deep learning algorithms in cryptocurrency markets. We propose a two-stage prediction framework designed for forecasting the directional movement of cryptocurrency prices. In the first stage, a feature extraction method that integrates Discrete Wavelet Transform (DWT) and Long Short-Term Memory-Autoencoder (LSTM-AE) in parallel, is employed to generate high-quality features from historical prices and technical indicators. In the second stage, features derived from DWT and LSTM-AE are concatenated and the resulting feature set (AE-DWT) is fed into a Gated Recurrent Unit (GRU) to predict directional movement of cryptocurrency prices. Our experimental results on a dataset encompassing 25 cryptocurrencies unveil compelling findings. Our baseline models for comparison include the GRU model trained on raw features without any feature extraction applied. The AE-DWT feature set improves prediction performance of 23 out of the 25 cryptocurrencies. Both classification performances measured by accuracy and F1-score, are improved by around 15% and 18%, respectively, when compared to base cases in which no feature extraction is employed. We evaluate the predictive capability of the proposed method by subjecting it to testing during challenging periods, such as the COVID-19 pandemic. This time frame is marked by the heightened volatility of cryptocurrencies, which presents an obstacle for forecasting models to make accurate predictions. The results show that our approach is robust to highly volatile periods and can achieve comparable prediction results with those of non-crisis periods characterized by lower volatilities.

Keywords: COVID-19, Cryptocurrency, Deep Learning, Discrete Wavelet Transform (DWT), Feature extraction, LSTM-Autoencoder

1. Introduction

Trend prediction of time series intended as a classification task, is a challenging task especially when it comes to the highly speculative cryptocurrency markets. While empirical studies show that these markets are to some extent predictable with conventional technical analysis, high volatility and regular price jumps complicate the forecasting problem (Puzyrev, 2019). Cryptocurrency prices are influenced by several factors such as market demand and supply, market sentiment, macroeconomic factors, and more. This makes traditional linear models like Autoregressive Integrated Moving Average (ARIMA) struggle to capture the non-linearities. In contrast to traditional linear models, the artificial intelligence (AI) approach demonstrates a unique capability to apprehend the non-linear property of cryptocurrencies behavior by automatically extracting relevant features from raw data. In this regard, deep learning (DL) methods, a subset of AI, have been actively employed by researchers of this field (e.g., Cavalli & Amoretti, 2021; Lahmiri & Bekiros, 2019, 2021; Ortu et al., 2022a; Oyedele et al., 2023a; Shintate & Pichl, 2019)

Despite the fact that DL models are capable of extracting deep hidden patterns within raw data, there is substantial evidence that incorporating high-quality features extracted through alternative methods can elevate their predictive performance. In our previous works (Amirshahi & Lahmiri, 2023a, 2023b), we proved that using high-quality features can enhance the predictive capabilities of DL methods in cryptocurrency market prediction. In one study (Amirshahi & Lahmiri, 2023a), we showed that using features generated by GARCH-type models improves the prediction performance of Feed Forward Neural Networks (DFFNNs) and Long Short-Term Memory (LSTM) networks in predicting cryptocurrencies' volatility. Additionally, we explored the effectiveness of leveraging Twitter sentiments as input features for LSTM and Convolutional Neural Networks (CNN) in forecasting cryptocurrency prices in another work (Amirshahi & Lahmiri, 2023b).

Aiming to extend our earlier research efforts, our present study takes a similar approach for predicting cryptocurrency price movement direction using DL models, a task categorized as classification. To achieve this goal, we develop a hybrid model comprising of a feature extraction part along with a DL prediction model. Historical price data and technical indicators are fed as input data into the feature extraction module, where two parallel components collaborate to generate high-quality features: a DWT component which removes noise of the input data and an LSTM-Autoencoder model that generates deep features by encoding the input data. The extracted features are concatenated, and the resulting feature set is subsequently utilized as inputs for a GRU model as a type of DL model to predict next period's price movement.

Technical indicators are mathematical transformation of historical price data and have been widely used in similar studies (Alonso-Monsalve et al., 2020; Basher & Sadorsky, 2022; Nakano et al., 2018; Orte et al., 2023; Ortu et al., 2022a; Wu et al., 2022a). They can capture recurring patterns and provide insights into potential future price trends within the cryptocurrency markets. DWT has been employed as a denoising technique in numerous studies improving the accuracy of financial time series forecasting (Guo et al., 2021; G. Ji et al., 2022; Nobre & Neves, 2019; Sener & Demir, 2022; Wu et al., 2022b; Zeng & Khushi, 2020; Zhao & Khushi, 2020). Through the decomposition of the input feature vector into high and low-frequency components, this methodology eliminates abrupt fluctuations while retaining the overall trend – referred to as denoised features.

The LSTM-Autoencoder is a specialized form of the Autoencoder model that integrates LSTM layers within both its encoding and decoding components. Autoencoders have demonstrated their efficacy in extracting deep latent information from input data, which can be used as informative features in financial market prediction (e.g., Li et al., 2023; Wu et al., 2022a). Given the sequential nature of cryptocurrency data, the LSTM hidden layers will be embedded in the encoding and decoding structures, replacing the conventional fully connected layers found in a traditional Autoencoder setup. LSTM layers capture temporal dependencies effectively, a characteristic that is often overlooked by fully connected layers.

We chose GRU as our prediction model since it is a type of Recurrent Neural Networks (RNNs) and RNNs tend to outperform the others in time-series prediction (Goutte et al., 2023). It has several advantages over the popular LSTM model that motivated us to select it as our prediction model. For instance, it has a simple structure with fewer parameters (Li et al., 2023) making it less prone to overfitting. Moreover, the gating mechanism in GRU enables it to quickly adapt to changing patterns in the data which is crucial for accurate predictions.

It has been shown that during the COVID-19 pandemic period, cryptocurrencies became more volatile (Lahmiri & Bekiros, 2020b). Therefore, we assess the robustness of the proposed model in crisis periods like the COVID-19 pandemic. For this purpose, we compare the predictive performance of our proposed method in three distinct periods, namely, before, during, and after the COVID-19 pandemic. This enables us to gauge the proposed method's adaptability to varying market dynamics and its efficacy in capturing cryptocurrency trends during times of crisis. To assess the predictive performance of our method, we will use appropriate metrics such as accuracy and F1-score.

Our study offers several contributions that are summarized as follows:

i. We employ the Discrete Wavelet Transform (DWT) technique for data denoising in the context of cryptocurrency market trend prediction. To the best of our knowledge, there are few studies that have utilized DWT in the context of cryptocurrency market prediction. Studies like (Guo et al., 2021) and (Sener & Demir, 2022) have employed this technique for cryptocurrencies price prediction which is a regression task. Our innovation lies in adopting DWT for the challenge of classification tasks.

ii. We generate informative features by combining two sets of features extracted by DWT and LSTM-Autoencoder model. This novel strategy is expected to improve the classification performance of GRU as a type of DL model in predicting cryptocurrency market trends.

iii. We gauge the efficacy of our proposed approach in different challenging periods including before, during, and after COVID-19 pandemic characterized by varying degrees of cryptocurrency market volatility.

iv. Most relevant studies have focused on Bitcoin as it has the highest market capitalization (e.g., Adcock & Gradojevic, 2019; Cavalli & Amoretti, 2021; Z. Chen et al., 2020; Jaquart et al., 2021; Lahmiri & Bekiros, 2020a; M. Liu et al., 2021; Shintate & Pichl, 2019). While there are some prior studies that have explored prediction models across various cryptocurrencies, our research stands out in several key aspects. Firstly, we carefully curated a dataset of 25 cryptocurrencies, focusing on those with substantial market capitalization, covering over 75% of the market and have available data from 2018 to 2023. This extensive coverage ensures a broad representation of the cryptocurrency landscape. Additionally, unlike many studies that aggregate results, we provide detailed insights by reporting results for each individual cryptocurrency. This approach offers a clearer understanding of the predictive performance of machine learning models for each specific cryptocurrency, enhancing our insights into market dynamics and performance variations.

2. Literature Review

2.1. Deep learning and cryptocurrency price movement prediction

To assess the effectiveness of CNN model in predicting cryptocurrencies' price trend, Alonso-Monsalve et al. (2020) conducted a comparative study of four models, namely, CNN, Convolutional LSTM, MLP, and Radial Basis Function Neural Network (RBFNN). Considering six popular cryptocurrencies, the authors used 18 different technical indicators as inputs for MLP and RBFNN. For the inputs of the networks with convolutional component, time series data were encoded into two-dimensional image-like matrices. To create these matrices, thousands of input matrices were generated based on technical indicators (first dimension) and their lagged values (second dimension) where last column of each matrix was considered as the output class. For each cryptocurrency, the best configuration of all models was found through hyper-parameter optimization and finally, the evaluation of networks on test samples showed that the hybrid model of CNN-LSTM is superior to the other networks. Moreover, the performance of the CNN model was also outstanding particularly for prediction of Bitcoin, Ether, and Litecoin price movement. This study contributes to the literature by exploring the application of CNN in cryptocurrency price movement prediction.

Chen et al. (2020) evaluated a number of statistical and machine learning models for predicting Bitcoin price direction. They found that machine learning models excel when dealing with high-frequency data, whereas statistical models perform better on low-frequency data. They assumed that daily features might not have a significant impact on the prediction of price direction in 5-minute intervals. Therefore, in case of high-frequency data, they included only a limited number of input features and compensated for it by employing more complex models. On the other hand, for low-frequency data, the authors considered a wide range of input features and their results showed that the statistical methods such as Logistic Regression (LR) and Linear Discriminant Analysis (LDA) outperformed the machine learning counterparts. Among the five machine learning models examined, namely, Random Forest (RF), XGBoost, Quadratic Discriminant Analysis (QDA), Support Vector Machine (SVM), and LSTM, the LSTM achieved the highest accuracy of 67.2%.

In their study, Jaquart et al. (2021) analysed a comprehensive feature set, including technical, blockchain-based, sentiment-/interest-based, and asset-based features to predict short-term price movement of Bitcoin. They showed that technical features are the most relevant features for most prediction methods utilized in their study. Considering four prediction horizons, ranging from 1 to 60 minutes, they found that the predictability increases for longer prediction horizons. Despite using a wide range of feature sets, the achieved accuracies were slightly over 50%, indicating that the bitcoin market predictability is somewhat limited (Jaquart et al., 2021). All in all, recurrent neural networks and gradient boosting classifiers were found to be best-performing methods across all prediction horizons.

The study of Basher & Sadorsky (2022) is focused on predicting the direction of Bitcoin and gold price movements using tree-based methods. By analysing the variable importance using time series cross-validation techniques, they found that technical indicators are the most important features. It was observed that the variable importance does change over time, but the changes are not significant enough to alter the ranking of the variables. According to the results reported in the paper, random forests predict Bitcoin and gold price directions with a higher degree of accuracy than logit models.

In addition to technical indicators, social media has emerged as a rich source of information for predicting the direction of price movement in cryptocurrency markets. In this regard, Ortu et al. (2022b) utilized pre-trained Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) to extract users' sentiment data from top discussions on Reddit and developers' comments on GitHub. They combined sentiment data with technical and trading indicators to predict directional movement of Bitcoin and Ethereum prices. Their results for the hourly frequency showed that adding social media data enhances the predictive performance of deep learning models significantly. On the contrary, for the daily data, they observed that, in general, technical indicators alone perform best in the classification of next day's price movement.

Recently, Zhong et al. (2023) proposed an innovative end-to-end model for price trend prediction, leveraging Long Short-Term Memory (LSTM) and Relationwise Graph Attention Network (ReGAT). Their approach considers both individual cryptocurrency features and inter-cryptocurrency relationships. By constructing a cryptocurrency network based on shared features and employing LSTM to capture sequential patterns, the authors aim to profile the behavior of individual cryptocurrencies. The authors validate the effectiveness of LSTM-ReGAT using real-world cryptocurrency market data, demonstrating its potential in price trend prediction.

In another study (Zhou et al., 2023), the researchers forecast future price movements for individual cryptocurrencies across different time horizons, specifically the next day, three days later, and five days later. Their approach integrates various data sources, including historical trading data, Google Trends, and Twitter tweets, to enhance cryptocurrency price prediction. The study involves gathering cryptocurrency-related tweets from Twitter over a specified timeframe and employing the VADER algorithm to assess sentiment. Subsequently, the researchers develop three distinct sentiment indicators and utilize SVM algorithms to forecast cryptocurrency price movements based on the amalgamated data. They could achieve average F1 score and accuracy values of 0.62 and 0.65, respectively. Additionally, they introduce a novel portfolio optimization model that incorporates both cryptocurrency forecasting outcomes and the minimum variance portfolio model.

2.2. Application of DWT and Autoencoder in financial time series prediction

The application of Discrete Wavelet Transform in financial markets is mostly focussed on stock markets, and its potential in the cryptocurrency market remains relatively underexplored, offering opportunities for further investigation and research. In this regard, Nobre & Neves (2019) combined Principal Component Analysis (PCA) with DWT in order to improve the XGBoost binary classifier's performance. Their work involved applying the DWT method to a set of 31 features, including raw prices and technical indicators. Subsequently, the denoised features were fed into the XGBoost classifier to predict price movements in the future. Using a set of five different financial markets, they obtained an accuracy of more than 50% which indicates that Machine Learning models can perform better than random guessing in financial markets trading.

In a more recent study by Ji et al. (2022), the DWT technique was applied to raw prices and the improved technical indicators were calculated based on denoised raw prices. Using a two-stage feature selection method along with the improved technical indicators, the authors showed that increasing the quality of features can significantly enhance the performance of models in predicting price movements of four different stocks over a three-day horizon.

Wu et al. (2022a) combined DWT and feature extraction methods to improve prediction performance of an Extreme Learning Machine (ELM) in Chinese stock markets. The feature extractor in this study is the encoder part of an Autoencoder model, which is trained with feature penalty — a regularization based on the penalty term of the loss function (Wu et al., 2022a). The encoded features are then used as inputs for the ELM to predict stock price trends. The proposed method exhibits a remarkable improvement of approximately 5% on average compared to the authors' previous results.

The application of DWT in predicting prices within the Forex market can be seen in (Zeng & Khushi, 2020) and (Zhao & Khushi, 2020), both addressing regression problems. The authors of (Zeng & Khushi, 2020) utilized DWT to filter out the disturbance signals in the forex time series. An attention based RNN and ARIMA models are trained on the denoised data to forecast non-linear and linear components of the denoised data, respectively. The two forecasted results are then integrated to obtain final predictions. They concluded that denoising data is necessary since models of the same architecture achieved better accuracy when the close price series were denoised (Zeng & Khushi, 2020). Similarly, the authors of (Zhao & Khushi, 2020) used DWT to denoise raw Open, High, Low, and Close prices (OHLC) data and calculated the technical indicators based on denoised data. A history of technical indicators formed image matrices which were then utilized as inputs to the ResNet CNN (He et al., 2015) for feature extraction. The extracted features along with the original technical indicators were used as inputs to the LightGBM model (Ke et al., 2017) to predict rate of change in Forex prices. The researchers constructed seven different baseline models for comparison. All the baseline models were found to underperform the proposed model.

To the best of our knowledge, the Discrete Wavelet Transform (DWT) has not been utilized in predicting the movement direction of cryptocurrency prices, i.e., in the context of classification tasks. However, there have been a few studies that employed the DWT for regression problems, where the objective is to predict the actual price of cryptocurrencies rather than their movement direction. For example, Guo et al. (2021) proposed a new forecasting model, WT-CATCN, which combines Wavelet Transform (WT) and Casual Multi-Head Attention (CA) Temporal Convolutional Network (TCN) techniques to predict cryptocurrency prices. The model effectively captures important patterns in input sequences and models correlations among different data features. Testing with real Bitcoin trading data shows that WT-CATCN outperforms other state-of-the-art forecasting models, improving forecasting accuracy by 25% in terms of the RMSE metric. In another study (Parvini et al., 2022), the authors use Gold, Oil, S&P500, ETH, XRP, and BTC historical price time series, along with VIX and USDI time series, as potential predictors to investigate their predictive power in forecasting one-step-ahead BTC return. Initially, they apply DWT with a db3 filter to eliminate noise incorporated in the time series. Then, they establish a forecasting model based on LSTM neural networks and forecast one-step-ahead BTC returns using both univariate regression and multivariate regression approaches. Moreover, they distinguish between wavelet decomposed and non-decomposed predictors to study the effect of wavelet decomposition on forecasting accuracy and economic profitability of noisy and volatile BTC returns and provide potential explication for the outcomes.

In a recent study conducted by Li et al. (2023), the authors proposed a novel financial time series prediction method that combines Fully Connected Neural Network (FNN), CNN, GRU, AE and Attention Mechanism (AM). To effectively extract features and capture underlying patterns from financial time series data, a CNN-GRU block is used within the encoding-decoding structure of the AE model. In addition, multiple bridges are built between the decoder and encoder parts of the AE model allowing shallow features to be more directly utilized by the encoder. The attention mechanism is applied in this connection to emphasize the vital information and reduce the redundant one. Their results showed the superiority of their method over the other methods including the single models of CNN and GRU.

Our work is inspired by the studies reviewed in this section with an application in the cryptocurrency markets. A recurring theme among most of these reviewed studies involves the utilization of a feature extraction method, notably the DWT, for removing noise from data and using the extracted high-quality features to improve the overall prediction performance. In our study, we use DWT along with an LSTM-based Autoencoder to extract high-quality features that will improve prediction performance of a GRU model in predicting the directional movement of cryptocurrencies' daily prices.

Table 25: Recent studies on cryptocurrency price movement prediction

Reference	Market	Period / Frequency	Features	Methodology	Outcome
Alonso-Monsalve et al. (2020)	Bitcoin, Dash, Ether, Litecoin, Monero, and Ripple	July 1, 2018, to June 30, 2019 / 1-minute	18 technical indicators with long and short moving averages (5, 20, 30 and 60 min)	CNN, CNN-LSTM, MLP, RBFNN	The hybrid model of CNN-LSTM is superior to the other networks.
Chen et al. (2020)	Bitcoin	For daily prediction: February 2, 2017 to February 1, 2019 For 5-minute prediction: July 17, 2017, to January 17, 2018	Property and network (hash rate, block size, etc.), trading and market (market capitalization, transaction value, etc.), attention from the media and investors, gold spot price, OHLCV	LR, LDA, RF, XGBoost, QDA, SVM, LSTM	The LSTM achieved the highest accuracy of 67.2%.
Jaquart et al. (2021)	Bitcoin	March 4, 2019, to December 10, 2019 / 1-minute	Technical Indicators, asset-based features such as S&P500 and gold returns, blockchain-based and sentiment-/interest-based features	Feedforward, LSTM, GRU, XGBoost, RF, LR, a meta-model consisting of all individual models	RNNs and gradient boosting classifiers were found to be best-performing methods across all prediction horizons.
Basher & Sadorsky (2022)	Bitcoin and gold	September 2014, to December 2021 / daily	Technical indicators and macroeconomic variables (interest rates, economic policy uncertainty, market uncertainty, and inflation)	logit, bagging decision tree, RF	RF predicts Bitcoin and gold price directions with a higher degree of accuracy than logit models.
Ortu et al. (2022b)	Bitcoin and Ethereum	November 2018, to February 2021 / hourly & daily	Technical and trading indicators, sentiment and emotions extracted from GitHub and Reddit comments	MLP, CNN, LSTM, Attention LSTM, BERT for extracting the sentiment, emotion, VAD	Social media data improved the prediction performance of hourly data.

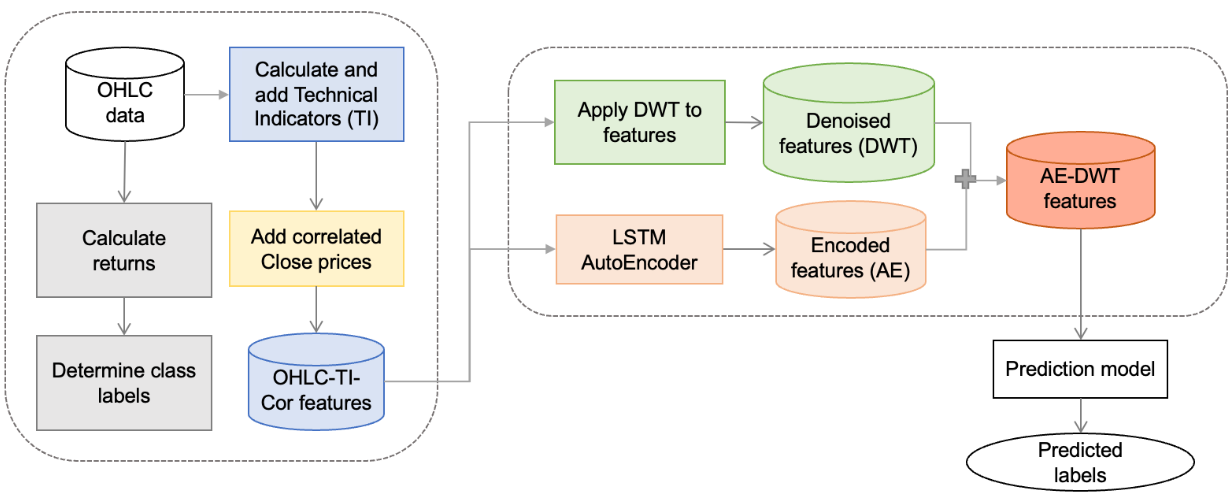
Zhong et al. (2023)	645 cryptocurrencies	March 30, 2020, to December 20, 2022 / daily	OHLCV, Technical indicators, Asset-based features, Google Trends Index	ReGAT/LSTM-ReGAT/ LSTM-GAT, baselines: LR, SVM, RF, DNN, LSTM, CNN	LSTM-ReGAT performs best in nearly all evaluation metrics.
Zhou et al. (2023)	Bitcoin, Ethereum, Ripple, Cardano, Dogecoin, Polkadot, and Litecoin	January 1, 2021, to June 30, 2021 (other data periods for Robustness test)	historical trading data, Google Trends, and tweets from Twitter	SVM for prediction, VADER algorithm to analyze the tweets for the sentiment.	The authors achieved average F1 score and accuracy values of 0.62 and 0.65, respectively.
Nobre & Neves (2019)	5 financial markets: Brent and Corn futures contract, Exxon Mobil Corporation stocks, Home Depot Inc. stocks, and S&P 500	February 25, 2003, to January 10, 2018 / daily	OHLCV + 26 technical indicators	PCA, DWT, XGBoost, GA for hyperparameter tuning of the XGBoost model	The combination of PCA and DWT improved the accuracies.
Ji et al. (2022)	SSE Composite Index, Hang Seng Index, Dow Jones Industrial Average, S&P 500	January 4, 2005, to March 8, 2021 / daily	18 technical indicators	DWT for denoising price data, random forest model as the classifier	Increasing the quality of features resulted in higher performance of models.
Wu et al. (2022a)	Dataset of 400 stocks in Chinese stock markets	January 1, 2001, to December 3, 2020 / daily	OHLCVA (open, high, low, close prices, trading volume, and trading amount) + 6 technical indicators	A hybrid framework using DWT, ELM, and AE with feature penalty	The authors achieved a substantial average accuracy improvement of 5% compared to their previous results.
Zeng & Khushi (2020)	Forex: USDJPY currency pair	January 01, 2019, to December 31, 2019 / 5-minutes	The raw close price series with 16 technical indicators	DWT, attention based RNN, and ARIMA	Denoising data is necessary since models of the same architecture achieved better accuracy when the close price series were denoised.
Zhao & Khushi (2020)	Forex: USDJPY currency pair	January 01, 2019, to June 10, 2020 / 5-minutes	OHLC + 26 technical indicators	DWT, ResNet and LightGBM (DRNL)	All the baseline models were found to underperform the proposed model.

3. Methodology

The proposed approach is schematically illustrated in Figure 15. The process begins by collecting raw price data including daily Open, High, Low, and Close prices (OHLC) and establishing class labels based on that. To enrich the initial feature set, technical indicators and correlated close prices of other cryptocurrencies are incorporated into the raw data, resulting in what is referred to as the OHLC-TI-Cor feature set. These features are then utilized as inputs for feature extraction module, where both the Discrete Wavelet Transform and LSTM-Autoencoder operate in parallel to generate final feature set denoted as AE-DWT. Subsequently, this informative feature set is fed into the prediction model to predict the directional movement of prices in cryptocurrency markets.

In this section, first, we review some background information on the technical indicators, feature extraction techniques including both Discrete Wavelet Transform and Autoencoder, the prediction model, which is a GRU model. Afterwards, we elaborate more on the implemented methodology.

Figure 15: Overall diagram of the proposed approach for cryptocurrency price movement prediction



3.1. Background

3.1.1. Technical Indicators (TI)

Technical Indicators related to historical data are the most frequently used features in cryptocurrency price trend prediction tasks (Zhong et al., 2023). They are statistical tools that are calculated based on market data such as open, high, low, close prices, and trading volume to assist traders and investors in making decisions regarding entering (buying) or existing (selling) the market. There are different categories of technical indicators each measuring a specific condition of the financial market. For example, trend-following indicators help identify the direction and strength of market trends. Oscillators indicate oversold or overbought conditions, while volatility indicators measure price fluctuations. No single technical indicator can independently determine the overall condition of the market. A combination of such indicators can provide a wider view and will be more useful in decision making. A detailed explanation of the employed technical indicator follows.

Simple Moving Average (SMA)

The Simple Moving Average (SMA) is the average of prices, usually close prices, in a specific period. It provides a smoothed version of the prices which aids in determining the overall trend direction of the price.

$$SMA_t = \frac{\sum_{i=t-n}^{i=t} P_i}{n} \tag{27}$$

- n: number of days in the specific period
- P_i: price on day i

Exponential Moving Average (EMA)

Exponential Moving Average (EMA) is an enhanced version of the SMA that assigns exponentially decreasing weights to the price values, giving more weight to recent prices and less weight to older prices. Therefore, EMA is more useful for short-term trading while the SMA is preferred for the long-term trading.

$$EMA_t = K(P_t - EMA_{t-1}) + EMA_{t-1} \tag{28}$$

- $K = \frac{2}{1+n}$: the weight given to the current price relative to the previous EMA value
- n: the number of days in the specific period
- P_t: the current price
- EMA_{t-1}: previous day's EMA or if calculating for the first time, SMA will be used instead

As can be seen in the Equation (28), the previous EMA is adjusted based on the difference between the current price and the previous EMA, gradually giving more weight to recent prices.

Relative Strength Index (RSI)

Relative Strength Index (RSI) measures the magnitude and velocity of price changes. In other words, it compares the size of recent gains to recent losses. It oscillates between the values of 0 and 100. When the RSI crosses above the 30 level from below, it is a signal for a potential buying opportunity or the end of a downtrend. On the other hand, when the RSI crosses below the 70 level from above, it indicates a potential selling opportunity or the end of an uptrend.

$$RSI_t = 100 - \frac{100}{1 + \left(\sum_{i=t-n}^{i=t} change_{i+} /n\right) \ / \left(\sum_{i=t-n}^{i=t} change_{i-} /n\right)} \tag{29}$$

- $change_{i+} = \left\{ \begin{matrix} (P_i - P_{i-1})\ if\ (P_i - P_{i-1}) > 0 \\ 0\ otherwise \end{matrix} \right\}$: the average upward change
- $change_{i+} = \left\{ \begin{matrix} Abs(P_i - P_{i-1})\ if\ (P_i - P_{i-1}) < 0 \\ 0\ otherwise \end{matrix} \right\}$: the average downward change

Bollinger Bands (baseline, upper, and lower bands)

Bollinger bands are a set of trendlines comprising of a simple moving average (typically a 20-day SMA) line as the base line and two other lines that are plotted as two standard deviations, both positively (upper band) and negatively (lower band), away from the SMA line. When the price line crosses the upper band, it may indicate a potential downward trend and on the other hand, when it crosses the lower band, it is an indication of upward trend.

$$UpperBand_t = SMA_t + 2 \times \sqrt{\frac{\sum_{i=t-n}^{i=t}(P_i - SMA_t)^2}{n}} \tag{30}$$

$$LowerBand_t = SMA_t - 2 \times \sqrt{\frac{\sum_{i=t-n}^{i=t}(P_i - SMA_t)^2}{n}} \tag{31}$$

Moving Average Convergence Divergence (MACD, MACD signal, MACD histogram)

The Moving Average Convergence Divergence (MACD) is calculated by subtracting the 26-period EMA from the 12-period EMA. The 9-day EMA of the MACD is called the MACD signal line. While crossing the signal line from above indicates a buying signal, crossing it from below is a sell signal. The difference between MACD and its signal line is usually represented by a histogram on the charts. When the histogram bars are shrinking (convergence of MACD and its signal line), it is an indication of reducing momentum which means the price trend is slowing down. On the other hand, when they are expanding (divergence of MACD and its signal line), it may indicate an increasing momentum which means acceleration of trend line.

$$MACD_t = 12period\ EMA_t - 26period\ EMA_t \tag{32}$$

Stochastic oscillators (K series & D series)

The Stochastic oscillator compares the close price of an asset to its range and generally consists of two lines: the *K* series which is the actual value and the *D* series which is an SMA of the *K* series. Their values are between 0 and 100 where the values close to 80 indicate the potential selling signal and the values close to 20 indicate the buying signal.

$$\%K_t = 100 \times \frac{Close_t - lowest\ Low\ price\ in\ the\ last\ n\ days}{highest\ High\ price\ in\ the\ last\ n\ days - \ lowest\ Low\ price\ in\ the\ last\ n\ days} \tag{33}$$

The usual values for the parameter n are 14 for the *%K* and 3 for the *%D* (Orte et al., 2023).

Williams %R

Similar to the stochastic oscillator, the Williams %R compares the close price of the asset to its high and low prices in a specific period (usually 14 days). Its value is in the range of -100 to 0 where −100 means the close price of today was the lowest low of the previous *n* days, and 0 means today's close was the highest high of the past *n* days.

$$\%R_t = -100 \times \frac{highest\ High\ price\ in\ the\ last\ n\ days - Close_t}{highest\ High\ price\ in\ the\ last\ n\ days - \ lowest\ Low\ price\ in\ the\ last\ n\ days} \tag{34}$$

3.1.2. Discrete Wavelet Transform (DWT)

Discrete Wavelet Transform (DWT) is a mathematical technique that decomposes a signal into two sets of coefficients: approximation (*cA*) and detail (*cD*). A wavelet is a wave-like oscillation that is localized in both time and frequency domains. This allows the wavelet transform to capture not only frequency information but also crucial time-domain details. When the DWT is applied to a time series data, the wavelet moves along the signal, analyzing different parts. The position parameter, denoted as *k,* indicates the starting point of the wavelet window as it slides along the time axis. For each *k*, the DWT computes the corresponding approximation and detail coefficients at that specific location. The scale parameter, denoted as *j,* controls the frequency resolution or the scale of the wavelet. A higher value of *j* corresponds to a smaller scale, leading to a narrower wavelet that captures higher frequencies. Conversely, a lower value of j corresponds to a larger scale, causing the wavelet to widen and capture lower frequencies.

The fundamental components of the DWT are the scaling wavelet basis (approximation part) and the wavelet basis (detail part) functions. These crucial components are derived as follows:

$$\phi_{j,k}(t) = 2^{\frac{j}{2}} . \phi(2^j t - k) \tag{35}$$

$$\psi_{j,k}(t) = 2^{\frac{j}{2}} . \psi(2^j t - k) \tag{36}$$

where:

- j is the scale parameter, and it is an integer that determines how many times the mother wavelet is dilated or compressed. The factor 2^j is the dilation factor, ensuring that the wavelet spans the appropriate range of frequencies at each scale. When j is increased by one, the dilation factor doubles, halving the width of the wavelet. This enables the wavelet to capture higher-frequency details in the signal.
- k is the position parameter, and it is an integer that determines the position of the wavelet along the time axis.
- $\phi\,(t)$ and $\psi\,(t)$ are the mother scaling wavelet and mother wavelet functions, respectively.

The choice of the mother functions in DWT depends on the specific wavelet family employed. In our study, we opt for the “db4” wavelet from the Daubechies (Daubechies, 1988) family that has been shown to perform well in denoising and feature extraction (e.g., Guo et al., 2021; Sener & Demir, 2022). The "4" in "db4" represents the order or the number of vanishing moments of the wavelet. For a wavelet with N vanishing moments, it can approximate polynomials of degree up to N-1. For a deeper understating of the mother functions, readers may refer to the original paper on the Daubechies wavelets by Daubechies (Daubechies, 1988).

Approximation and detail coefficients can be calculated by convolving the time series with $\phi\,(t)$ and $\psi\,(t)$ as below:

$$cA_{j,k} = \sum_{t=0}^{N-1} X(t). \phi_{j,k}(t) \tag{37}$$

$$cD_{j,k} = \sum_{t=0}^{N-1} X(t).\psi_{j,k}(t) \tag{38}$$

where:

- $X(t)$ is the input time series.
- N is the length of the X_t.
- $cA_{j,k}$ represents the approximation coefficient at level *j* and position *k*.
- $cD_{j,k}$ represents the detail coefficient at level *j* and position *k*.

To represent a time series at various levels of resolutions, the formulas can be applied recursively for multiple levels to obtain multiple sets of approximation and detail coefficients. The implementation details will be discussed in Section 3.2.

3.1.3. Long Short-Term Memory-Autoencoder (LSTM-AE)

An Autoencoder (AE) is a type of unsupervised deep learning model designed to predict its own input data, X_t, as output. The AE model consists of two main components: the encoder and the decoder, both can have one or multiple hidden layers with the same configuration. There is a bottleneck layer between the two components to extract abstract features that can be used in down-stream supervised tasks, such as cryptocurrency price movement prediction in our case.

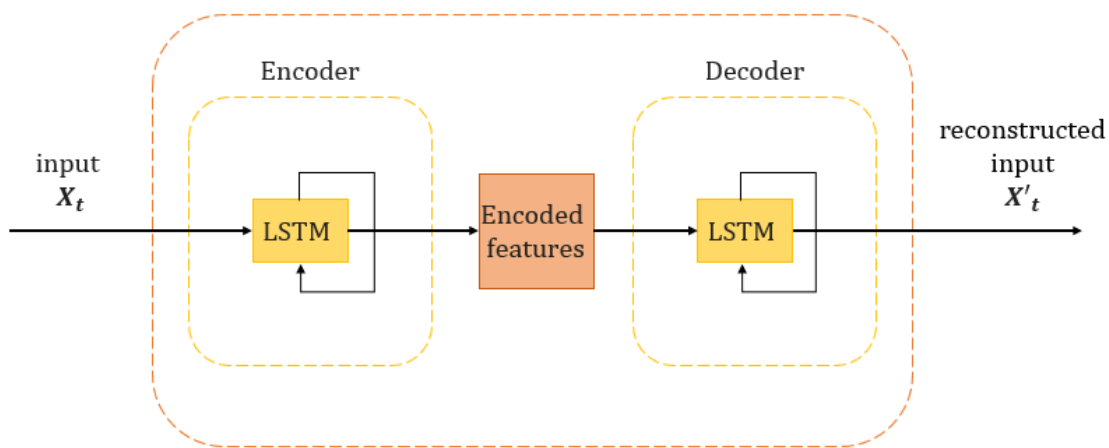
The LSTM-AE combines the concepts of autoencoders and LSTMs. Unlike traditional autoencoders that rely on fully connected layers for their encoder and decoder components, the LSTM-AE is a better choice for extracting temporal dependencies and patterns that are inherent in time-series data. Consequently, they can extract meaningful features that take into account the sequential order and historical context, leading to more effective representations for down-stream tasks like the price movement prediction.

As Figure 16 shows, first, the input data with length of h, with each time step having *M* features, passes through the encoder. At each time step *t*, the input vector X_t is fed into the LSTM cell where the hidden state h_t and cell state C_t are generated for the current time step. The number of units in the last layer of the LSTM encoder is usually less than the length of input sequence. This forces the network to reduce information, reducing noise and enabling it to approximate the original data rather than copying it end-to-end. The final encoded hidden state, which is the output of the last encoder LSTM layer, is a compressed representation of the input sequence which is referred as new set of data (features).

The LSTM decoder part is responsible for reconstructing the input data by taking both C_t and h_t from the encoder which are served as the decoder's initial cell and hidden states. At each time step, the decoder predicts the next value in the sequence based on the current hidden and cell states while the ground truth values from the original input sequence are fed as inputs. These predicted values are stacked to generate the reconstructed sequence (X'_t).

The LSTM-AE model is trained on the input data by minimizing the reconstruction loss, which is the difference between the original input data and its reconstructed version, i.e., $L(X_t, X'_t)$. The loss function can be Mean Squared Error (MSE) or Mean Absolute Error (MAE). By minimizing the reconstruction loss, the LSTM-AE model focuses on the most relevant information in the original input sequence. Once the training is completed, the encoder part can be employed to generate a representation of the original input sequence, containing essential and relevant information that can be used as new features for down-stream supervised tasks.

Figure 16: LSTM-Autoencoder structure



3.1.4. Gated Recurrent Unit (GRU)

A Gated Recurrent Unit (GRU) is a variant of Recurrent Neural Networks (RNNs) introduced by Cho et al. (2014). The key idea behind GRU is to use gating mechanism to control how much information from previous time steps should be retained at each time step of the input sequence. Figure 17 illustrates the different components of a typical GRU which are explained below.

At each time step *t,* the GRU receives an input vector X_t and produces a hidden representation or hidden state vector h_t, which serves as the network's memory or what has been learned up to time *t*.

Update gate: The update gate determines how much information from the previous hidden state (h_{t-1}) should be carried over to the current hidden state (h_t) (Cho et al., 2014). This helps the GRU learn long-term dependencies in the sequence. Mathematically, the current input vector (X_t) and the information from previous time step (h_{t-1}) are multiplied by their own weight matrices, *W(z)* and *U(z)*, respectively, which are learned through model training. The results of the computations are combined, and then a sigmoid function (σ) is applied to the combined value, resulting in a value between 0 and 1.

$$z_t = \sigma(h_{t-1}.U_z + X_t.W_z) \tag{39}$$

Reset gate: The reset gate controls which parts of the previous hidden state (h_{t-1}) should be forgotten. It helps the GRU to capture short-term dependencies in the sequence. The equation is similar to the update gate with different weight matrices.

$$r_t = \sigma(h_{t-1}.U_r + X_t.W_r) \tag{40}$$

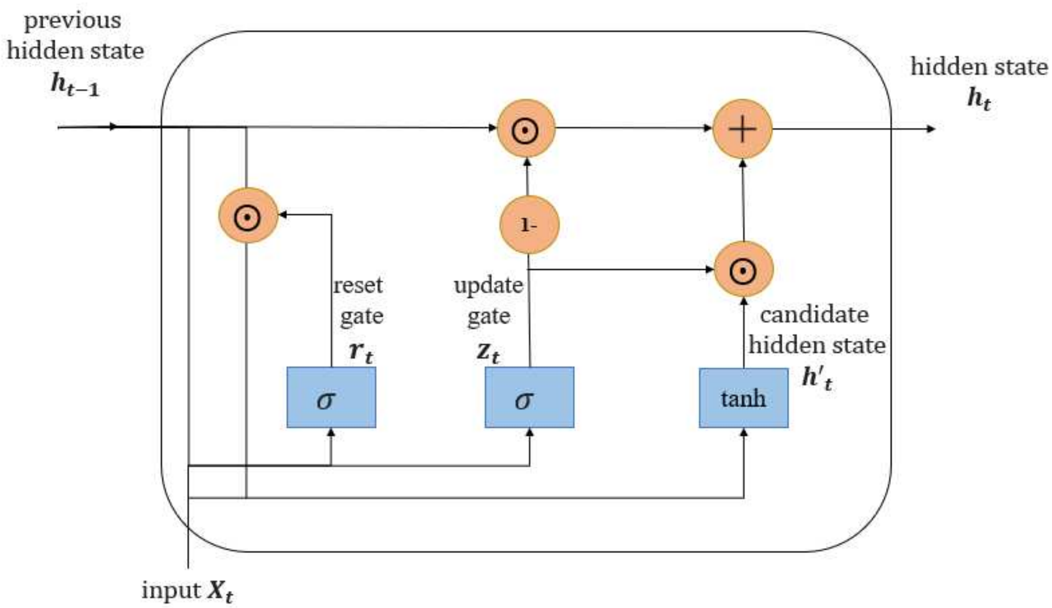
Candidate hidden state: The element-wise product of the reset gate and the previous hidden state vector is calculated and then, it is multiplied by its own weight parameters. The result is added to the parameterized current input vector. The hyperbolic tangent activation function is applied to the result and generates the candidate hidden state (h'_t) with a value between -1 and 1. When the reset gate is close to 0, the hidden state is forced to ignore the previous hidden state and reset with the current input only (Cho et al., 2014).

$$h'_t = tanh\,(X_t.W_h + (r_t \odot h_{t-1}).U_h) \tag{41}$$

Hidden state: Finally, the information from the update gate (z_t) is combined with the candidate hidden state (h'_t) to generate the final hidden state (h_t). A value of z_t close to 1 means retaining the old state (h_{t-1}) and ignoring the information from current step. A value of z_t close to 1 means that the current hidden state approaches the candidate hidden state (h'_t).

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \tag{42}$$

Figure 17: GRU unit structure



3.2. Implementation details

3.2.1. Feature extraction and class labeling

Open, High, Low, and Close prices (OHLC)

The daily prices of 25 most traded cryptocurrencies, which have data available for a period of five years, were collected from the website "www.investing.com". The datasets include the Open, High, Low, and Close prices, spanning from February 2018 to February 2023. The cryptocurrencies consist of Bitcoin, XRP (Ripple), Monero, Ethereum, Ethereum Classic, Litecoin, Zcash, Stellar, Dash, IOTA, EOS, OMG Network, Bitcoin Cash, Waves, Neo, Binance Coin, TRON, Cardano, Maker, Filecoin, Decentraland, Chainlink, Tezos, Enjin Coin, and Theta. Collectively, as of August 2023, these cryptocurrencies have a market cap of approximately $975 billion, accounting for 75% of the total market capitalization of all cryptocurrencies, which amounts to $1.3 trillion.

Class labeling

The binary classification problem of this study is defined as follows:

$$Class_t = \begin{cases} 1 & if\ r_t > 0 \\ 0 & if\ r_t \leq 0 \end{cases} \tag{43}$$

where r_t is the log return of the Close price data and it is calculated using the formula $r_t = \log\left(\frac{Close_t}{Close_{t-1}}\right)$. When $Class_t = 1$, it indicates a positive return value for the current day, implying that today's Close price has increased compared to the previous day's Close price. Conversely, when $Class_t = 0$, it means a negative return value for the current day, implying that today's Close price has decreased compared to the previous day's Close price. The objective is to predict the direction of the next day's price movement, i.e., $y_t = Class_{t+1}$ given a sequence of input feature sets, including the past h days, denoted as $(X_{t-h}, \ldots, X_t)$. Table 26 presents the distribution of classes for each cryptocurrency across the entire dataset. As can be seen in the table, the distribution is slightly unbalanced.

Table 26: Distributions of binary class labels across the entire dataset

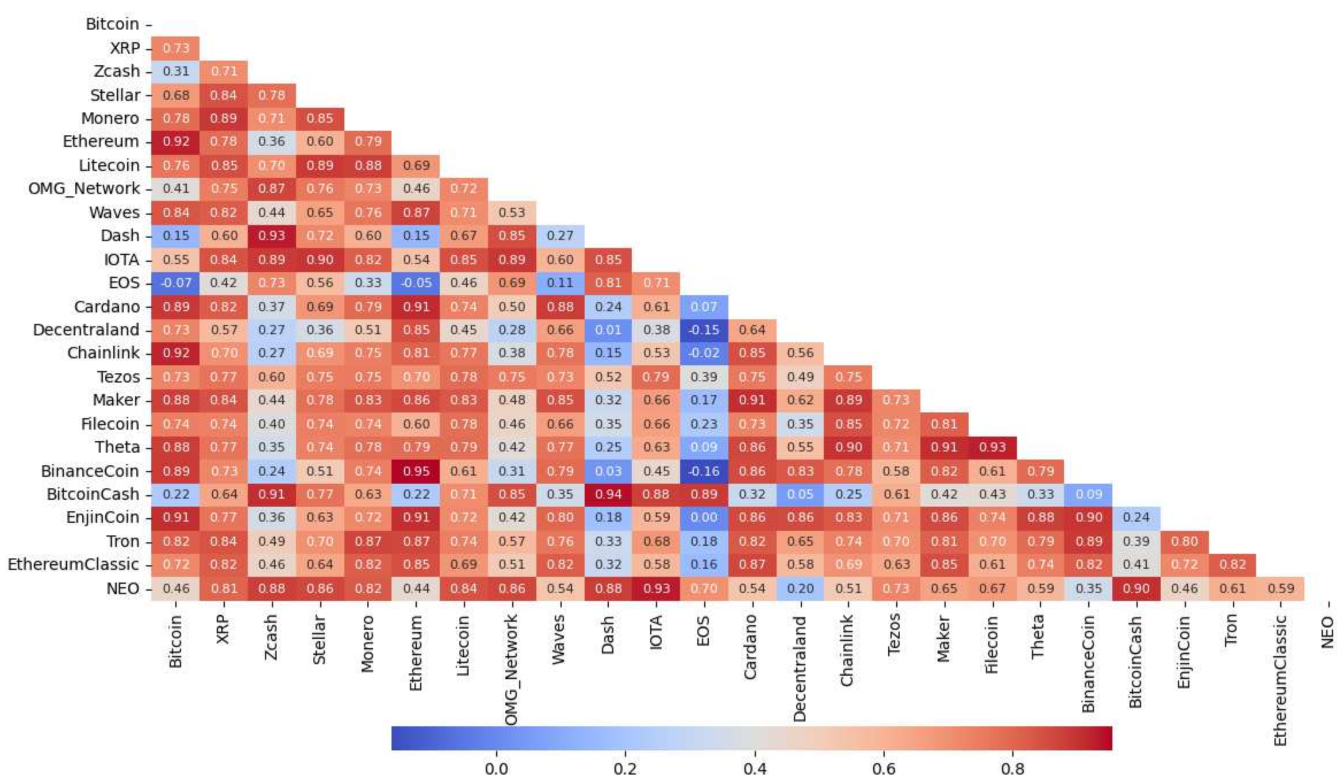
Cryptocurrency	**%Class 0**	**%Class 1**
Binance Coin	48.36	51.64
Bitcoin	48.96	51.04
Bitcoin Cash	50.88	49.12
Cardano	50.66	49.34
Chainlink	51.21	48.79
Dash	49.95	50.05
Decentraland	50.11	49.89
EOS	51.1	48.9
Enjin Coin	53.12	46.88
Ethereum	48.85	51.15
Ethereum Classic	49.62	50.38
Filecoin	53.56	46.44
IOTA	49.34	50.66
Litecoin	50.11	49.89
Maker	51.59	48.41
Monero	47.75	52.25
NEO	48.9	51.1
OMG Network	50.77	49.23
Stellar	49.95	50.05
Tezos	50.88	49.12
Theta	48.96	51.04
TRON	48.25	51.75
Waves	50.71	49.29

XRP	50	50
Zcash	50.82	49.18

Correlated Close prices (Cor)

The trend of one cryptocurrency is not only determined by its own dynamics but also closely correlated with other cryptocurrencies (Sun et al., 2020b). Figure 18 presents the correlation heatmap of close prices among all 25 cryptocurrencies analyzed in this study. For each cryptocurrency dataset, we added the Close price data of the other cryptocurrencies that have correlation values of higher than 0.85 with that cryptocurrency. The choice of 0.85 was because we tested different values such as 0.9 and 0.8, and the threshold value of 0.85 was found to add a fair number of new features to each cryptocurrency's dataset.

Figure 18: Correlation heatmap of cryptocurrencies' close prices



Technical Indicators (TI)

To generate TI values for all 25 cryptocurrencies, we utilized the stockstats Python library [4]. In line with similar studies (Nakano et al., 2018; Ortu et al., 2022a; Wu et al., 2022a), a list of commonly used technical indicators (See Table 27) were extracted and by considering various components of each technical indicator and employing different values for the window size parameter, a total of 25 technical indicators were generated as TI feature set.

Table 27: Technical indicators and their associated time window size

Technical Indicator	**Type**	**Time window size (n)**	**Brief explanation**
Simple Moving Average (SMA)	Trend-following	5-day, 10-day, 20-day,	Calculated based on close price
Exponential Moving Average (EMA)	Trend-following	5-day, 10-day, 20-day	Calculated based on close price
Relative Strength Index (RSI)	Momentum oscillator	Window = 14	Compares the size of recent gains to recent losses
Bollinger Bands (baseline, upper, and lower bands)	Volatility indicator	Window of boll = 15, 20	
Moving Average Convergence Divergence (MACD, MACD signal, MACD histogram)	Trend-following and momentum indicator		The MACD line is calculated by subtracting the 26-period EMA from the 12-period EMA
Stochastic oscillators (K series & D series)	Momentum oscillator	Window = 5, 9, 14	A momentum indicator that compares the close price to the high-low range over a specific period
Williams %R	Momentum oscillator	Period of 14 & 6	A momentum indicator that compares the close price to the high-low range over a specific period

After conducting correlation analysis for the technical indicator features across all cryptocurrencies, we realized that SMA, EMA, and Bollinger Bands exhibit correlation due to their shared reliance on historical price data and moving average calculations (Please see Appendix C1). Despite their similarities, these indicators may capture different aspects of price behavior and

[4] https://pypi.org/project/stockstats/

provide complementary information. By including all three indicators in our feature set, we can capture a broader range of market dynamics and enhance the robustness of our analysis.

Discrete Wavelet Transform (DWT)

To represent a time series at various levels of resolutions, the formulas can be applied recursively for multiple levels to obtain multiple sets of approximation and detail coefficients. In line with similar studies (Nobre & Neves, 2019; Sener & Demir, 2022), we set this parameter to 5 in our experiments.

Our primary aim in employing the DWT is to remove noise from features, and a commonly used technique for that is soft thresholding (Donoho et al., 1995). Soft thresholding or in general, shrinking methods reduce the magnitude of larger coefficients while zeroing out small coefficients, thereby preserving important information. To achieve this, we applied the universal threshold value, a data-adaptive threshold derived through Equation (44):

$$threshold = \sigma . \sqrt{2 . \log (N)} \tag{44}$$

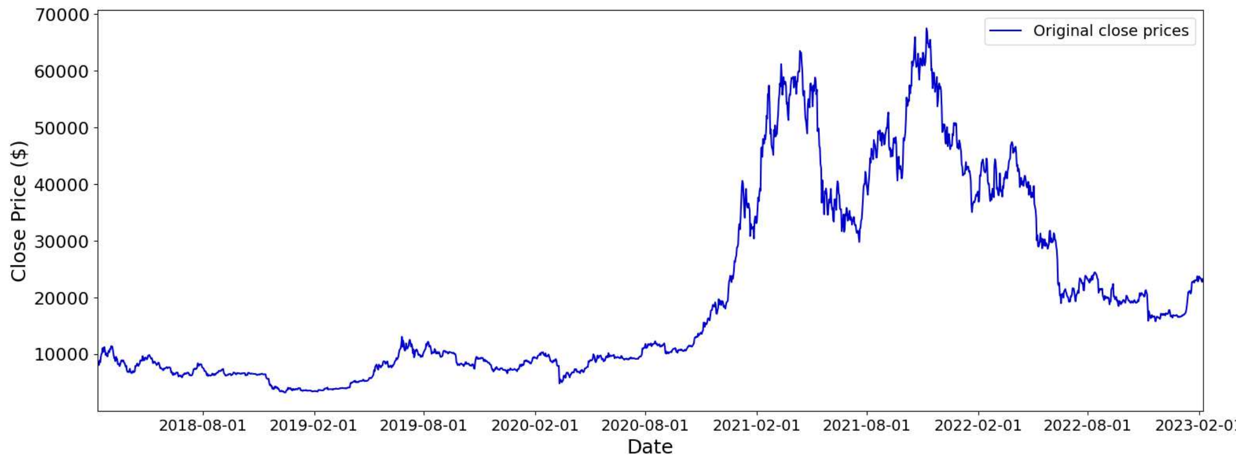
where:

- σ is the estimated standard deviation of the noise in the signal.
- N is the length of the input time series data.

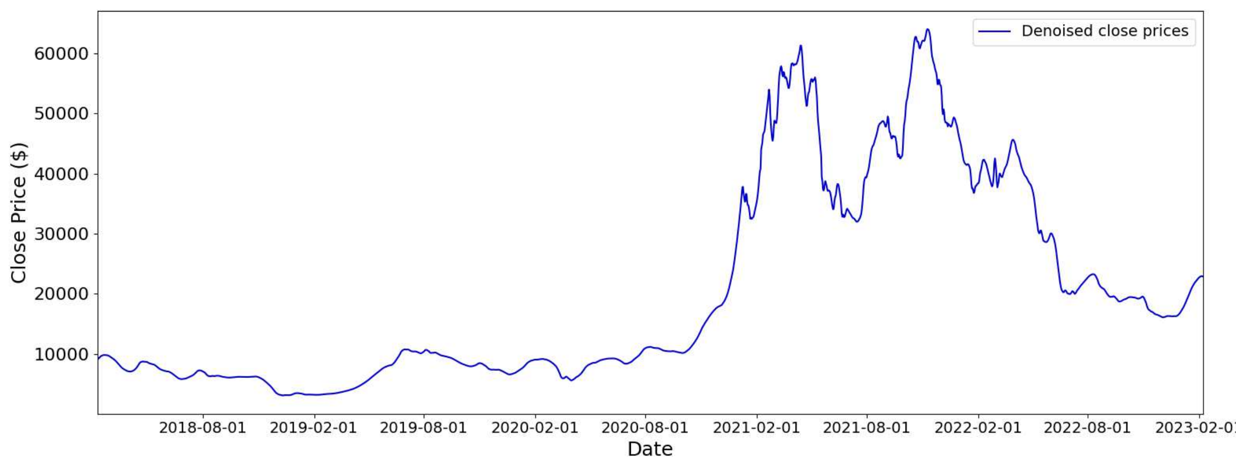
The universal threshold is proportional to the standard deviation of the noise and increases with the logarithm of the data length. As the length of the data increases, the universal threshold becomes larger, allowing for more aggressive noise reduction. We employ PyWavelets python library[5] and apply DWT for denoising all features, including OHLC data and Technical Indicators. After extracting the detail and approximation coefficients through wavelet decomposition, the denoised series is reconstructed by first applying thresholding to the detail coefficients to suppress noise, while leaving the approximation coefficients unchanged. This modified set of coefficients is then used in the inverse wavelet transform process, which combines the approximation and detail coefficients to reconstruct the denoised series. Figure 19 shows the original and denoised Close prices of Bitcoin over time, respectively.

[5] https://pywavelets.readthedocs.io/en/latest/

Figure 19: a) Original, b) Denoised Close prices over time for Bitcoin's data as an example



(a)



(b)

3.2.2. Feature scaling

Following the similar works (Li et al., 2023; Nobre & Neves, 2019; Zeng & Khushi, 2020), we will rescale every feature in the data sets to the range of [0, 1] using the Min-Max normalization technique presented in Equation (45).

$$\tilde{X}_t = \frac{X_t - X_{min}}{X_{max} - X_{min}} \tag{45}$$

where $\tilde{X}_t \in [0, 1]$, X_{min}, and X_{max} are the standardized, minimum, and maximum values of the feature X_t, respectively. It is worth mentioning that our feature scaling process occurs after data splitting. In other words, the data is initially divided into training and testing sets. Subsequently, feature normalization is executed within each partition. This approach ensures that the calculation of the minimum and maximum values for the formula does not involve the use of unknown future data.

3.2.3. Hyper-parameter tuning and performance metrics

For hyper-parameter tuning of our models, we utilize the time-series Cross-Validation (tsCV) method, similar to the work of Basher & Sadorsky (2022). Specifically, we employ the Rolling-Window CV, illustrated in

Figure 20 Figure 20. In this method, the training set expands over time while the validation set moves forward in time. This ensures that only prior data points are used to forecast future data points, maintaining the temporal order of time series data. In the regular CV, data is randomly split into multiple folds as the order of data is not important.

The most important hyper-parameters of the AE and GRU models were selected to be optimized using the grid search method. This method involves defining a grid of possible values for each hyper-parameter and exploring all combinations of these values during the model training process. A grid of hyper-parameters was formed which is available in Table 28. For each set of hyper-parameter values, the model will be trained on a training set and evaluated on the corresponding validation set. The hyper-parameter set that achieves the best performance, as measured by the average forecast accuracy, is selected as the optimized set of hyper-parameters.

Figure 20: Schematic of time-series Cross-Validation (tsCV) method

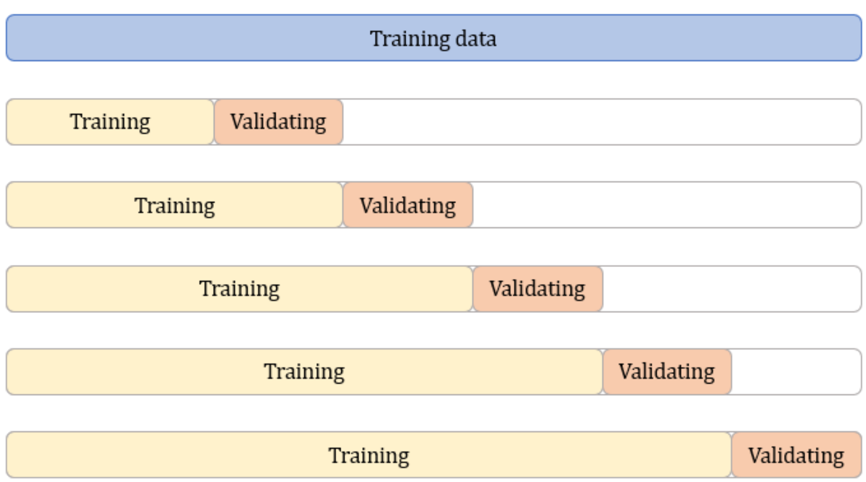


Table 28: The grid of hyper-parameters with possible values

```
Hyperparameters_ grid = {
    'gru_units': [[32], [64, 32], [64, 32, 16]],
    'gru_activation': ['tanh', 'relu'],
    'gru_dropout': [0.0, 0.2],
    'gru_learning_rate': [0.001, 0.01],
    'lstm_ae_units': [ [24, 8], [24, 16], [16] ]}
```

Regarding the evaluation metrics, since we have slightly unbalanced distribution of classes (See Table 26), prediction models tend to classify all samples as the majority class, disregarding the minority class. Consequently, the model's accuracy may appear high because it performs well in classifying the majority class, while its performance in classifying the minority class is overlooked. To address this issue, a more comprehensive evaluation measure is required. The F1-score considers the accuracy of predictions for both classes, considering the model's performance on both the majority and minority classes. By doing so, the F1-score provides a balanced assessment of the model's predictive capability for all classes, making it a more reliable performance metric. The accuracy and F1 metrics are defined as follows:

$$Accuracy = \frac{Number\ of\ correct\ predicted\ labels}{Total\ number\ of\ predicted\ labels} \times 100 \tag{46}$$

$$F_1 = \frac{2.precision.recall}{precision + recall} \tag{47}$$

where precision is the proportion of true positive predictions out of all positive predictions, and recall is the proportion of true positive predictions out of all actual positive instances in the dataset. As shown in Equation (47), the F1-score makes a balance between precision and recall. Values of F1 close to 1 means that the model has achieved high values for both precision and recall, which indicates that it has made fewer false positive and false negative predictions.

4. Experiments and results

4.1. Experimental setup

Our primary goal is to assess the prediction performance of the GRU model, a type of deep learning model, in forecasting directional movement of cryptocurrency markets using the following sets of features as its inputs:

1. Open, High, Low, Close prices (OHLC).
2. A combination of OHLC, technical indicators, and correlated close prices (OHLC-TI-Cor).
3. Denoised features (DWT applied to the OHLC-TI-Cor feature set).
4. Features encoded by the LSTM-AE model when it is fed by the OHLC-TI-Cor features (AE).
5. A combination of denoised and encoded features (DWT-AE).

Both the LSTM in the feature extraction part and the GRU fall under the category of sequence models. These models need a sequence of input data to generate corresponding output data. More precisely, a sequence of previous h days of each feature set $(X_{t-h}, \dots, X_t)$ is fed into the GRU model to predict whether the next day's close price (C_{t+1}) goes up or down. Similarly, for the LSTM-AE, a sequence of previous h days of OHLC-TI-Cor feature set is fed into the model to extract new feature sets (i.e., AE-DWT). During our exploration, we experimented with different input sequence lengths of 7, 14, and 21 previous days. We found that using a history of 14 previous days slightly outperformed the other options of 7 or 21 days. Consequently, each training data point is represented as a 2-dimensional matrix sized $h \times M$, where h is the lookback period, and M is the number of features, varying from one feature set to another. For more details, please refer to Table 29.

Table 30 presents the details of our experiments. In the first experiment, we employed the entire 5-year dataset, splitting it into training (80%) and testing (20%) sets. For each feature case, we employed a 5-fold tsCV strategy along with the grid search method and selected the most suitable hyper-parameters. The hyper-parameters grid is presented in Table 28. For the LSTM-AE feature extractor, we focused on tuning the number of hidden layers and hidden units. For example, [24, 8] indicates that the LSTM models in the encoder and decoder parts of the AE have 24 units each, and the bottleneck layer is composed of 8 units. As a result, 8 new features will be extracted from the input features. It is worth mentioning that the AE was not tuned independently from the

GRU model. In other words, hidden units parameter of the LSTM-AE was optimized jointly with the GRU's hyper-parameters. The remaining parameters of the AE model were held fixed as follows: a batch size of 32, 50 epochs for training, "MSE" as the loss function, "ReLU" as the activation function for hidden layers, and "linear" activation function for the output layer since the output represents continuous values (X_t'). Additionally, we employed the "Adam" optimizer with a fixed learning rate of 0.001.

We conducted hyper-parameter tuning for the GRU model, optimizing the following parameters: hidden units, activation functions for hidden layers, dropout rate, and learning rate. On the other hand, certain parameters were kept fixed: batch size = 32, number of epochs = 100, loss function = binary cross-entropy, suitable for binary classification problems, activation function of the last layer: Sigmoid, given the binary nature of the classification task, and optimizer = "Adam". To prevent overfitting, we employed the following strategies:

- **Early Stopping**: The Early Stopping callback was used to monitor the validation loss during training. If the validation loss did not improve for a specified number of epochs (patience), training was stopped early, preventing the model from overfitting to the training data.
- **Model Checkpoint**: The Model Checkpoint callback was employed to save the best-performing model based on validation loss. This ensured that only the model with the lowest validation loss is saved, helping to prevent overfitting by retaining the model that generalizes well to unseen data.
- **Dropout**: Dropout regularization was implemented in the GRU layers via the dropout parameter. Dropout randomly deactivates a fraction of neurons during training, forcing the model to learn more robust and generalized representations of the data and reducing the risk of overfitting.

The tuned models were evaluated on the out-of-sample test set. Due to the stochastic nature of Neural Networks, we conducted ten evaluations of each model using ten distinct random seeds, as suggested by prior studies (Jaquart et al., 2021; Nobre & Neves, 2019; Wu et al., 2022a; Zhong et al., 2023). This approach allowed us to mitigate the impact of randomness and ensured the reliability and robustness of our findings. The average and standard deviations of both accuracies and F1-scores were computed based on ten repeated runs, and the results were reported in Table 32 and Table 33. The results will be discussed in the next following sections.

Table 29: Details of the feature sets

Features	$X_t = \begin{pmatrix} X_t^{(1)} & \cdots & X_t^{(M)} \\ \vdots & \ddots & \vdots \\ X_{t-14}^{(1)} & \cdots & X_{t-14}^{(M)} \end{pmatrix}$
OHLC	Open, High, Low, Close prices ($M = 4$)
OHLC-TI-Cor	Open, High, Low, Close prices Simple Moving Averages: close_5_sma, close_10_sma, close_20_sma Exponential Moving Average: close_5_ema, close_10_ema, close_20_ema Relative Strength Index: rsi_14 Bollinger Bands: boll, boll_ub, boll_lb, boll_15, boll_ub_15, boll_lb_15 Moving Average Convergence Divergence: macd, macds, macdh Stochastic oscillators: kdjk_5, kdjd_5, kdjk_9, kdjd_9, kdjk_14, kdjd_14 Williams %R: wr_14, wr_6 correlated close prices. ($M \geq 28$)
DWT	Discrete Wavelet Transform of all OHLC-TI-Cor features ($M \geq 28$)
AE	Features encoded by the LSTM-AE model when it is fed by the OHLC-TI-Cor features (depending on the size of the encoder layer, ($M = 8$ or 16)
DWT-AE	A combination of DWT and AE features ($M \geq (28 + 8)$ or $M \geq (28 + 16)$)

Table 30: The details of the experiments in price movement prediction study

Period	**Period dates**	**Testing strategy**
Experiment 1:		
Whole data	5 years: 2018-02-08 to 2023-02-07	80% for training (5-fold tsCV for hyper-parameter tuning), 20% for testing (out-of-sample)
Experiment 2:		
Before COVID-19 pandemic	2 years: 2018-02-08 to 2020-02-07	80% for training (5-fold tsCV for hyper-parameter tuning), 20% for testing
During COVID-19 pandemic	2 years: 2020-02-08 to 2022-02-07	80% for training (5-fold tsCV for hyper-parameter tuning), 20% for testing
After COVID-19 pandemic	1 year: 2022-02-08 to 2023-02-07	80% for training (5-fold tsCV for hyper-parameter tuning), 20% for testing

In the second experiment, our aim was to gauge the efficacy of our proposed approach during challenging periods such as the COVID-19 pandemic. For this purpose, each dataset was divided into three periods:

1. The first period spans two years from February 2018 to February 2020, representing the time before the COVID-19 pandemic.
2. The second period covers the subsequent two years, from February 2020 to February 2022, representing the period during the COVID-19 pandemic.
3. The final period consists of the last year, from February 2022 to February 2023, representing the time after the COVID-19 pandemic.

The choice of February 2020 as the splitting month was based on the fact that the beginning of the COVID-19 pandemic in majority of the countries was between January 2020 and March 2020. Although the end date of the COVID-19 pandemic was officially declared by the head of the UN World Health Organization (WHO) on May 4, 2023 [6], we chose February 2022 as the splitting point between the two periods of during and after the pandemic in our study. This decision was based on the overall trend of daily new death cases attributed to COVID-19 which seems to start a downward trend in February 2022. Furthermore, when comparing this trend with the (Close) price trend of cryptocurrencies, we observed that the most volatile period for the cryptocurrencies aligns with the period from February 2020 to February 2022, during which the highest number of deaths were reported. This correlation is evident in Figure 21 and Figure 22 which present the trends for the two most popular cryptocurrencies, Bitcoin and Ethereum. The highest peaks of both daily death cases and the close prices of cryptocurrencies are happening between February 2020 and February 2022.

[6] https://www.who.int/news/item/05-05-2023-statement-on-the-fifteenth-meeting-of-the-international-health-regulations-(2005)-emergency-committee-regarding-the-coronavirus-disease-(covid-19)-pandemic

Figure 21: Bitcoin Close prices and COVID-19 daily new death cases over the period of the study

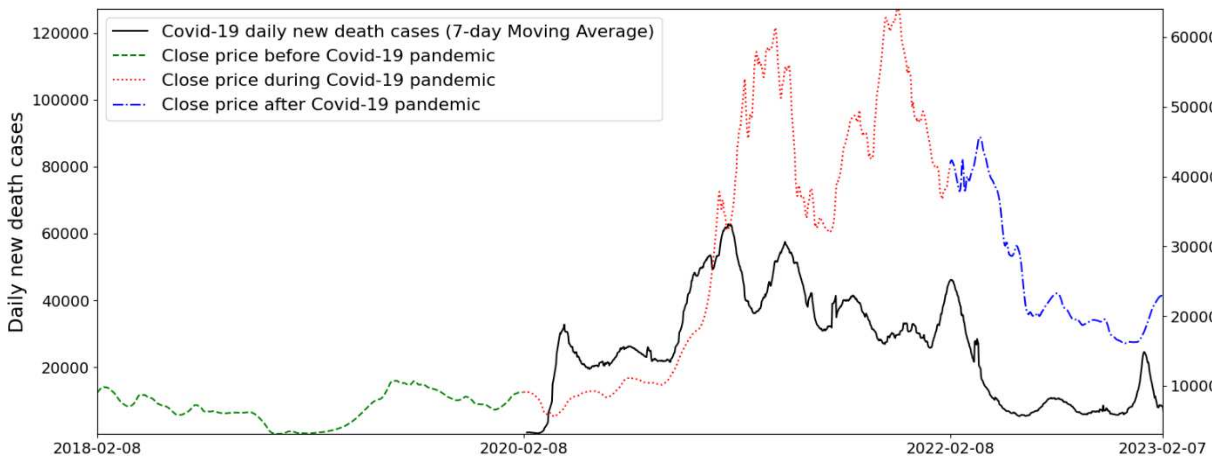


Figure 22: Ethereum Close prices and COVID-19 daily new death cases over the period of the study

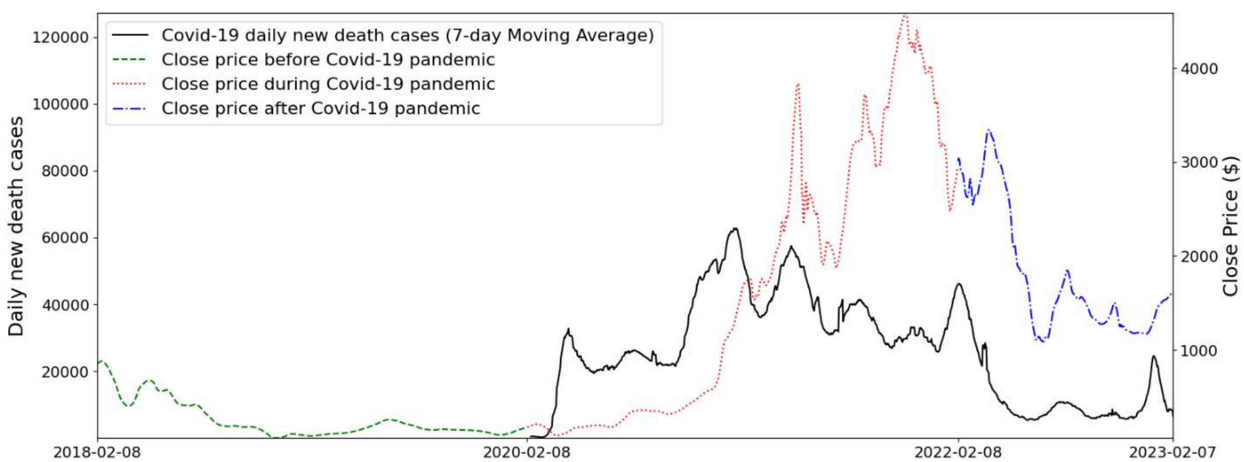


Table 31 presents the distribution of classes across three studied periods for all 25 cryptocurrencies. As can be seen in the table, a slight class imbalance is observed in the three sub-periods. Prior to the COVID-19 outbreak, the majority of class labels are zero, indicating a prevalence of downward trends or negative returns. Conversely, during the COVID-19 pandemic, there is a higher proportion of data labelled as 1, implying more positive returns. The distribution of classes appears to be more evenly distributed in the period following the pandemic.

The three periods show varying degrees of cryptocurrency volatility, as depicted in Figure 21 and Figure 22. Therefore, it is crucial to conduct experiments to compare our model's predictive capabilities across these distinct time frames. The prediction results obtained on each period’s testing sample were reported in Table 34. The results will be discussed in the following sections.

Table 31: Distributions of binary class labels across three sub-periods

Cryptocurrency	%Class 0 - before pandemic	%Class 1- before pandemic	%Class 0 - during pandemic	%Class 1- during pandemic	%Class 0 - after pandemic	%Class 1- after pandemic
Binance Coin	49.32	50.68	45.83	54.17	51.52	48.48
Bitcoin	49.18	50.82	46.37	53.63	53.72	46.28
Bitcoin Cash	54.66	45.34	47.06	52.94	50.96	49.04
Cardano	52.88	47.12	47.74	52.26	52.07	47.93
Chainlink	56.44	43.56	46.65	53.35	49.86	50.14
Dash	53.29	46.71	47.88	52.12	47.38	52.62
Decentraland	50.14	49.86	48.84	51.16	52.62	47.38
EOS	51.64	48.36	48.7	51.3	54.82	45.18
Enjin Coin	58.22	41.78	48.7	51.3	51.79	48.21
Ethereum	51.51	48.49	45.14	54.86	50.96	49.04
Ethereum Classic	50	50	47.61	52.39	52.89	47.11
Filecoin	54.79	45.21	52.8	47.2	52.62	47.38
IOTA	52.47	47.53	47.06	52.94	47.66	52.34
Litecoin	53.29	46.71	47.06	52.94	49.86	50.14
Maker	53.42	46.58	48.02	51.98	55.1	44.9
Monero	50.68	49.32	44.87	55.13	47.66	52.34
NEO	51.78	48.22	46.65	53.35	47.66	52.34
OMG Network	53.01	46.99	49.52	50.48	48.76	51.24
Stellar	52.05	47.95	48.29	51.71	49.04	50.96
Tezos	54.79	45.21	47.88	52.12	49.04	50.96
Theta	51.1	48.9	45.96	54.04	50.69	49.31
TRON	50.82	49.18	45.69	54.31	48.21	51.79
Waves	53.84	46.16	45.83	54.17	54.27	45.73
XRP	52.74	47.26	46.92	53.08	50.69	49.31
Zcash	53.97	46.03	47.47	52.53	51.24	48.76

4.2. Results

Experiment 1: Out-of-sample prediction results

The results obtained from experiment 1 are presented in Table 32 and Table 33. Moreover, Figure 23 illustrates the results through boxplots. An initial observation is that adding technical indicators and correlated close prices to the OHLC feature set enhances both average accuracy and F1-score. In fact, we did some experiments in which the DWT and AE were applied only to OHLC data, and the results were unsatisfactory. This highlights the practicality of combining technical indicators and correlated close prices with OHLC data.

As evident from both boxplots and tables, employing the DWT technique to eliminate noise from the OHLC-TI-Cor feature set has resulted in enhanced accuracy and F1-score for all cryptocurrencies. An interesting finding is that while the utilization of the AE feature set alone does not lead in outstanding prediction performances, its combination with the DWT feature set boosts the prediction performance by a significant amount. The proposed method, i.e., using the AE-DWT feature set achieves the best performances for nearly all cryptocurrencies, except for Ethereum in terms of accuracy and Ethereum Classic and Waves in terms of F1-score. The AE-DWT feature set achieves an average accuracy of 68% and an F1-score of 66.5%. This indicates great improvements compared to the other four scenarios: cases without feature extraction methods (OHLC and OHLC-TI-Cor), and cases with only a single feature extraction method (DWT and AE).

Upon excluding the results of Ethereum, Ethereum Classic, and Waves from the dataset, both accuracy and F1-score exhibit an improvement, reaching 69.5%. These particular cryptocurrencies exhibit higher volatility and variability compared to others, and adding the AE feature set to the DWT feature set does not improve the prediction performance of our GRU model in their cases.

Table 32: Out-of-sample accuracies achieved by GRU model using various feature sets (average and standard deviations of 10 repeated runs)

Cryptocurrency	**OHLC**	**OHLC-TI-Cor**	**DWT**	**AE**	**AE-DWT**
Binance Coin	0.512 (0.019)	0.513 (0.023)	0.594 (0.048)	0.522 (0.018)	**0.684 (0.018)**
Bitcoin	0.513 (0.018)	0.557 (0.018)	0.604 (0.016)	0.563 (0.025)	**0.704 (0.009)**
Bitcoin Cash	0.541 (0.017)	0.557 (0.01)	0.613 (0.013)	0.538 (0.016)	**0.678 (0.017)**
Cardano	0.54 (0.019)	0.528 (0.014)	0.634 (0.016)	0.534 (0.012)	**0.683 (0.011)**
Chainlink	0.495 (0.01)	0.478 (0.014)	0.643 (0.014)	0.492 (0.013)	**0.654 (0.05)**
Dash	0.509 (0.028)	0.524 (0.012)	0.619 (0.013)	0.535 (0.01)	**0.679 (0.015)**
Decentraland	0.533 (0.011)	0.538 (0.006)	0.611 (0.022)	0.533 (0.012)	**0.715 (0.013)**
EOS	0.545 (0.013)	0.566 (0.012)	0.67 (0.012)	0.584 (0.009)	**0.726 (0.01)**
Enjin Coin	0.532 (0.026)	0.538 (0.015)	0.604 (0.017)	0.537 (0.013)	**0.691 (0.015)**
Ethereum	0.502 (0.002)	0.503 (0.004)	**0.595 (0.049)**	0.501 (0.001)	0.556 (0.084)
Ethereum Classic	0.528 (0.014)	0.524 (0.009)	**0.58 (0.035)**	0.527 (0.011)	**0.58 (0.064)**
Filecoin	0.534 (0.015)	0.55 (0.007)	0.571 (0.016)	0.553 (0.009)	**0.707 (0.024)**
IOTA	0.536 (0.016)	0.547 (0.013)	0.624 (0.028)	0.546 (0.01)	**0.708 (0.06)**
Litecoin	0.516 (0.008)	0.514 (0.009)	0.628 (0.023)	0.556 (0.011)	**0.702 (0.022)**
Maker	0.552 (0.012)	0.558 (0.009)	0.633 (0.008)	0.542 (0.009)	**0.668 (0.013)**
Monero	0.548 (0.019)	0.541 (0.013)	0.652 (0.008)	0.536 (0.019)	**0.721 (0.007)**
NEO	0.54 (0.01)	0.549 (0.01)	0.615 (0.023)	0.552 (0.018)	**0.696 (0.014)**
OMG Network	0.512 (0.024)	0.514 (0.012)	0.662 (0.01)	0.524 (0.007)	**0.712 (0.009)**
Stellar	0.519 (0.011)	0.53 (0.022)	0.639 (0.046)	0.514 (0.006)	**0.7 (0.064)**
Tezos	0.513 (0.012)	0.556 (0.014)	0.661 (0.015)	0.544 (0.018)	**0.709 (0.02)**
Theta	0.517 (0.026)	0.52 (0.019)	0.665 (0.007)	0.509 (0.018)	**0.738 (0.013)**
Tron	0.526 (0.01)	0.521 (0.003)	0.572 (0.046)	0.532 (0.017)	**0.585 (0.065)**
Waves	0.543 (0.003)	0.548 (0.009)	0.601 (0.044)	0.544 (0.0)	**0.619 (0.069)**
XRP	0.564 (0.019)	0.583 (0.012)	0.649 (0.012)	0.583 (0.011)	**0.732 (0.008)**
Zcash	0.505 (0.008)	0.526 (0.012)	0.613 (0.024)	0.523 (0.02)	**0.679 (0.021)**
Average	0.527 (0.017)	0.535 (0.022)	0.622 (0.028)	0.537 (0.021)	**0.681 (0.047)**

Table 33: Out-of-sample F1-scores achieved by GRU model using various feature sets (average and standard deviations of 10 repeated runs)

Cryptocurrency	**OHLC**	**OHLC-TI-Cor**	**DWT**	**AE**	**AE-DWT**
Binance Coin	0.33 (0.279)	0.441 (0.24)	0.57 (0.174)	0.469 (0.186)	**0.676 (0.014)**
Bitcoin	0.403 (0.167)	0.44 (0.108)	0.576 (0.03)	0.448 (0.079)	**0.671 (0.02)**
Bitcoin Cash	0.48 (0.085)	0.464 (0.055)	0.602 (0.03)	0.496 (0.035)	**0.675 (0.024)**
Cardano	0.42 (0.2)	0.429 (0.061)	0.617 (0.035)	0.47 (0.094)	**0.673 (0.014)**
Chainlink	0.517 (0.228)	0.422 (0.19)	0.671 (0.02)	0.36 (0.282)	**0.665 (0.029)**
Dash	0.46 (0.302)	0.643 (0.045)	0.638 (0.033)	0.659 (0.034)	**0.694 (0.015)**
Decentraland	0.38 (0.182)	0.537 (0.023)	0.486 (0.081)	0.514 (0.056)	**0.68 (0.023)**
EOS	0.356 (0.182)	0.452 (0.036)	0.634 (0.019)	0.493 (0.02)	**0.69 (0.014)**
Enjin Coin	0.388 (0.132)	0.478 (0.095)	0.584 (0.046)	0.502 (0.092)	**0.671 (0.024)**
Ethereum	0.601 (0.2)	0.601 (0.2)	0.613 (0.044)	0.534 (0.267)	**0.668 (0.004)**
Ethereum Classic	0.116 (0.156)	0.134 (0.17)	**0.511 (0.109)**	0.173 (0.226)	0.354 (0.281)
Filecoin	0.24 (0.183)	0.426 (0.036)	0.369 (0.067)	0.469 (0.029)	**0.658 (0.039)**
IOTA	0.612 (0.084)	0.595 (0.058)	0.654 (0.056)	0.638 (0.056)	**0.748 (0.026)**
Litecoin	0.612 (0.066)	0.595 (0.077)	0.65 (0.024)	0.545 (0.039)	**0.705 (0.025)**
Maker	0.214 (0.204)	0.235 (0.113)	0.547 (0.032)	0.239 (0.179)	**0.622 (0.039)**
Monero	0.666 (0.029)	0.644 (0.053)	0.688 (0.01)	0.687 (0.016)	**0.743 (0.009)**
NEO	0.678 (0.017)	0.613 (0.088)	0.631 (0.068)	0.624 (0.032)	**0.712 (0.025)**
OMG Network	0.402 (0.28)	0.495 (0.108)	0.658 (0.013)	0.592 (0.1)	**0.723 (0.013)**
Stellar	0.662 (0.056)	0.634 (0.05)	0.651 (0.026)	0.677 (0.011)	**0.716 (0.027)**
Tezos	0.592 (0.206)	0.607 (0.06)	0.67 (0.028)	0.564 (0.061)	**0.712 (0.026)**
Theta	0.513 (0.259)	0.489 (0.172)	0.653 (0.022)	0.561 (0.205)	**0.738 (0.014)**
Tron	0.682 (0.005)	0.679 (0.016)	0.652 (0.05)	0.634 (0.053)	**0.677 (0.024)**
Waves	0.314 (0.0)	0.305 (0.076)	**0.373 (0.255)**	0.33 (0.0)	0.369 (0.301)
XRP	0.571 (0.013)	0.532 (0.027)	0.645 (0.021)	0.557 (0.021)	**0.729 (0.013)**
Zcash	0.348 (0.248)	0.219 (0.153)	0.573 (0.059)	0.378 (0.189)	**0.667 (0.024)**
Average	0.462 (0.154)	0.484 (0.140)	0.597 (0.083)	0.505 (0.128)	**0.665 (0.094)**

Figure 23: Boxplots of out-of-sample accuracies achieved by GRU using various feature sets

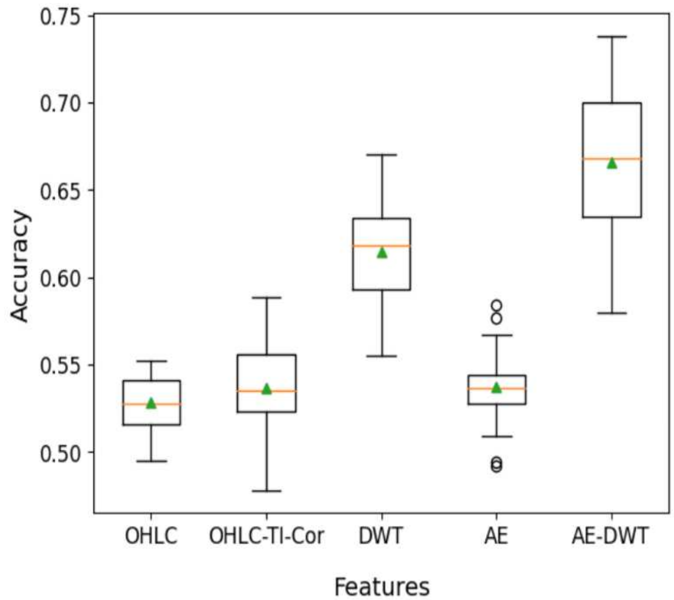
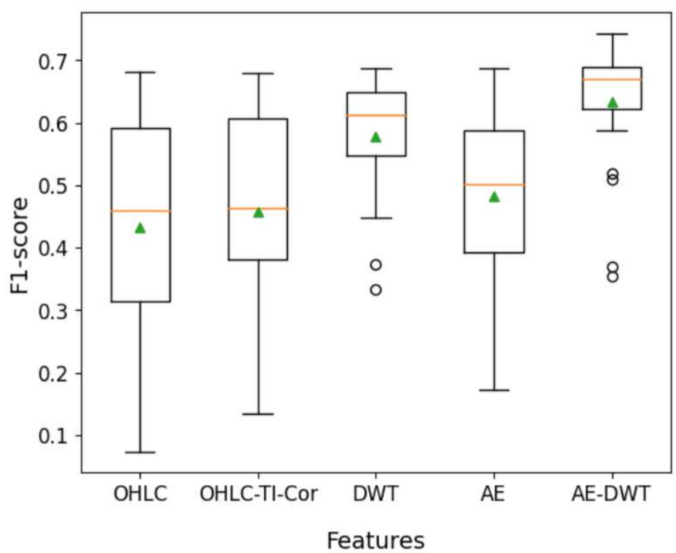


Figure 24: Boxplots of out-of-sample F1-scores achieved by GRU using various feature sets



Experiment 2: Comparing the prediction performance of the proposed method in various periods: before, during, and after COVID-19 pandemic

To gauge the robustness of our proposed method across different periods with varying degrees of volatility, we assessed its performance in challenging periods like the COVID-19 pandemic. As can be seen in Table 34 and Figure 25, the accuracy achieved by the GRU model with AE-DWT feature set is stable regardless of which period it was evaluated. According to F1-

scores presented in Figure 26, a slight difference can be seen between the F1-scores of the pre- and post-pandemic periods but no significant difference between the F1-score of the during-pandemic compared to those of the pre- and post-pandemic periods. All in all, periods with higher volatilities like the COVID-19 pandemic does not affect the predictability of our proposed method.

Table 34: AE-DWT feature set prediction results across three sub-periods (averages and standard deviations from 10 repeated runs)

	Accuracy			F1-score		
Cryptocurrency	**Before pandemic**	**During pandemic**	**After pandemic**	**Before pandemic**	**During pandemic**	**After pandemic**
Binance Coin	0.646 (0.035)	0.651 (0.024)	0.712 (0.025)	0.635 (0.102)	0.68 (0.024)	0.741 (0.03)
Bitcoin	0.628 (0.017)	0.653 (0.014)	0.564 (0.047)	0.579 (0.051)	0.648 (0.015)	0.505 (0.193)
Bitcoin Cash	0.652 (0.023)	0.685 (0.014)	0.646 (0.094)	0.593 (0.038)	0.695 (0.014)	0.673 (0.097)
Cardano	0.64 (0.021)	0.621 (0.032)	0.658 (0.017)	0.679 (0.031)	0.583 (0.046)	0.687 (0.023)
Chainlink	0.611 (0.037)	0.659 (0.013)	0.692 (0.033)	0.596 (0.048)	0.704 (0.021)	0.708 (0.056)
Dash	0.658 (0.013)	0.643 (0.015)	0.708 (0.034)	0.61 (0.016)	0.647 (0.032)	0.779 (0.039)
Decentraland	0.649 (0.032)	0.623 (0.017)	0.726 (0.02)	0.69 (0.028)	0.532 (0.052)	0.752 (0.017)
Enjin Coin	0.651 (0.013)	0.651 (0.023)	0.634 (0.037)	0.655 (0.019)	0.597 (0.037)	0.606 (0.1)
EOS	0.658 (0.019)	0.624 (0.032)	0.593 (0.03)	0.599 (0.038)	0.629 (0.062)	0.646 (0.04)
Ethereum	0.628 (0.02)	0.7 (0.019)	0.655 (0.027)	0.612 (0.033)	0.724 (0.017)	0.681 (0.037)
Ethereum Classic	0.588 (0.062)	0.633 (0.029)	0.581 (0.036)	0.592 (0.212)	0.582 (0.08)	0.588 (0.079)
Filecoin	0.703 (0.026)	0.658 (0.028)	0.649 (0.033)	0.699 (0.033)	0.6 (0.047)	0.676 (0.032)
IOTA	0.683 (0.019)	0.633 (0.026)	0.693 (0.03)	0.702 (0.019)	0.678 (0.055)	0.773 (0.018)
Litecoin	0.713 (0.019)	0.671 (0.018)	0.577 (0.026)	0.693 (0.031)	0.683 (0.023)	0.56 (0.035)
Maker	0.604 (0.021)	0.682 (0.031)	0.665 (0.023)	0.554 (0.064)	0.684 (0.046)	0.696 (0.025)
Monero	0.652 (0.02)	0.699 (0.013)	0.606 (0.028)	0.611 (0.036)	0.725 (0.013)	0.726 (0.023)
Neo	0.635 (0.019)	0.666 (0.016)	0.677 (0.039)	0.641 (0.043)	0.66 (0.027)	0.737 (0.039)
OMG Network	0.699 (0.011)	0.632 (0.023)	0.616 (0.034)	0.689 (0.02)	0.641 (0.042)	0.715 (0.039)
Stellar	0.635 (0.017)	0.724 (0.009)	0.662 (0.047)	0.648 (0.015)	0.692 (0.024)	0.713 (0.054)
Tezos	0.628 (0.028)	0.654 (0.029)	0.669 (0.036)	0.614 (0.031)	0.629 (0.039)	0.698 (0.042)
Theta	0.685 (0.008)	0.666 (0.017)	0.693 (0.046)	0.707 (0.011)	0.664 (0.029)	0.731 (0.048)
Tron	0.659 (0.02)	0.681 (0.023)	0.552 (0.024)	0.663 (0.031)	0.71 (0.038)	0.654 (0.045)
Waves	0.618 (0.046)	0.645 (0.022)	0.651 (0.03)	0.642 (0.035)	0.617 (0.045)	0.675 (0.031)
XRP	0.634 (0.035)	0.668 (0.022)	0.667 (0.029)	0.633 (0.028)	0.668 (0.044)	0.649 (0.056)
Zcash	0.671 (0.022)	0.715 (0.038)	0.6 (0.042)	0.663 (0.025)	0.723 (0.043)	0.614 (0.082)
Average	0.649 (0.031)	0.661 (0.028)	0.646 (0.048)	0.640 (0.043)	0.656 (0.050)	0.679 (0.066)

Figure 25: Boxplots of accuracies achieved by the AE-DWT feature set in three sub-periods

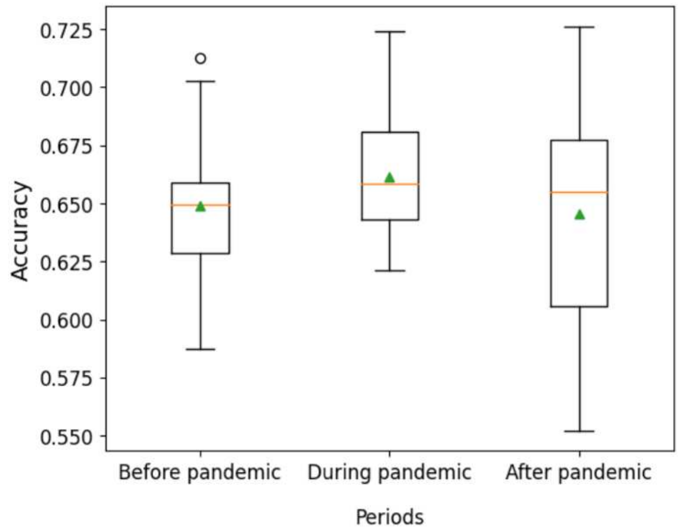
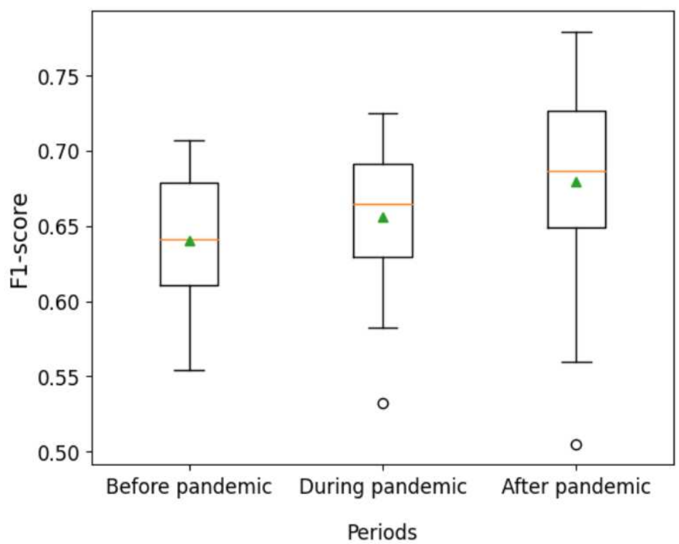


Figure 26: Boxplots of F1-scores achieved by the AE-DWT feature set in three sub-periods



Statistical test results

We conducted a pairwise t-test where each row of two samples corresponds to the same subject; in this context, two accuracy or F1-score values correspond to the same cryptocurrency. Additionally, to increase the power of the test, we utilized the one-tailed version of the test. The respective hypotheses and t-test p-values can be found in Table 35, Table 36, Table 37, and Table 38. All p-values support our findings outlined in the previous section.

The difference between accuracy values obtained by a feature set compared to another is reported in Table 35, along with the corresponding p-values. Similar outcomes for the F1-score are presented in Table 36. As evident from these two tables, the improvements in accuracy and F1-score by the AE-DWT feature set over the base case of OHLC-TI-Cor — where no feature extraction was employed — are approximately 15% and 18%, respectively. These improvements are statistically significant at 1% level of significance.

The t-test results for the experiment 2 are also presented in Table 37 and Table 38. The p-values are available in front of the performance difference values. In terms of accuracy, all p-values are big enough to conclude that the accuracy of the AE-DWT feature set remains consistent across different periods. Concerning F1-score, the only significant difference is observed when comparing the after-pandemic and before-pandemic periods, where the F1-score is approximately 4% higher in the post-pandemic period. However, the average F1-score achieved for the during-pandemic period is not statistically different from those achieved for the other two periods.

Table 35: Out-of-sample accuracy improvement (%) by feature set i with respect to feature set j (i and j are row and column indices, respectively. $H_o: \overline{ACC}_{(i)} - \overline{ACC}_{(j)} \leq 0$, $H_a: \overline{ACC}_{(i)} - \overline{ACC}_{(j)} > 0$)

	OHLC	OHLC-TI-Cor	DWT	AE	AE-DWT
OHLC	-	-	-	-	-
OHLC-TI-Cor	0.83 (0.0030)	-	-	-	-
DWT	9.5 (0.0000)	8.67 (0.0000)	-	-	-
AE	1.0 (0.0025)	0.16 (0.2598)	-8.51 (1.0000)	-	-
AE-DWT	15.39 (0.0000)	14.56 (0.0000)	5.89 (0.0000)	14.4 (0.0000)	-

Table 36: Out-of-sample F1-score improvement (%) by feature set i with respect to feature set j (i and j are row and column indices, respectively. $H_o: \overline{F1}_{(i)} - \overline{F1}_{(j)} \leq 0$, $H_a: \overline{F1}_{(i)} - \overline{F1}_{(j)} > 0$)

	OHLC	OHLC-TI-Cor	DWT	AE	AE-DWT
OHLC	-	-	-	-	-
OHLC-TI-Cor	2.2 (0.0833)	-	-	-	-
DWT	13.44 (0.0000)	11.24 (0.0000)	-	-	-
AE	4.22 (0.0128)	2.02 (0.0239)	-9.21 (0.9999)	-	-
AE-DWT	20.31 (0.0000)	18.1 (0.0000)	6.87 (0.0000)	16.08 (0.0000)	-

Table 37: Difference in accuracies (%) from period i to period j (i and j are row and column indices, respectively. $H_o: \overline{ACC}_{(i)} - \overline{ACC}_{(j)} \leq 0$, $H_a: \overline{ACC}_{(i)} - \overline{ACC}_{(j)} > 0$)

	Before pandemic	During pandemic	After pandemic
Before pandemic	-	-	-
During pandemic	1.23 (0.080)	-	-
After pandemic	-0.34 (0.609)	-1.56 (0.897)	-

Table 38: Difference in F1-scores (%) from period i to period j (i and j are row and column indices, respectively. $H_o: \overline{F1}_{(i)} - \overline{F1}_{(j)} \leq 0$, $H_a: \overline{F1}_{(i)} - \overline{F1}_{(j)} > 0$)

	Before pandemic	During pandemic	After pandemic
Before pandemic	-	-	-
During pandemic	1.58 (0.146)	-	-
After pandemic	3.93 (0.005)	2.35 (0.081)	-

Summary of findings:

- Enriching the OHLC feature set with technical indicators and correlated close prices significantly elevates the predictive capability of the GRU model.
- The DWT feature set consistently outperforms results derived from OHLC, OHLC-TI-Cor, and AE feature sets across all cryptocurrencies studied.
- While the AE feature set alone does not excel in prediction performance, its combination with the DWT feature set notably enhances the prediction performances.
- The AE-DWT feature set achieves the highest average accuracy and F1-score.
- Regardless of any period, the stability of AE-DWT accuracy remains consistent. Notably, F1-score demonstrates improved performance in post-pandemic compared to pre-pandemic periods.

5. Discussion

In this section, we examine the robustness of our findings by providing the results of statistical tests along with comparing the predictive performance of our method with some benchmark results from the recent studies found in literature.

5.1. Comparison with other methods

To prove the superiority of our proposed method from another perspective, we explore whether alternative machine learning or neural network models can achieve comparable results in our experiments. We break down this into two questions as follows:

1. Can our proposed feature extraction technique be integrated with mainstream models? To answer this, we replace the GRU with an ANN model.
2. Can other feature extraction techniques be better integrated with the GRU model? To answer this, we employ another feature extraction technique so-called Principal Component Analysis (PCA) in place of our proposed LSTM-Autoencoder model.

Using the same testing strategies described in Table 30 (experiment 1), we obtained the out-of-sample accuracy and F1-scores. According to the results reported in the Appendix C2, the combination of features achieves the highest accuracies and F1-scores. This is in consistent with the results of our proposed method which employs the combined features as inputs of the GRU model. Therefore, combination of DWT features with other features extracted either by AE or PCA shows the highest performance. In addition, when the ANN is used in place of the GRU, the same pattern for the improvement of performance results can be observed. Nonetheless, our proposed method which includes the AE-DWT feature set as inputs for the GRU is superior to the other tested methods (See Figure 27)

Figure 27: Boxplots of accuracies achieved by three methods

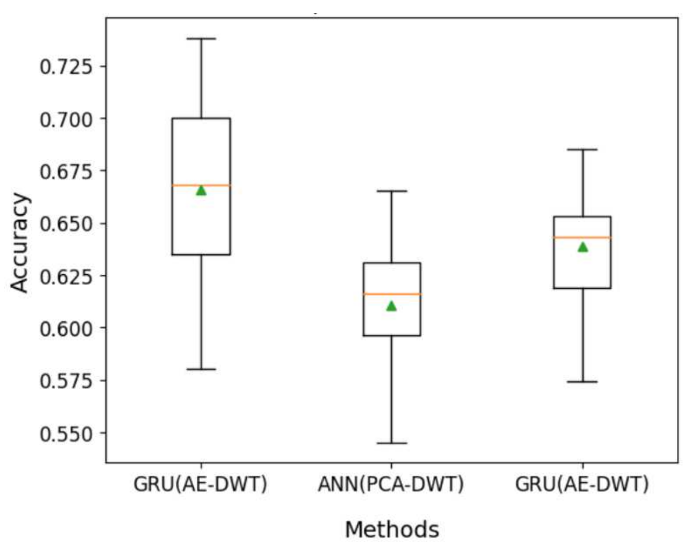
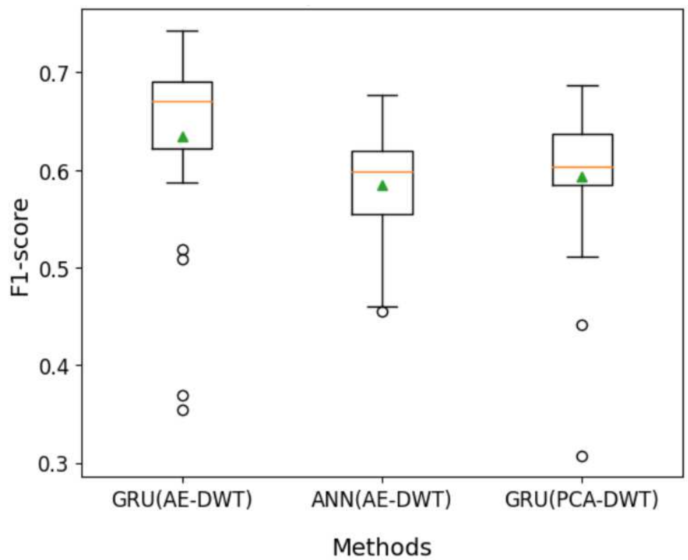


Figure 28: Boxplots of F1-scores achieved by three methods



5.2. Comparison with recent studies

Table 39 and Table 40 present a comparative analysis of our method's results with those reported in (Zhong et al., 2023) and (Ozer & Okan Sakar, 2022). We selected these two papers due to their recent publication dates in 2022 and 2023. Their choice of up-to-date data, encompassing multiple cryptocurrencies, closely aligns with the scope of our study. To make fair comparisons, we employed identical data periods and frequencies as used in both aforementioned studies. The authors of (Zhong et al., 2023) carried out their experiments on a dataset containing 645 cryptocurrencies, while our study focused on datasets comprising 25 cryptocurrencies. On the other hand, the authors of (Ozer & Okan Sakar, 2022) restricted their analysis to three cryptocurrencies — Bitcoin, Ethereum, and Litecoin — deriving results from the maximum average accuracy achieved across four distinct test sets. Thus, to facilitate direct comparisons, we adopted the same cryptocurrencies and test sets in our evaluation.

As can be seen in Table 39, our approach obtains an approximate 10% enhancement in accuracy and a remarkable 20% enhancement in F1-score compared to the best results reported by Zhong et al. (2023). Nevertheless, it should be noted that our results represent averages across 25 cryptocurrencies, whereas Zhong et al. (2023) centered their results on 645 cryptocurrencies. Regarding the results presented in Table 4 our approach achieves 7% to 9% accuracy improvement compared to the results reported by Ozer & Okan Sakar (2022). The reason for getting different results in this section compared to the results reported in Table 32 and Table 33 could be due to the fact that data periods used in this section are different than the one used in our own experiments.

Table 39: Comparing our method's performance with a recent study (Zhong et al., 2023)

Model	**Experimental setup**	**Accuracy**	**F1-score**
Model from Zhong et al. (2023): long short-term memory - reactionwise graph attention network (LSTMReGAT)	Data period: from March 30, 2020, to December 20, 2022 %10 for testing sample Testing the model with the following setting: • Learning rate: 0.0001 • Embedding size F=64 Time window length T=6 • Dropout rate: 0.3 The number of epochs: 20 • The number of iterations K=2 The number of attention head M=6 The number of neurons in DNN: 256, 128 • Average of 10 repeated runs for 645 cryptocurrencies from Table 5 of the (Zhong et al., 2023)	0.6297	0.5469
Our model: GRU with AE-DWT feature set	Data period: from March 30, 2020, to December 20, 2022 %10 for testing sample Testing the model with optimized hyperparameters achieved for each cryptocurrency's dataset Fixed parameters: • Time window length T = 14 • Batch size: 32 The number of epochs: 100 • Early stopping with patience of 10 • Average of 10 repeated runs for 25 cryptocurrencies	0.7333	0.7494

Table 40: Comparing our method's accuracy with a recent study (Ozer & Okan Sakar, 2022)

Model	**Bitcoin**	**Ethereum**	**Litecoin**	**Experimental setup**
The maximum accuracy reported for the daily data in Table 7 of (Ozer & Okan Sakar, 2022)	0.531	0.531	0.540	Four different training and test periods in the period of 2017-07-01 to 2021-04-30 The average results over four test sets are compared
Our model: GRU with AE-DWT feature set	0.618	0.601	0.619	

5.3. Trading simulation

To provide insights into the practical effectiveness of our proposed method for cryptocurrency trading, we conducted trading simulations for all 25 cryptocurrencies. Utilizing our prediction models, we devised a buy-sell strategy based on the models' binary predictions. Buy signals were generated when the model predicted a positive price movement (y=1), while sell signals were triggered for negative predictions (y=0). Please refer to the appendix C3 for more details on the trading algorithm. By simulating the execution of trades using historical data, we calculated the investment returns by considering the price differences between buy and sell points. Additionally, we computed the Sharpe Ratio, a key metric for assessing risk-adjusted returns, using the Equation (48):

$$SharpeRatio = \frac{E(R_s) - R_{rf}}{\sigma_s} \tag{48}$$

where $E(R_s)$ represents the expected value of returns obtained from the buy and hold strategy based on the trades executed according to our models' predictions, R_{rf} is the risk-free return, and σ_s indicates the standard derivation of strategy returns. Since treasury yields are often considered as a proxy for the risk-free rate of return in financial analysis, we used this measure in our computations. Given that our test sample spans from February 2022 to February 2023, we used the average of daily treasury yield rates of the USA reported for the year 2022[7] as the risk-free rate. In addition, given the volatile and relatively short history of cryptocurrencies, and the fact that we are evaluating short-term investments (trades), we opted for yields with durations of four weeks as they provide a more responsive estimate of the risk-free rate that aligns with the fast-paced nature of cryptocurrency trading. We divided the average value (1.6%) by the number of trading days in our test set, which amounts to 363 days, to annualize the risk-free rate for daily calculations. Consequently, the resulting Sharpe ratio reflects a daily Sharpe ratio.

A higher Sharpe ratio indicates a better risk-adjusted return, suggesting that the investment or strategy has generated more return for each unit of risk taken. Conversely, a lower Sharpe ratio suggests that the investment might not be adequately compensating the investor for the risk undertaken. Table 41 presents the Sharpe ratios calculated in our trading simulations for all 25

[7] https://home.treasury.gov/resource-center/data-chart-center/interest-rates/TextView?type=daily_treasury_bill_rates&field_tdr_date_value=2022

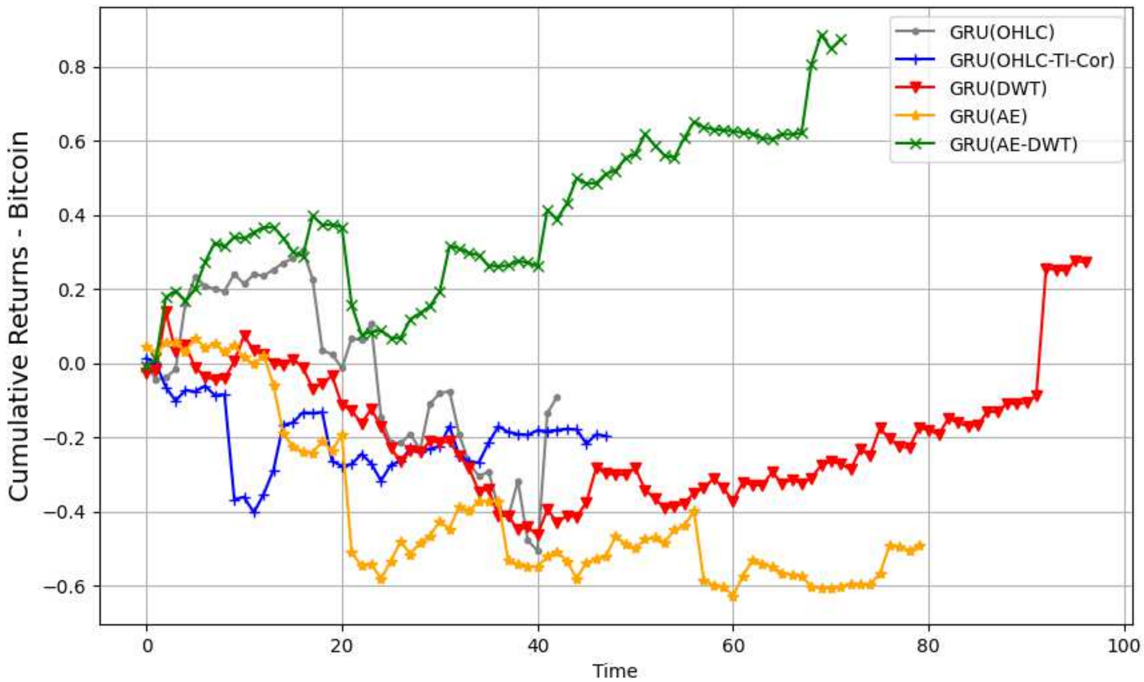
cryptocurrencies using each feature set (method). As this table shows, the AE-DWT feature set achieves significantly better performances, where the Sharpe ratios are improved greatly compared to those obtained by the other feature sets. If the model could not capture the existing pattern in data, it predicts only one of the classes of 0 or 1. Then, according to the trading algorithm (See Appendix C3), the Sharpe ratios cannot be calculated, and this will result in NaN values for the respective feature set. It is worth mentioning that cryptocurrency markets are known for their extreme volatility, which can lead to significant fluctuations in prices. Higher volatility typically translates to higher standard deviations of the returns, which can result in lower Sharp ratios. That is why we obtain ratios that are typically less than 1.

Additionally, Figure 29 presents the buy-sell strategy excess returns over time for Bitcoin and XRP as two example cryptocurrencies. According to this figure, the cumulative excess returns are significantly higher for the AE-DWT feature set, and they fall above the zero line, indicating the superiority of this feature set in generating returns compared to other feature sets (methods). This underscores the effectiveness of utilizing the AE-DWT feature set in enhancing both risk-adjusted returns and overall profitability in cryptocurrency trading strategies.

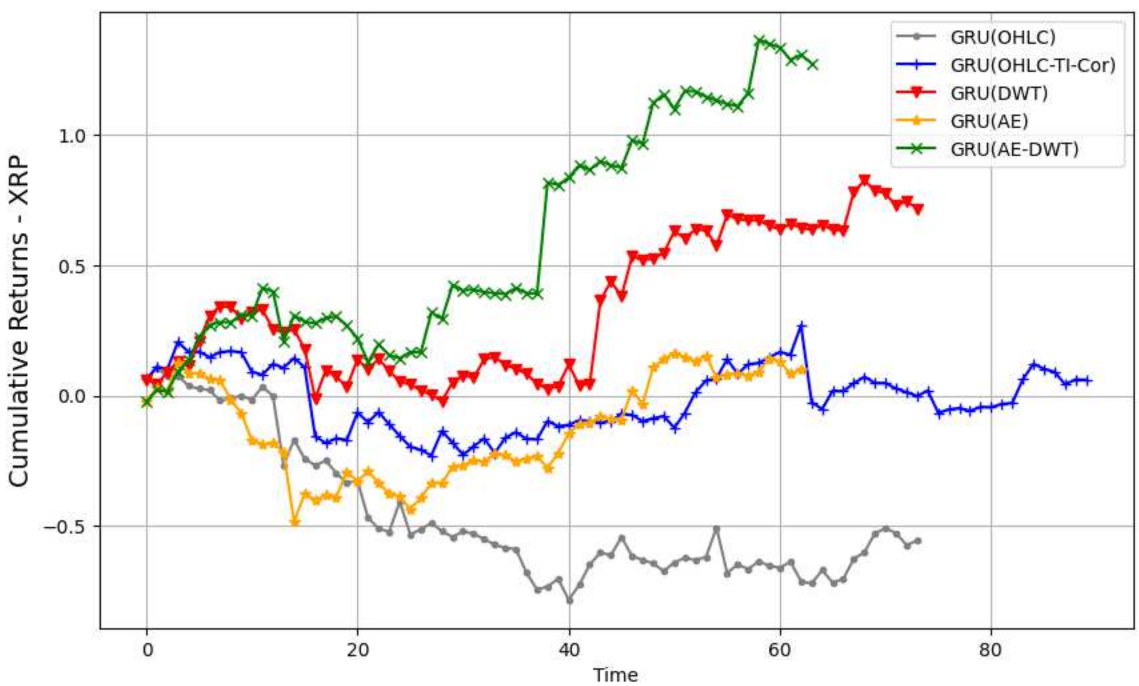
Table 41: Sharpe ratio values obtained based on the buy-sell strategy using various feature sets (methods)

	GRU(OHLC)	GRU(OHLC-TI-Cor)	GRU(DWT)	GRU(AE)	GRU(AE-DWT)
BinanceCoin	0.218	0.725	0.279	NaN	0.272
Bitcoin	-0.022	-0.073	-0.145	-0.113	0.230
BitcoinCash	-0.103	0.022	-0.017	0.002	0.265
Cardano	-0.029	-0.075	0.052	-0.072	0.225
Chainlink	-1.061	NaN	0.066	-0.520	0.425
Dash	NaN	0.113	-0.004	-0.201	0.258
Decentraland	NaN	0.008	0.100	-0.064	0.264
EOS	0.069	0.090	0.237	0.041	0.342
EnjinCoin	0.111	-0.134	-0.056	-0.137	0.136
Ethereum	NaN	NaN	0.208	NaN	0.341
EthereumClassic	0.002	NaN	0.444	NaN	0.432
Filecoin	-0.303	-0.113	0.298	-0.092	0.217
IOTA	-0.054	-0.040	0.049	-0.162	0.135
Litecoin	0.002	0.188	0.023	0.048	0.541
Maker	-0.072	0.005	0.133	-0.209	0.197
Monero	0.017	0.023	0.077	NaN	0.255
NEO	-0.149	-0.310	-0.058	-0.281	0.148
OMG_Network	NaN	-0.428	0.115	-0.082	0.228
Stellar	NaN	NaN	0.158	NaN	0.363
Tezos	NaN	-0.070	-0.027	-0.163	0.279
Theta	NaN	-0.226	0.108	-0.087	0.275
Tron	NaN	NaN	0.042	NaN	0.245
Waves	NaN	NaN	0.374	NaN	0.190
XRP	-0.124	0.011	0.052	0.028	0.146
Zcash	0.033	-0.057	-0.012	-0.226	0.239

Figure 29**:** Buy-Sell strategy excess returns over time for a) Bitcoin b) XRP



(a)



(b)

5.4. Implications of our study with respect to Fama's weak-form market efficiency

In this study, we investigated the effectiveness of utilizing technical indicators and advanced feature extraction techniques for cryptocurrency price forecasting. Our findings revealed promising results, indicating that incorporating these methods can lead to improved predictive accuracy. To contextualize our findings, we discuss their alignment with Fama's weak-form market efficiency hypothesis, which posits that all information contained in past prices and trading volumes is already reflected in current asset prices, making it impossible for investors to consistently outperform the market using historical data alone. In the context of financial markets, technical analysis, which relies on historical price data and various technical indicators, is a common method used by traders to forecast future price movements. In our study, we exclusively focus on technical features derived from OHLC data and additional technical indicators, which are commonly utilized in technical analysis. By employing advanced feature extraction techniques such as Discrete Wavelet Transform (DWT) and Autoencoder, we aim to extract more meaningful and informative features from the raw data. Our results demonstrate that feature sets derived from these advanced techniques, particularly when combined, outperform traditional feature sets containing only OHLC data and technical indicators when used in a GRU model for cryptocurrency price forecasting. This finding is intriguing in the context of weak-form efficiency, as it suggests that with incorporating sophisticated feature extraction methods, it is possible to achieve better predictive performance, potentially exploiting inefficiencies in the market that are not fully captured by traditional technical analysis methods alone. This implies that there may be patterns or relationships in the historical price data that are not fully captured by current market prices, allowing for the development of predictive models that outperform traditional approaches.

6. Conclusion

This study proposes a novel feature extraction method that enhances prediction performance of DL models in predicting directional movement of cryptocurrency markets. For each of the 25 cryptocurrencies analyzed in this study, the initial feature set includes historical prices, technical indicators, and highly correlated close prices from other cryptocurrencies' datasets. First, the hidden representations are extracted from the initial feature set using an LSTM-Autoencoder model. Then, Discrete Wavelet Transform (DWT) is applied to the initial feature set to extract denoised features as smoother versions of the original features. The two sets of features extracted by the LSTM-Autoencoder and DWT techniques are combined, and the resulting feature set is fed into a Gated Recurrent Unit (GRU) model as its input data to predict directional movement of cryptocurrencies' prices.

Our feature extraction method exhibited outstanding performance results for predicting price movement direction of 23 out of 25 cryptocurrencies studied. On average, the accuracy, and F1-score values of 0.681 and 0.665 were achieved which shows 15% and 18% improvement over the baseline cases were no feature extraction is employed. We also measured predictive power of our method during highly volatile periods like the COVID-19 pandemic and compared the results with the results achieved for the periods before and after the pandemic. Our findings indicate that the predictive performance of our method is not affected by high values of volatility in challenging periods like the COVID-19 pandemic.

We demonstrated the effectiveness of our approach not only with the GRU model as the primary prediction model but also with the ANN model as an alternative. However, it is noteworthy that the GRU model consistently outperformed the ANN model in our experiments. Nevertheless, our proposed feature extraction method can be utilized with other types of neural networks such as LSTM, CNN, etc. As future research, we would like to explore the effectiveness of ensemble methods that combines multiple prediction models to potentially achieve higher predictive performances. In addition, we plan to explore the potential improvements of accuracy and F1 scores by incorporating people's sentiment along with macroeconomic factors.

Our findings have implications for traders, investors, and researchers who are interested in forecasting price movement direction in cryptocurrency markets and may help in making more informed trading decisions.

Thesis Conclusion

This thesis consists of three essays addressing cryptocurrency market analytics from various perspectives. The first essay focuses on predicting cryptocurrencies' volatility by integrating deep learning with traditional econometrics methods, specifically GARCH-type models. Using close price data from the 27 most traded cryptocurrencies, we calculated price returns. The standard deviation of close prices was used as a proxy for the actual values of volatility, considering volatility as a latent variable that cannot be directly measured. The returns data were utilized as initial features for the econometrics method, and the conditional standard deviation of returns data was estimated as volatilities. These volatilities were then used as inputs for the deep learning methods. In essence, the econometric methods acted as feature extractors for the deep learning models, forming what we refer to as hybrid models. More precisely, a history of the previous six days' data, including both actual and estimated values of volatilities, was employed to predict the volatility for the subsequent day. Our results revealed that not only are deep learning methods superior to GARCH-type models in forecasting volatility of cryptocurrencies, but also utilizing the outputs of GARCH-type models as inputs for the deep learning models can significantly improve the forecasting errors of such models.

The second essay addressed the prediction of cryptocurrencies' price values, in particular, daily close prices. Similar to the approach in the first essay, our proposed method combined a feature extractor component with deep learning models in a hybrid way. The feature extractor generated sentiments from tweets related to cryptocurrencies, utilizing three pre-trained language models in an ensemble manner for more reliable sentiment extraction compared to relying on a single model. Our proposed model featured a flexible architecture, accommodating different components and variations in the way input data are fed into the model. This adaptability allows it to be tailored for different cryptocurrencies with distinct characteristics. Specifically, we examined input sequences of both close prices and sentiment data with lengths of 7, 14, and 21 to determine the optimal sequence length. Our results suggest that using longer data sequences is more beneficial in predicting daily close prices of cryptocurrencies. Furthermore, sentiment data proved to be valuable for over 70% of the cryptocurrencies analyzed in this thesis. This finding underscores the significance of people's opinions and sentiments expressed in their social media posts, particularly on platforms like Twitter, in influencing and predicting price values in cryptocurrency markets.

In the third essay, we framed the prediction of cryptocurrency markets as a classification problem, aiming to forecast whether the next day's price will rise or fall. Similar to the approaches in essays 1 and 2, our prediction model integrates a feature extraction component that generates high-quality features for a deep learning model, predicting the directional movement of daily close prices in cryptocurrency markets. We calculated technical indicators from historical prices and employed the Discrete Wavelet Transform (DWT) method to eliminate noise from the data. Simultaneously, we input the initial data into a Long Short-Term Memory Autoencoder (LSTM-AE) model to extract informative features. The encoder part of the autoencoder compresses the input into a lower-dimensional representation (latent space), capturing essential features while discarding noise and redundant information. The features derived from both methods were combined and used as final input data for a Gated Recurrent Unit (GRU) model. This approach significantly improved the accuracy and F1-score of predictions by 15% and 18%, respectively, compared to cases where no feature extraction method was employed before feeding the data into the GRU model. We tested our approach across three distinct periods: before, during, and after the COVID-19 pandemic, characterized by varying levels of volatilities in cryptocurrency markets. Remarkably, our method demonstrated stable performance across all evaluated periods.

Overall, this thesis makes multiple contributions to existing literature:

- Determining the most effective GARCH-type model for accurate out-of-sample forecasting of cryptocurrency volatility.
- Evaluating and comparing the efficacy of common deep learning models, Deep Feedforward Neural Network (DFFNN) and LSTM, in enhancing the forecasting capabilities of GARCH-type models.
- Investigating the impact of incorporating GARCH-type forecasts into DFFNN and LSTM models on the improvement of overall forecasting accuracies.
- Utilizing a comprehensive dataset featuring 27 cryptocurrencies to draw general and robust conclusions, diverging from the common focus on Bitcoin in previous studies.

- Proposing a flexible hybrid model for predicting daily close prices of cryptocurrencies based on LSTM and Convolutional Neural Network (CNN), tailored for different cryptocurrencies to accommodate their unique characteristics, and leveraging the strengths of both models.
- Demonstrating the effectiveness of sentiment data from a vast Twitter dataset in predicting the close prices of over 70% of the analyzed cryptocurrencies.
- Distinguishing our sentiment analysis by employing an ensemble of three pre-trained language models, emphasizing the importance of contextual understanding in finance and social media sentiment analysis.
- Innovatively applying DWT for data denoising in cryptocurrency market trend prediction, contributing to the classification tasks in a field where its use has been limited.
- Introducing a novel strategy by combining features extracted by DWT and LSTM-Autoencoder models, aiming to enhance the classification performance of GRU in predicting cryptocurrency market trends.
- Assessing the proposed approach's efficacy during different challenging periods, including pre, during, and post-COVID-19 pandemic, reflecting varying degrees of cryptocurrency market volatility.

Looking ahead, our future research in cryptocurrency prediction will extend beyond single-day forecasts, reaching into multi-day scenarios to provide deeper insights for investors with longer-term perspectives. Additionally, working with minutely or hourly data, presents an opportunity to refine predictions by capturing finer-grained patterns in market dynamics.

Numerous factors are influencing the behavior of cryptocurrency markets. By incorporating feature extraction methods and integrating social media sentiment analysis, we have gained deeper insights into the dynamic nature of cryptocurrency markets. Future research endeavors will explore the inclusion of additional influencing factors. Among them macroeconomic indicators like inflation and interest rates, technological advancements, and market liquidity.

Investors and finance experts can apply the insights from this thesis to improve how they make decisions about cryptocurrency investments. Our research expands the understanding of cryptocurrency analysis, providing a foundation for academics and researchers to explore more factors and refine predictive models. Risk management professionals, especially in companies dealing with digital assets, can use our predictive models to identify and reduce risks associated with cryptocurrency investments. Data scientists and analysts can find inspiration in our methods, using them in various fields beyond just cryptocurrency. The techniques we developed in this thesis have the potential to be useful in many different areas.

References

Adcock, R., & Gradojevic, N. (2019). Non-fundamental, non-parametric Bitcoin forecasting. *Physica A: Statistical Mechanics and Its Applications*, *531*. https://doi.org/10.1016/j.physa.2019.121727

Alonso-Monsalve, S., Suárez-Cetrulo, A. L., Cervantes, A., & Quintana, D. (2020). Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. *Expert Systems with Applications*, *149*. https://doi.org/10.1016/j.eswa.2020.113250

Amirshahi, B., & Lahmiri, S. (2023a). Hybrid deep learning and GARCH-family models for forecasting volatility of cryptocurrencies. *Machine Learning with Applications*, *12*, 100465. https://doi.org/10.1016/j.mlwa.2023.100465

Amirshahi, B., & Lahmiri, S. (2023b). Investigating the effectiveness of Twitter sentiment in cryptocurrency close price prediction by using deep learning. *Expert Systems*. https://doi.org/10.1111/exsy.13428

Anbaee Farimani, S., Vafaei Jahan, M., Milani Fard, A., & Tabbakh, S. R. K. (2022). Investigating the informativeness of technical indicators and news sentiment in financial market price prediction. *Knowledge-Based Systems*, *247*. https://doi.org/10.1016/j.knosys.2022.108742

Araci, D. (2019). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. http://arxiv.org/abs/1908.10063

Aras, S. (2021). Stacking hybrid GARCH models for forecasting Bitcoin volatility. *Expert Systems with Applications*, *174*. https://doi.org/10.1016/j.eswa.2021.114747

Barbieri, F., Camacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). *TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification*. http://arxiv.org/abs/2010.12421

Basher, S. A., & Sadorsky, P. (2022). Forecasting Bitcoin price direction with random forests: How important are interest rates, inflation, and market volatility? *Machine Learning with Applications*, *9*, 100355. https://doi.org/10.1016/j.mlwa.2022.100355

Beltagy, I., Lo, K., & Cohan, A. (2019). *SciBERT: A Pretrained Language Model for Scientific Text*. http://arxiv.org/abs/1903.10676

Bollerslev, T. (1986). GENERALIZED AUTOREGRESSIVE CONDITIONAL HETEROSKEDASTICITY. In *Journal of Econometrics* (Vol. 31).

Bollerslev, T. (1987). A Conditionally Heteroskedastic Time Series Model for Speculative Prices and Rates of Return. In *Source: The Review of Economics and Statistics* (Vol. 69, Issue 3). https://about.jstor.org/terms

Cavalli, S., & Amoretti, M. (2021). CNN-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, *101*. https://doi.org/10.1016/j.asoc.2020.107065

Chang, V., Baudier, P., Zhang, H., Xu, Q., Zhang, J., & Arami, M. (2020). How Blockchain can impact financial services – The overview, challenges and recommendations from expert interviewees. *Technological Forecasting and Social Change*, *158*. https://doi.org/10.1016/j.techfore.2020.120166

Chen, W., Xu, H., Jia, L., & Gao, Y. (2021). Machine learning model for Bitcoin exchange rate prediction using economic and technology determinants. *International Journal of Forecasting*, *37*(1), 28–43. https://doi.org/10.1016/j.ijforecast.2020.02.008

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*. https://doi.org/10.1016/j.cam.2019.112395

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*.

D'Amato, V., Levantesi, S., & Piscopo, G. (2022). Deep learning in predicting cryptocurrency volatility. *Physica A: Statistical Mechanics and Its Applications*, *596*. https://doi.org/10.1016/j.physa.2022.127158

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, *41*, 909–996. https://api.semanticscholar.org/CorpusID:2727398

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. http://arxiv.org/abs/1810.04805

Ding, Z., Granger, C. W. J., & Engle, R. F. (1993). A long memory property of stock market returns and a new model*. In *Journal of Empirical Finance* (Vol. 1).

Donoho, D. L., Johnstone, I. M., Kerkyacharian, G., & Picard, D. (1995). Wavelet Shrinkage: Asymptopia? *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(2), 301–337. https://doi.org/10.1111/j.2517-6161.1995.tb02032.x

Engle, R. F. (1982). *Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation* (Vol. 50, Issue 4). https://www.jstor.org/stable/1912773

Engle, R. F., & Bollerslev, T. (1986). Modelling the persistence of conditional variances. *Econometric Reviews*, *5*(1), 1–50. https://doi.org/10.1080/07474938608800095

Engle, R.F. and Lee, G. (1999) A Permanent and Transitory Component Model of Stock Return Volatility. Cointegration, Causality and Forecasting A Festschrift in Honor of Clive W.J. Granger. Oxford University Press, New York. - References - Scientific Research Publishing. Retrieved May 8, 2022, from https://scirp.org/reference/referencespapers.aspx?referenceid=1232518

Glosten, L. R., Jagannathan, R., & Runkle, D. E. (1993). On the Relation between the Expected Value and the Volatility of the Nominal Excess Return on Stocks. In *Source: The Journal of Finance* (Vol. 48, Issue 5).

Goutte, S., Le, H. V., Liu, F., & von Mettenheim, H. J. (2023). Deep learning and technical analysis in cryptocurrency market. *Finance Research Letters*, *54*. https://doi.org/10.1016/j.frl.2023.103809

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

Guo, H., Zhang, D., Liu, S., Wang, L., & Ding, Y. (2021). Bitcoin price forecasting: A perspective of underlying blockchain transactions. *Decision Support Systems*, *151*. https://doi.org/10.1016/j.dss.2021.113650

Hajizadeh, E., Seifi, A., Fazel Zarandi, M. H., & Turksen, I. B. (2012). A hybrid modeling approach for forecasting the volatility of S&P 500 index return. *Expert Systems with Applications*, *39*(1), 431–436. https://doi.org/10.1016/j.eswa.2011.07.033

He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://arxiv.org/abs/1512.03385

Hiew, J. Z. G., Huang, X., Mou, H., Li, D., Wu, Q., & Xu, Y. (2019). *BERT-based Financial Sentiment Index and LSTM-based Stock Return Predictability*. http://arxiv.org/abs/1906.09024

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366. https://doi.org/10.1016/0893-6080(89)90020-8

Hu, Y., Ni, J., & Wen, L. (2020). A hybrid deep learning approach by integrating LSTM-ANN networks with GARCH model for copper price volatility prediction. *Physica A: Statistical Mechanics and Its Applications*, *557*. https://doi.org/10.1016/j.physa.2020.124907

Huang, K., Altosaar, J., & Ranganath, R. (2019). *ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission*. http://arxiv.org/abs/1904.05342

Hutto, C. J., & Gilbert, E. (2014). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. http://sentic.net/

Jaquart, P., Dann, D., & Weinhardt, C. (2021). Short-term bitcoin market prediction via machine learning. *Journal of Finance and Data Science*, *7*, 45–66. https://doi.org/10.1016/j.jfds.2021.03.001

Ji, G., Yu, J., Hu, K., Xie, J., & Ji, X. (2022). An adaptive feature selection schema using improved technical indicators for predicting stock price movements. *Expert Systems with Applications*, *200*. https://doi.org/10.1016/j.eswa.2022.116941

Ji, S., Kim, J., & Im, H. (2019). A comparative study of bitcoin price prediction using deep learning. *Mathematics*, *7*(10). https://doi.org/10.3390/math7100898

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. https://github.com/Microsoft/LightGBM.

Khaldi, R., el Afia, A., & Chiheb, R. (2019). Forecasting of BTC volatility: comparative study between parametric and nonparametric models. *Progress in Artificial Intelligence*, *8*(4), 511–523. https://doi.org/10.1007/s13748-019-00196-w

Kim, H. M., Bock, G. W., & Lee, G. (2021). Predicting Ethereum prices with machine learning based on Blockchain information. *Expert Systems with Applications*, *184*. https://doi.org/10.1016/j.eswa.2021.115480

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

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). *D Convolutional Neural Networks and Applications-A Survey*.

Kraaijeveld, O., & de Smedt, J. (2020). The predictive power of public Twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, *65*. https://doi.org/10.1016/j.intfin.2020.101188

Kristjanpoller R, W., & Hernández P, E. (2017). Volatility of main metals forecasted by a hybrid ANN-GARCH model with regressors. *Expert Systems with Applications*, *84*, 290–300. https://doi.org/10.1016/j.eswa.2017.05.024

Kristjanpoller, W., Fadic, A., & Minutolo, M. C. (2014). Volatility forecast using hybrid Neural Network models. *Expert Systems with Applications*, *41*(5), 2437–2442. https://doi.org/10.1016/j.eswa.2013.09.043

Kristjanpoller, W., & Minutolo, M. C. (2015). Gold price volatility: A forecasting approach using the Artificial Neural Network-GARCH model. *Expert Systems with Applications*, *42*(20), 7245–7251. https://doi.org/10.1016/j.eswa.2015.04.058

Kristjanpoller, W., & Minutolo, M. C. (2016). Forecasting volatility of oil price using an artificial neural network-GARCH model. *Expert Systems with Applications*, *65*, 233–241. https://doi.org/10.1016/j.eswa.2016.08.045

Kristjanpoller, W., & Minutolo, M. C. (2018). A hybrid volatility forecasting framework integrating GARCH, artificial neural network, technical analysis and principal components analysis. *Expert Systems with Applications*, *109*, 1–11. https://doi.org/10.1016/j.eswa.2018.05.011

Lahmiri, S. (2017). Modeling and predicting historical volatility in exchange rate markets. *Physica A: Statistical Mechanics and Its Applications*, *471*, 387–395. https://doi.org/10.1016/j.physa.2016.12.061

Lahmiri, S., & Bekiros, S. (2019). Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solitons and Fractals*, *118*, 35–40. https://doi.org/10.1016/j.chaos.2018.11.014

Lahmiri, S., & Bekiros, S. (2020a). Intelligent forecasting with machine learning trading systems in chaotic intraday Bitcoin market. *Chaos, Solitons and Fractals*, *133*. https://doi.org/10.1016/j.chaos.2020.109641

Lahmiri, S., & Bekiros, S. (2020b). The impact of COVID-19 pandemic upon stability and sequential irregularity of equity and cryptocurrency markets. *Chaos, Solitons and Fractals*, *138*. https://doi.org/10.1016/j.chaos.2020.109936

Lahmiri, S., & Bekiros, S. (2021). Deep Learning Forecasting in Cryptocurrency High-Frequency Trading. In *Cognitive Computation* (Vol. 13, Issue 2, pp. 485–487). Springer. https://doi.org/10.1007/s12559-021-09841-w

Lahmiri, S., Bekiros, S., & Salvi, A. (2018). Long-range memory, distributional variation and randomness of bitcoin volatility. *Chaos, Solitons and Fractals*, *107*, 43–48. https://doi.org/10.1016/j.chaos.2017.12.018

Lahmiri, S., & Boukadoum, M. (2015). An ensemble system based on hybrid EGARCH-ANN with different distributional assumptions to predict s&P 500 intraday volatility. *Fluctuation and Noise Letters*, *14*(1). https://doi.org/10.1142/S0219477515500017

Li, S., Huang, X., Cheng, Z., Zou, W., & Yi, Y. (2023). AE-ACG: A novel deep learning-based method for stock price movement prediction. *Finance Research Letters*, *58*, 104304. https://doi.org/10.1016/j.frl.2023.104304

Liu, M., Li, G., Li, J., Zhu, X., & Yao, Y. (2021). Forecasting the price of Bitcoin using deep learning. *Finance Research Letters*, *40*. https://doi.org/10.1016/j.frl.2020.101755

Liu, Y. (2019). Novel volatility forecasting using deep learning–Long Short Term Memory Recurrent Neural Networks. *Expert Systems with Applications*, *132*, 99–109. https://doi.org/10.1016/j.eswa.2019.04.038

Loureiro, D., Barbieri, F., Neves, L., Anke, L. E., & Camacho-Collados, J. (2022). *TimeLMs: Diachronic Language Models from Twitter*. http://arxiv.org/abs/2202.03829

Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. www.bitcoin.org

Malo, P., Sinha, A., Takala, P., Korhonen, P., & Wallenius, J. (2013). *Good Debt or Bad Debt: Detecting Semantic Orientations in Economic Texts*. http://arxiv.org/abs/1307.5336

Nakano, M., Takahashi, A., & Takahashi, S. (2018). Bitcoin technical trading with artificial neural network. *Physica A: Statistical Mechanics and Its Applications*, *510*, 587–609. https://doi.org/10.1016/j.physa.2018.07.017

Nelson, D. B. (1991). *Conditional Heteroskedasticity in Asset Returns: A New Approach* (Vol. 59, Issue 2).

Nobre, J., & Neves, R. F. (2019). Combining Principal Component Analysis, Discrete Wavelet Transform and XGBoost to trade in the financial markets. *Expert Systems with Applications*, *125*, 181–194. https://doi.org/10.1016/j.eswa.2019.01.083

Orte, F., Mira, J., Sánchez, M. J., & Solana, P. (2023). A random forest-based model for crypto asset forecasts in futures markets with out-of-sample prediction. *Research in International Business and Finance*, *64*. https://doi.org/10.1016/j.ribaf.2022.101829

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022a). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, *198*. https://doi.org/10.1016/j.eswa.2022.116804

Ortu, M., Uras, N., Conversano, C., Bartolucci, S., & Destefanis, G. (2022b). On technical trading and social media indicators for cryptocurrency price classification through deep learning. *Expert Systems with Applications*, *198*. https://doi.org/10.1016/j.eswa.2022.116804

Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023a). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications*, *213*. https://doi.org/10.1016/j.eswa.2022.119233

Oyedele, A. A., Ajayi, A. O., Oyedele, L. O., Bello, S. A., & Jimoh, K. O. (2023b). Performance evaluation of deep learning and boosted trees for cryptocurrency closing price prediction. *Expert Systems with Applications*, *213*, 119233. https://doi.org/10.1016/j.eswa.2022.119233

Ozer, F., & Okan Sakar, C. (2022). An automated cryptocurrency trading system based on the detection of unusual price movements with a Time-Series Clustering-Based approach. *Expert Systems with Applications*, *200*. https://doi.org/10.1016/j.eswa.2022.117017

Parvini, N., Abdollahi, M., Seifollahi, S., & Ahmadian, D. (2022). Forecasting Bitcoin returns with long short-term memory networks and wavelet decomposition: A comparison of several market determinants. *Applied Soft Computing*, *121*. https://doi.org/10.1016/j.asoc.2022.108707

Patel, M. M., Tanwar, S., Gupta, R., & Kumar, N. (2020). A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions. *Journal of Information Security and Applications*, *55*. https://doi.org/10.1016/j.jisa.2020.102583

Peng, Y., Albuquerque, P. H. M., Camboim de Sá, J. M., Padula, A. J. A., & Montenegro, M. R. (2018). The best of two worlds: Forecasting high frequency volatility for cryptocurrencies and traditional currencies with Support Vector Regression. *Expert Systems with Applications*, *97*, 177–192. https://doi.org/10.1016/j.eswa.2017.12.004

Pérez, J. M., Furman, D. A., Alemany, L. A., & Luque, F. (2021). *RoBERTuito: a pre-trained language model for social media text in Spanish*. http://arxiv.org/abs/2111.09453

Pérez-Enciso, M., & Zingaretti, L. M. (2019). A guide for using deep learning for complex trait genomic prediction. In *Genes* (Vol. 10, Issue 7). MDPI AG. https://doi.org/10.3390/genes10070553

Puzyrev, V. (2019). *Deep convolutional autoencoder for cryptocurrency market analysis*. http://arxiv.org/abs/1910.12281

Quoc Nguyen, D., Vu, T., Tuan Nguyen, A., & Research, V. (2020). *BERTweet: A pre-trained language model for English Tweets*. https://pypi.org/project/emoji

Ramos-Pérez, E., Alonso-González, P. J., & Núñez-Velázquez, J. J. (2019). Forecasting volatility with a stacked model based on a hybridized Artificial Neural Network. *Expert Systems with Applications*, *129*, 1–9. https://doi.org/10.1016/j.eswa.2019.03.046

Rosenthal, S., Farra, N., & Nakov, P. (2017). *SemEval-2017 Task 4: Sentiment Analysis in Twitter*. https://trends24.in/

Sener, E., & Demir, I. (2022). Wavelet-Denoised ARIMA and LSTM Method to Predict Cryptocurrency Price. *ISMSIT 2022 - 6th International Symposium on Multidisciplinary Studies and Innovative Technologies, Proceedings*, 577–582. https://doi.org/10.1109/ISMSIT56059.2022.9932727

Seo, M., & Kim, G. (2020). Hybrid forecasting models based on the neural networks for the volatility of bitcoin. *Applied Sciences (Switzerland)*, *10*(14). https://doi.org/10.3390/app10144768

Shintate, T., & Pichl, L. (2019). Trend Prediction Classification for High Frequency Bitcoin Time Series with Deep Learning. *Journal of Risk and Financial Management*, *12*(1), 17. https://doi.org/10.3390/jrfm12010017

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). *Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank*. http://nlp.stanford.edu/

Sun, X., Liu, M., & Sima, Z. (2020a). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, *32*. https://doi.org/10.1016/j.frl.2018.12.032

Sun, X., Liu, M., & Sima, Z. (2020b). A novel cryptocurrency price trend forecasting model based on LightGBM. *Finance Research Letters*, *32*. https://doi.org/10.1016/j.frl.2018.12.032

Tapia, S., & Kristjanpoller, W. (2022). Framework based on multiplicative error and residual analysis to forecast bitcoin intraday-volatility. *Physica A: Statistical Mechanics and Its Applications*, *589*. https://doi.org/10.1016/j.physa.2021.126613

Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.1111/J.2517-6161.1996.TB02080.X

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762

Vidal, A., & Kristjanpoller, W. (2020). Gold volatility prediction using a CNN-LSTM approach. *Expert Systems with Applications*, *157*. https://doi.org/10.1016/j.eswa.2020.113481

Wu, D., Wang, X., & Wu, S. (2022a). A hybrid framework based on extreme learning machine, discrete wavelet transform, and autoencoder with feature penalty for stock prediction. *Expert Systems with Applications*, *207*. https://doi.org/10.1016/j.eswa.2022.118006

Wu, D., Wang, X., & Wu, S. (2022b). A hybrid framework based on extreme learning machine, discrete wavelet transform, and autoencoder with feature penalty for stock prediction. *Expert Systems with Applications*, *207*. https://doi.org/10.1016/j.eswa.2022.118006

Yu, M. (2019). Forecasting Bitcoin volatility: The role of leverage effect and uncertainty. *Physica A: Statistical Mechanics and Its Applications*, *533*. https://doi.org/10.1016/j.physa.2019.03.072

Zakoian, J.-M. (1994). Threshold heteroskedastic models. In *Journal of Economic Dynamics and Control* (Vol. 18).

Zeng, Z., & Khushi, M. (2020). Wavelet Denoising and Attention-based RNN- ARIMA Model to Predict Forex Price. *2020 International Joint Conference on Neural Networks (IJCNN)*, 1–7. https://doi.org/10.1109/IJCNN48605.2020.9206832

Zhang, Z., Dai, H. N., Zhou, J., Mondal, S. K., García, M. M., & Wang, H. (2021). Forecasting cryptocurrency price using convolutional neural networks with weighted and attentive memory channels. *Expert Systems with Applications*, *183*. https://doi.org/10.1016/j.eswa.2021.115378

Zhao, Y., & Khushi, M. (2020). Wavelet Denoised-ResNet CNN and LightGBM Method to Predict Forex Rate of Change. *IEEE International Conference on Data Mining Workshops, ICDMW*, *2020-November*, 385–391. https://doi.org/10.1109/ICDMW51313.2020.00060

Zhong, C., Du, W., Xu, W., Huang, Q., Zhao, Y., & Wang, M. (2023). LSTM-ReGAT: A network-centric approach for cryptocurrency price trend prediction. *Decision Support Systems*, *169*. https://doi.org/10.1016/j.dss.2023.113955

Zhou, Z., Song, Z., Xiao, H., & Ren, T. (2023). Multi-source data driven cryptocurrency price movement prediction and portfolio optimization. *Expert Systems with Applications*, *219*. https://doi.org/10.1016/j.eswa.2023.119600

Zhu, Y., Kiros, R., Zemel, R., Salakhutdinov, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). *Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books*. http://arxiv.org/abs/1506.06724

Zou, Y., & Herremans, D. (2022). *A multimodal model with Twitter FinBERT embeddings for extreme price movement prediction of Bitcoin*. http://arxiv.org/abs/2206.00648

Zoumpekas, T., Houstis, E., & Vavalis, M. (2020). ETH analysis and predictions utilizing deep learning. *Expert Systems with Applications*, *162*. https://doi.org/10.1016/j.eswa.2020.113866

Appendices

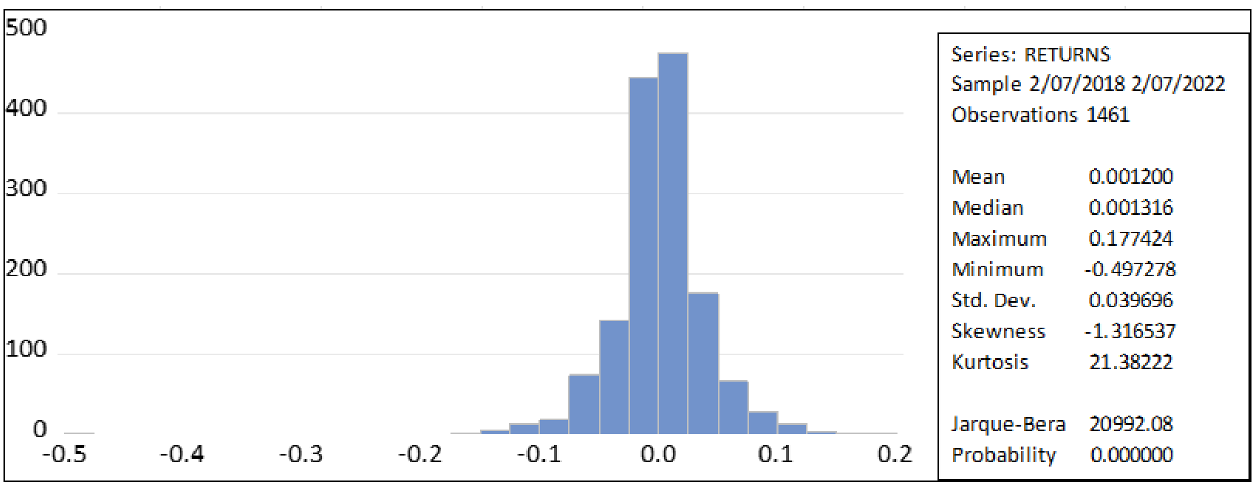
Appendix A. Study 1

A1. Bitcoin’s descriptive statistics as an example: output from EViews software

Unit root tests for returns data

		t-Statistic	Prob.*
Augmented Dickey-Fuller (ADF) test statistic		-41.54251	0
Test critical values:	1% level	-3.434621	
	5% level	-2.863313	
	10% level	-2.567763	
		Adj. t-Stat	Prob.*
Phillips-Perron (PP) test statistic		-41.39538	0
Test critical values:	1% level	-3.434621	
	5% level	-2.863313	
	10% level	-2.567763	

Histogram of returns and normality test (Jarque-Bera) results



A2. The results of model estimation for GARCH – normal model for Bitcoin data as an example: output from EViews software

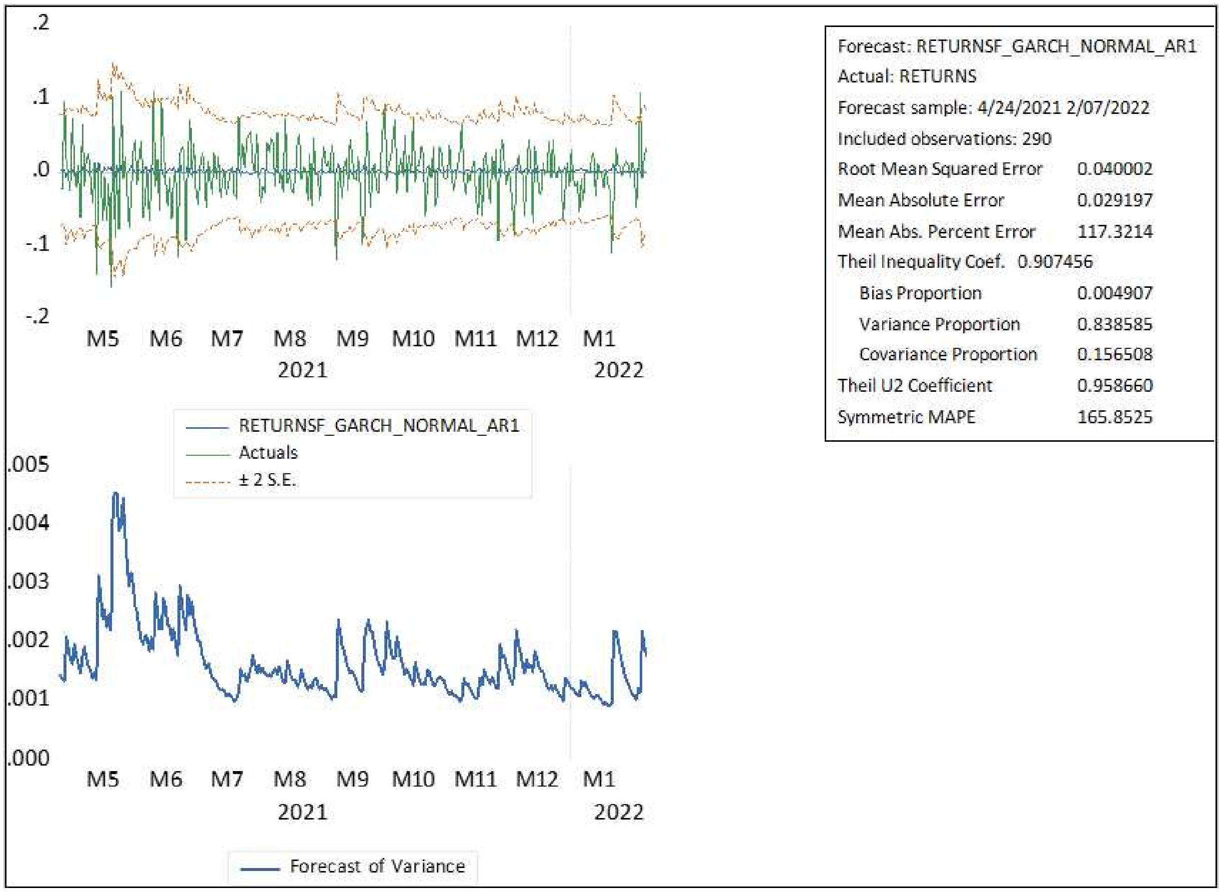
Estimation using the whole sample (2/07/2018 to 2/07/2022) for the purpose of feature generation for hybrid models

Mean equation	AIC	Log likelihood	non-significant parameters	Serial Correlation LM(12) Test			
zero mean	-3.665926	2680.959		F-statistic	0.218319	Prob. F(12,1436)	0.9977
				Obs*R-squared	2.638728	Prob. Chi-Square(12)	0.9976
constant mean	-3.666376	2682.288		F-statistic	0.229417	Prob. F(12,1436)	0.997
				Obs*R-squared	2.772614	Prob. Chi-Square(12)	0.9969
AR(1)	**-3.669136**	**2683.469**		F-statistic	0.261707	Prob. F(12,1435)	0.9944
				Obs*R-squared	3.162011	Prob. Chi-Square(12)	0.9943
AR(2)	-3.669833	2683.143	AR(2)	F-statistic	0.260838	Prob. F(12,1434)	0.9945
				Obs*R-squared	3.151552	Prob. Chi-Square(12)	0.9944
ARMA(1,1)	-3.668509	2684.012	AR(1) and MA(1)	F-statistic	0.256792	Prob. F(12,1435)	0.9949
				Obs*R-squared	3.102763	Prob. Chi-Square(12)	0.9948

Estimation using the training sample (2/07/2018 to 4/23/2021) for the purpose of volatility forecasting in the testing sample

Mean equation	AIC	Log likelihood	non-significant parameters	Serial Correlation LM(12) Test			
AR(1)	-3.686034	2161.33		F-statistic	0.229948	Prob. F(12,1145)	0.997
				Obs*R-squared	2.783997	Prob. Chi-Square(12)	0.9969

Forecasting the returns and conditional variance in the testing sample (4/24/2021 to 2/07/2022)



A3: Calculated and forecasted historical volatilities in the testing set of Bitcoin data as an example: output from EViews software

Date	HV	HV_apgarch _ged	HV_apgarch _normal	HV_apgarch _t	HV_egarch _ged	HV_egarch _normal	HV_egarch _t	HV_garch _ged	HV_garch _normal	HV_garch _t
2021-04-24	0.03002	0.03749	0.04423	0.04217	0.03722	0.04354	0.04137	0.03505	0.03819	0.04044
2021-04-25	0.02713	0.03714	0.04404	0.04165	0.03689	0.04329	0.04087	0.03448	0.03736	0.03971
2021-04-26	0.03280	0.03708	0.04421	0.04144	0.03681	0.04336	0.04064	0.03410	0.03683	0.03921
2021-04-27	0.03300	0.04406	0.04660	0.05133	0.04350	0.04566	0.05010	0.04157	0.04599	0.04905
2021-04-28	0.03240	0.04397	0.04486	0.05134	0.04339	0.04405	0.05005	0.04076	0.04411	0.04797
2021-04-29	0.03238	0.04127	0.04223	0.04812	0.04104	0.04170	0.04731	0.03930	0.04180	0.04619
2021-04-30	0.03537	0.04075	0.04253	0.04735	0.04053	0.04192	0.04659	0.03851	0.04058	0.04524
2021-05-01	0.03537	0.04507	0.04381	0.05364	0.04449	0.04315	0.05228	0.04171	0.04449	0.04951
2021-05-02	0.03555	0.04293	0.04133	0.05088	0.04260	0.04091	0.04990	0.04025	0.04215	0.04770
2021-05-03	0.03509	0.04199	0.04139	0.04960	0.04172	0.04092	0.04874	0.03926	0.04071	0.04650
2021-05-04	0.03664	0.03983	0.03915	0.04707	0.03982	0.03895	0.04655	0.03790	0.03884	0.04483
2021-05-05	0.03867	0.04256	0.04414	0.05005	0.04223	0.04364	0.04909	0.04020	0.04206	0.04780
2021-05-06	0.03869	0.04545	0.04452	0.05455	0.04482	0.04399	0.05309	0.04177	0.04402	0.05006
2021-05-07	0.03823	0.04339	0.04306	0.05209	0.04299	0.04266	0.05097	0.04037	0.04195	0.04840
2021-05-08	0.03791	0.04177	0.04098	0.05027	0.04154	0.04079	0.04937	0.03907	0.04008	0.04680
2021-05-09	0.03796	0.04164	0.03992	0.05030	0.04139	0.03982	0.04933	0.03834	0.03904	0.04586
2021-05-10	0.03834	0.03965	0.03863	0.04776	0.03962	0.03864	0.04713	0.03707	0.03748	0.04427
2021-05-11	0.03846	0.04090	0.04161	0.04890	0.04067	0.04132	0.04803	0.03774	0.03861	0.04504
2021-05-12	0.04580	0.03909	0.03951	0.04684	0.03907	0.03945	0.04624	0.03650	0.03709	0.04350
2021-05-13	0.04415	0.04834	0.05323	0.05735	0.04804	0.05377	0.05615	0.05108	0.05608	0.06183
2021-05-14	0.04420	0.04599	0.05020	0.05415	0.04591	0.05050	0.05337	0.04914	0.05244	0.05951
2021-05-15	0.04534	0.04308	0.04655	0.05066	0.04333	0.04698	0.05035	0.04723	0.04915	0.05724
2021-05-16	0.04520	0.04586	0.05100	0.05373	0.04572	0.05082	0.05295	0.04874	0.05070	0.05920
2021-05-17	0.04625	0.04405	0.04884	0.05136	0.04407	0.04854	0.05085	0.04701	0.04780	0.05710
2021-05-18	0.04507	0.04677	0.05302	0.05439	0.04642	0.05213	0.05342	0.04857	0.04962	0.05908
2021-05-19	0.05241	0.04563	0.05161	0.05285	0.04534	0.05053	0.05202	0.04711	0.04717	0.05727
2021-05-20	0.05641	0.05608	0.06612	0.06490	0.05543	0.06501	0.06341	0.06214	0.06655	0.07580
2021-05-21	0.05776	0.06031	0.06515	0.07201	0.05911	0.06346	0.06978	0.06369	0.06756	0.07848
2021-05-22	0.05759	0.06248	0.06862	0.07478	0.06085	0.06595	0.07213	0.06447	0.06758	0.07992
2021-05-23	0.05884	0.05857	0.06324	0.07003	0.05733	0.06055	0.06806	0.06182	0.06271	0.07684
2021-05-24	0.06301	0.06098	0.06687	0.07280	0.05931	0.06339	0.07039	0.06283	0.06325	0.07817
2021-05-25	0.06295	0.06645	0.06670	0.08132	0.06419	0.06289	0.07808	0.06600	0.06688	0.08244
2021-05-26	0.05998	0.06174	0.06081	0.07557	0.05999	0.05750	0.07321	0.06324	0.06206	0.07924
2021-05-27	0.05975	0.05919	0.05679	0.07288	0.05764	0.05382	0.07084	0.06085	0.05805	0.07646
2021-05-28	0.06080	0.05663	0.05462	0.06977	0.05529	0.05180	0.06810	0.05859	0.05457	0.07382
2021-05-29	0.06085	0.05926	0.05940	0.07256	0.05752	0.05616	0.07042	0.05999	0.05658	0.07542
2021-05-30	0.05912	0.05847	0.05911	0.07129	0.05670	0.05564	0.06923	0.05855	0.05423	0.07360
2021-05-31	0.06009	0.05675	0.05565	0.06973	0.05511	0.05253	0.06781	0.05656	0.05136	0.07129
2021-06-01	0.06008	0.05736	0.05384	0.07103	0.05557	0.05093	0.06885	0.05572	0.05023	0.07022
2021-06-02	0.06032	0.05431	0.05122	0.06736	0.05286	0.04864	0.06567	0.05358	0.04742	0.06767
2021-06-03	0.06048	0.05244	0.04848	0.06534	0.05117	0.04629	0.06386	0.05171	0.04512	0.06542
2021-06-04	0.05935	0.05309	0.04741	0.06653	0.05169	0.04544	0.06480	0.05103	0.04466	0.06449
2021-06-05	0.05949	0.05428	0.05088	0.06768	0.05267	0.04872	0.06571	0.05143	0.04622	0.06501

2021-06-06	0.05936	0.05431	0.05227	0.06727	0.05264	0.04988	0.06527	0.05086	0.04576	0.06413
2021-06-07	0.05947	0.05071	0.04841	0.06298	0.04951	0.04652	0.06161	0.04886	0.04324	0.06168
2021-06-08	0.05950	0.05272	0.05251	0.06503	0.05125	0.05033	0.06329	0.05008	0.04588	0.06308
2021-06-09	0.06387	0.05028	0.05012	0.06178	0.04909	0.04809	0.06047	0.04828	0.04354	0.06081
2021-06-10	0.06365	0.05762	0.05260	0.07214	0.05597	0.05050	0.07002	0.05494	0.05347	0.06878
2021-06-11	0.05945	0.05403	0.04958	0.06795	0.05280	0.04784	0.06642	0.05276	0.05019	0.06622
2021-06-12	0.05984	0.05163	0.04674	0.06514	0.05064	0.04534	0.06393	0.05080	0.04736	0.06387
2021-06-13	0.06291	0.05229	0.04934	0.06557	0.05113	0.04771	0.06418	0.05062	0.04743	0.06363
2021-06-14	0.06251	0.05731	0.05074	0.07290	0.05569	0.04906	0.07074	0.05408	0.05261	0.06796
2021-06-15	0.06251	0.05792	0.04954	0.07380	0.05614	0.04791	0.07142	0.05344	0.05115	0.06699
2021-06-16	0.06201	0.05427	0.04672	0.06927	0.05292	0.04541	0.06753	0.05133	0.04807	0.06446
2021-06-17	0.06198	0.05460	0.04915	0.06920	0.05312	0.04758	0.06736	0.05101	0.04781	0.06401
2021-06-18	0.05620	0.05195	0.04717	0.06562	0.05076	0.04573	0.06425	0.04914	0.04524	0.06170
2021-06-19	0.05264	0.05383	0.05138	0.06741	0.05236	0.04960	0.06568	0.05031	0.04736	0.06302
2021-06-20	0.05025	0.05134	0.04920	0.06404	0.05015	0.04753	0.06274	0.04850	0.04486	0.06076
2021-06-21	0.05441	0.04777	0.04562	0.05957	0.04704	0.04437	0.05891	0.04662	0.04244	0.05844
2021-06-22	0.05297	0.05450	0.05615	0.06706	0.05328	0.05483	0.06558	0.05477	0.05459	0.06820
2021-06-23	0.04906	0.05161	0.05228	0.06403	0.05070	0.05119	0.06292	0.05264	0.05122	0.06573
2021-06-24	0.04943	0.05167	0.05033	0.06441	0.05067	0.04928	0.06314	0.05151	0.04932	0.06434
2021-06-25	0.05177	0.05120	0.04833	0.06403	0.05019	0.04737	0.06270	0.05019	0.04730	0.06269
2021-06-26	0.05194	0.05537	0.05548	0.06860	0.05390	0.05407	0.06666	0.05403	0.05310	0.06742
2021-06-27	0.05219	0.05220	0.05156	0.06512	0.05110	0.05042	0.06365	0.05191	0.04984	0.06492
2021-06-28	0.05190	0.05583	0.05180	0.07034	0.05433	0.05051	0.06823	0.05362	0.05197	0.06692
2021-06-29	0.05209	0.05198	0.04788	0.06541	0.05096	0.04699	0.06402	0.05147	0.04874	0.06435
2021-06-30	0.05154	0.05195	0.04650	0.06571	0.05086	0.04570	0.06419	0.05039	0.04721	0.06306
2021-07-01	0.05203	0.04998	0.04565	0.06312	0.04907	0.04486	0.06190	0.04870	0.04503	0.06103
2021-07-02	0.05184	0.05056	0.04801	0.06334	0.04951	0.04691	0.06198	0.04850	0.04507	0.06067
2021-07-03	0.05144	0.04719	0.04467	0.05923	0.04657	0.04393	0.05844	0.04661	0.04263	0.05836
2021-07-04	0.05039	0.04660	0.04314	0.05863	0.04600	0.04255	0.05782	0.04539	0.04126	0.05681
2021-07-05	0.05066	0.04518	0.04129	0.05682	0.04471	0.04088	0.05616	0.04392	0.03959	0.05493
2021-07-06	0.05073	0.04608	0.04416	0.05749	0.04544	0.04350	0.05663	0.04406	0.04062	0.05504
2021-07-07	0.04936	0.04384	0.04175	0.05483	0.04346	0.04136	0.05429	0.04248	0.03884	0.05307
2021-07-08	0.04966	0.04177	0.04036	0.05207	0.04163	0.04009	0.05185	0.04100	0.03734	0.05118
2021-07-09	0.04528	0.04182	0.04174	0.05170	0.04161	0.04128	0.05141	0.04047	0.03732	0.05039
2021-07-10	0.04520	0.04135	0.04042	0.05130	0.04114	0.04012	0.05096	0.03950	0.03656	0.04914
2021-07-11	0.04526	0.03921	0.03884	0.04850	0.03925	0.03870	0.04849	0.03813	0.03529	0.04736
2021-07-12	0.04473	0.03855	0.03765	0.04772	0.03863	0.03764	0.04771	0.03714	0.03456	0.04603
2021-07-13	0.04075	0.03883	0.03936	0.04769	0.03881	0.03918	0.04756	0.03689	0.03506	0.04560
2021-07-14	0.03994	0.03780	0.03901	0.04604	0.03787	0.03879	0.04605	0.03592	0.03429	0.04422
2021-07-15	0.04017	0.03542	0.03676	0.04295	0.03578	0.03679	0.04335	0.03471	0.03323	0.04259
2021-07-16	0.03952	0.03592	0.03848	0.04318	0.03615	0.03833	0.04341	0.03462	0.03385	0.04231
2021-07-17	0.03957	0.03534	0.03850	0.04209	0.03558	0.03828	0.04236	0.03387	0.03337	0.04119
2021-07-18	0.03820	0.03326	0.03634	0.03942	0.03374	0.03638	0.04000	0.03279	0.03245	0.03969
2021-07-19	0.03850	0.03202	0.03482	0.03778	0.03260	0.03502	0.03848	0.03182	0.03172	0.03833
2021-07-20	0.03885	0.03276	0.03673	0.03835	0.03318	0.03673	0.03883	0.03194	0.03257	0.03832
2021-07-21	0.03573	0.03418	0.03933	0.03964	0.03439	0.03907	0.03982	0.03271	0.03402	0.03904
2021-07-22	0.03544	0.03903	0.04100	0.04648	0.03889	0.04068	0.04607	0.03686	0.03962	0.04434
2021-07-23	0.03559	0.03780	0.03911	0.04477	0.03779	0.03896	0.04453	0.03575	0.03796	0.04284
2021-07-24	0.03519	0.03925	0.03900	0.04690	0.03904	0.03888	0.04632	0.03608	0.03826	0.04318
2021-07-25	0.03159	0.03771	0.03724	0.04489	0.03768	0.03729	0.04454	0.03496	0.03674	0.04169

2021-07-26	0.03267	0.03974	0.03771	0.04779	0.03948	0.03773	0.04703	0.03586	0.03787	0.04275
2021-07-27	0.03163	0.04270	0.03872	0.05185	0.04215	0.03866	0.05060	0.03773	0.03996	0.04492
2021-07-28	0.03162	0.04595	0.03991	0.05633	0.04509	0.03976	0.05457	0.03994	0.04233	0.04753
2021-07-29	0.03098	0.04474	0.03854	0.05479	0.04401	0.03850	0.05322	0.03885	0.04054	0.04612
2021-07-30	0.03185	0.04176	0.03639	0.05101	0.04144	0.03658	0.05003	0.03748	0.03865	0.04440
2021-07-31	0.03075	0.04417	0.03740	0.05448	0.04358	0.03753	0.05305	0.03879	0.04032	0.04609
2021-08-01	0.03199	0.04212	0.03672	0.05189	0.04177	0.03691	0.05082	0.03753	0.03867	0.04456
2021-08-02	0.03200	0.04314	0.03993	0.05269	0.04263	0.03981	0.05144	0.03814	0.03955	0.04528
2021-08-03	0.03240	0.04235	0.04032	0.05136	0.04190	0.04007	0.05025	0.03737	0.03850	0.04424
2021-08-04	0.03176	0.04215	0.04142	0.05080	0.04169	0.04100	0.04971	0.03699	0.03808	0.04372
2021-08-05	0.03197	0.04300	0.04092	0.05230	0.04242	0.04057	0.05097	0.03706	0.03827	0.04389
2021-08-06	0.03264	0.04327	0.04014	0.05277	0.04263	0.03986	0.05133	0.03676	0.03779	0.04343
2021-08-07	0.03239	0.04514	0.04040	0.05546	0.04429	0.04011	0.05367	0.03775	0.03901	0.04462
2021-08-08	0.03263	0.04662	0.04048	0.05757	0.04560	0.04018	0.05550	0.03842	0.03968	0.04537
2021-08-09	0.03354	0.04473	0.03989	0.05516	0.04394	0.03965	0.05344	0.03730	0.03828	0.04402
2021-08-10	0.03380	0.04684	0.04044	0.05831	0.04582	0.04018	0.05621	0.03860	0.04004	0.04577
2021-08-11	0.03290	0.04449	0.03928	0.05536	0.04377	0.03915	0.05371	0.03734	0.03841	0.04424
2021-08-12	0.03332	0.04184	0.03744	0.05187	0.04147	0.03748	0.05074	0.03607	0.03682	0.04262
2021-08-13	0.03526	0.04142	0.03859	0.05100	0.04106	0.03848	0.04994	0.03562	0.03648	0.04202
2021-08-14	0.03473	0.04557	0.04032	0.05686	0.04485	0.04013	0.05522	0.03915	0.04125	0.04663
2021-08-15	0.03447	0.04317	0.03900	0.05388	0.04275	0.03896	0.05267	0.03783	0.03943	0.04503
2021-08-16	0.03502	0.04075	0.03735	0.05063	0.04062	0.03744	0.04988	0.03654	0.03772	0.04339
2021-08-17	0.03582	0.04006	0.03801	0.04944	0.03997	0.03799	0.04878	0.03584	0.03695	0.04247
2021-08-18	0.03502	0.04027	0.03968	0.04925	0.04010	0.03944	0.04852	0.03573	0.03696	0.04224
2021-08-19	0.03433	0.03801	0.03779	0.04621	0.03811	0.03773	0.04590	0.03455	0.03556	0.04070
2021-08-20	0.03321	0.03987	0.03811	0.04883	0.03973	0.03804	0.04813	0.03536	0.03676	0.04171
2021-08-21	0.03345	0.04296	0.03915	0.05301	0.04252	0.03901	0.05179	0.03746	0.03930	0.04421
2021-08-22	0.03311	0.04047	0.03750	0.04990	0.04034	0.03756	0.04913	0.03618	0.03763	0.04262
2021-08-23	0.03311	0.03836	0.03572	0.04720	0.03848	0.03596	0.04678	0.03499	0.03614	0.04110
2021-08-24	0.03375	0.03645	0.03416	0.04466	0.03678	0.03454	0.04456	0.03387	0.03485	0.03963
2021-08-25	0.03297	0.03745	0.03702	0.04544	0.03759	0.03714	0.04513	0.03435	0.03586	0.04023
2021-08-26	0.03314	0.03713	0.03620	0.04523	0.03726	0.03641	0.04487	0.03370	0.03520	0.03943
2021-08-27	0.03395	0.03850	0.03944	0.04654	0.03843	0.03938	0.04592	0.03465	0.03681	0.04070
2021-08-28	0.03398	0.03998	0.03940	0.04884	0.03972	0.03936	0.04788	0.03519	0.03759	0.04151
2021-08-29	0.03283	0.03748	0.03713	0.04554	0.03754	0.03736	0.04507	0.03402	0.03608	0.03999
2021-08-30	0.03353	0.03548	0.03565	0.04287	0.03578	0.03599	0.04274	0.03295	0.03481	0.03856
2021-08-31	0.03242	0.03663	0.03839	0.04384	0.03673	0.03845	0.04349	0.03358	0.03588	0.03931
2021-09-01	0.03255	0.03459	0.03651	0.04108	0.03493	0.03672	0.04108	0.03252	0.03461	0.03789
2021-09-02	0.03190	0.03586	0.03649	0.04288	0.03601	0.03669	0.04257	0.03281	0.03507	0.03822
2021-09-03	0.03132	0.03484	0.03518	0.04148	0.03508	0.03549	0.04130	0.03194	0.03404	0.03702
2021-09-04	0.03112	0.03413	0.03420	0.04058	0.03441	0.03458	0.04044	0.03120	0.03327	0.03599
2021-09-05	0.03075	0.03218	0.03279	0.03803	0.03268	0.03329	0.03821	0.03028	0.03237	0.03472
2021-09-06	0.03010	0.03373	0.03329	0.04016	0.03401	0.03373	0.03999	0.03085	0.03332	0.03541
2021-09-07	0.03750	0.03370	0.03287	0.04006	0.03393	0.03330	0.03982	0.03041	0.03288	0.03473
2021-09-08	0.03625	0.04174	0.04518	0.04913	0.04173	0.04628	0.04836	0.04330	0.04907	0.05142
2021-09-09	0.03616	0.04165	0.04552	0.04855	0.04157	0.04613	0.04778	0.04248	0.04696	0.05028
2021-09-10	0.03670	0.03930	0.04261	0.04578	0.03949	0.04334	0.04538	0.04093	0.04431	0.04842
2021-09-11	0.03641	0.03971	0.04392	0.04611	0.03979	0.04430	0.04557	0.04051	0.04338	0.04798
2021-09-12	0.03390	0.03732	0.04111	0.04330	0.03768	0.04167	0.04314	0.03905	0.04115	0.04620
2021-09-13	0.03408	0.03687	0.03963	0.04292	0.03722	0.04022	0.04272	0.03802	0.03959	0.04492

2021-09-14	0.03517	0.03658	0.04010	0.04241	0.03688	0.04050	0.04219	0.03724	0.03861	0.04398
2021-09-15	0.03516	0.03844	0.04007	0.04524	0.03850	0.04042	0.04463	0.03768	0.03921	0.04464
2021-09-16	0.03474	0.03859	0.03909	0.04552	0.03858	0.03944	0.04480	0.03703	0.03822	0.04375
2021-09-17	0.03481	0.03671	0.03768	0.04316	0.03691	0.03812	0.04274	0.03581	0.03672	0.04221
2021-09-18	0.03408	0.03545	0.03703	0.04143	0.03576	0.03743	0.04121	0.03477	0.03555	0.04085
2021-09-19	0.03286	0.03514	0.03608	0.04120	0.03543	0.03654	0.04093	0.03400	0.03479	0.03987
2021-09-20	0.03720	0.03480	0.03675	0.04060	0.03507	0.03707	0.04033	0.03340	0.03440	0.03909
2021-09-21	0.03816	0.04130	0.04652	0.04786	0.04121	0.04680	0.04700	0.04219	0.04585	0.05010
2021-09-22	0.04053	0.04424	0.05080	0.05092	0.04381	0.05040	0.04962	0.04429	0.04762	0.05247
2021-09-23	0.04049	0.04733	0.05048	0.05577	0.04657	0.04993	0.05397	0.04569	0.04888	0.05458
2021-09-24	0.04091	0.04771	0.04870	0.05656	0.04686	0.04812	0.05463	0.04501	0.04716	0.05373
2021-09-25	0.04018	0.04825	0.05044	0.05713	0.04728	0.04951	0.05510	0.04494	0.04676	0.05382
2021-09-26	0.03920	0.04580	0.04780	0.05405	0.04513	0.04697	0.05248	0.04333	0.04421	0.05186
2021-09-27	0.03936	0.04372	0.04494	0.05169	0.04329	0.04436	0.05045	0.04180	0.04197	0.05003
2021-09-28	0.03958	0.04286	0.04477	0.05055	0.04249	0.04411	0.04943	0.04079	0.04068	0.04882
2021-09-29	0.03925	0.04275	0.04545	0.05019	0.04234	0.04461	0.04906	0.04023	0.04003	0.04810
2021-09-30	0.04063	0.04070	0.04278	0.04787	0.04053	0.04224	0.04705	0.03884	0.03828	0.04639
2021-10-01	0.04381	0.04335	0.04303	0.05173	0.04290	0.04248	0.05045	0.04013	0.04015	0.04799
2021-10-02	0.04380	0.05025	0.04578	0.06124	0.04940	0.04508	0.05931	0.04688	0.04867	0.05623
2021-10-03	0.04376	0.04678	0.04288	0.05721	0.04636	0.04253	0.05590	0.04508	0.04583	0.05410
2021-10-04	0.04395	0.04444	0.04058	0.05440	0.04427	0.04045	0.05346	0.04343	0.04336	0.05214
2021-10-05	0.04421	0.04354	0.03927	0.05340	0.04342	0.03925	0.05253	0.04219	0.04159	0.05063
2021-10-06	0.04606	0.04515	0.03951	0.05574	0.04479	0.03947	0.05451	0.04242	0.04188	0.05094
2021-10-07	0.04054	0.04962	0.04143	0.06190	0.04886	0.04125	0.06003	0.04564	0.04597	0.05484
2021-10-08	0.04037	0.04798	0.04148	0.05983	0.04735	0.04127	0.05822	0.04428	0.04408	0.05332
2021-10-09	0.04045	0.04489	0.03915	0.05582	0.04464	0.03914	0.05480	0.04262	0.04176	0.05131
2021-10-10	0.03981	0.04363	0.03785	0.05431	0.04348	0.03796	0.05344	0.04131	0.04006	0.04973
2021-10-11	0.04058	0.04105	0.03626	0.05098	0.04119	0.03652	0.05056	0.03982	0.03827	0.04788
2021-10-12	0.04098	0.04310	0.03705	0.05387	0.04296	0.03724	0.05302	0.04048	0.03950	0.04873
2021-10-13	0.04067	0.04205	0.03765	0.05238	0.04198	0.03776	0.05167	0.03944	0.03846	0.04750
2021-10-14	0.04005	0.04123	0.03668	0.05147	0.04120	0.03688	0.05081	0.03840	0.03735	0.04623
2021-10-15	0.04176	0.03862	0.03472	0.04796	0.03891	0.03513	0.04778	0.03706	0.03587	0.04451
2021-10-16	0.04184	0.04320	0.03704	0.05421	0.04309	0.03730	0.05339	0.04056	0.04097	0.04888
2021-10-17	0.04170	0.04084	0.03591	0.05124	0.04099	0.03630	0.05080	0.03914	0.03913	0.04713
2021-10-18	0.04164	0.03897	0.03450	0.04881	0.03931	0.03500	0.04865	0.03781	0.03750	0.04546
2021-10-19	0.04150	0.03732	0.03327	0.04656	0.03780	0.03385	0.04664	0.03657	0.03607	0.04386
2021-10-20	0.03628	0.03841	0.03370	0.04801	0.03870	0.03421	0.04779	0.03653	0.03632	0.04375
2021-10-21	0.03666	0.03882	0.03372	0.04847	0.03900	0.03417	0.04809	0.03612	0.03597	0.04312
2021-10-22	0.03581	0.04122	0.03887	0.05089	0.04109	0.03905	0.05011	0.03813	0.03922	0.04570
2021-10-23	0.03564	0.04136	0.04040	0.05054	0.04115	0.04031	0.04972	0.03784	0.03879	0.04518
2021-10-24	0.03415	0.03923	0.03829	0.04793	0.03927	0.03841	0.04745	0.03656	0.03717	0.04356
2021-10-25	0.03430	0.03736	0.03705	0.04542	0.03760	0.03726	0.04526	0.03538	0.03582	0.04204
2021-10-26	0.03588	0.03829	0.03691	0.04681	0.03838	0.03713	0.04637	0.03534	0.03602	0.04198
2021-10-27	0.03616	0.03944	0.03990	0.04786	0.03933	0.03985	0.04720	0.03599	0.03731	0.04284
2021-10-28	0.03569	0.04017	0.04192	0.04828	0.03993	0.04159	0.04748	0.03624	0.03770	0.04300
2021-10-29	0.03578	0.04062	0.04103	0.04921	0.04030	0.04079	0.04823	0.03599	0.03746	0.04274
2021-10-30	0.03513	0.04094	0.04014	0.04972	0.04055	0.03997	0.04861	0.03568	0.03700	0.04225
2021-10-31	0.03164	0.03867	0.03837	0.04684	0.03857	0.03838	0.04615	0.03451	0.03562	0.04074
2021-11-01	0.03157	0.03713	0.03747	0.04471	0.03720	0.03753	0.04427	0.03351	0.03456	0.03941
2021-11-02	0.03201	0.03556	0.03647	0.04254	0.03579	0.03658	0.04236	0.03253	0.03359	0.03808

2021-11-03	0.03202	0.03683	0.03650	0.04438	0.03688	0.03663	0.04389	0.03289	0.03435	0.03850
2021-11-04	0.03179	0.03468	0.03489	0.04165	0.03499	0.03520	0.04152	0.03186	0.03330	0.03713
2021-11-05	0.02929	0.03481	0.03625	0.04144	0.03503	0.03636	0.04125	0.03169	0.03344	0.03681
2021-11-06	0.02869	0.03366	0.03567	0.03974	0.03399	0.03579	0.03972	0.03088	0.03270	0.03566
2021-11-07	0.02902	0.03224	0.03417	0.03792	0.03271	0.03445	0.03809	0.03001	0.03191	0.03446
2021-11-08	0.03093	0.03314	0.03412	0.03916	0.03345	0.03439	0.03907	0.03012	0.03232	0.03451
2021-11-09	0.03100	0.03779	0.03625	0.04542	0.03776	0.03639	0.04478	0.03432	0.03753	0.03980
2021-11-10	0.03059	0.03566	0.03486	0.04279	0.03588	0.03517	0.04250	0.03322	0.03608	0.03839
2021-11-11	0.03010	0.03623	0.03693	0.04311	0.03632	0.03700	0.04268	0.03332	0.03625	0.03848
2021-11-12	0.03000	0.03457	0.03573	0.04081	0.03485	0.03587	0.04067	0.03232	0.03498	0.03714
2021-11-13	0.02999	0.03350	0.03533	0.03928	0.03386	0.03548	0.03928	0.03149	0.03406	0.03603
2021-11-14	0.02717	0.03164	0.03362	0.03689	0.03220	0.03395	0.03718	0.03054	0.03304	0.03475
2021-11-15	0.02765	0.03140	0.03296	0.03660	0.03192	0.03333	0.03682	0.02997	0.03250	0.03394
2021-11-16	0.02954	0.03205	0.03493	0.03710	0.03242	0.03509	0.03715	0.03014	0.03308	0.03416
2021-11-17	0.02951	0.03543	0.04034	0.04070	0.03547	0.04021	0.04027	0.03361	0.03745	0.03854
2021-11-18	0.03048	0.03356	0.03825	0.03819	0.03382	0.03827	0.03809	0.03255	0.03597	0.03715
2021-11-19	0.03028	0.03659	0.04299	0.04153	0.03653	0.04270	0.04098	0.03530	0.03940	0.04077
2021-11-20	0.02900	0.03542	0.04088	0.04044	0.03548	0.04080	0.04000	0.03427	0.03785	0.03953
2021-11-21	0.02882	0.03608	0.03995	0.04150	0.03603	0.03993	0.04086	0.03403	0.03728	0.03920
2021-11-22	0.02966	0.03513	0.03941	0.04028	0.03517	0.03939	0.03976	0.03318	0.03620	0.03814
2021-11-23	0.03000	0.03684	0.04235	0.04204	0.03666	0.04203	0.04124	0.03435	0.03764	0.03960
2021-11-24	0.02917	0.03610	0.04057	0.04146	0.03599	0.04043	0.04070	0.03351	0.03649	0.03858
2021-11-25	0.02871	0.03436	0.03885	0.03931	0.03446	0.03887	0.03884	0.03248	0.03520	0.03725
2021-11-26	0.03282	0.03506	0.03819	0.04048	0.03504	0.03827	0.03980	0.03236	0.03508	0.03712
2021-11-27	0.03227	0.04061	0.04661	0.04680	0.04024	0.04654	0.04558	0.03965	0.04441	0.04671
2021-11-28	0.03301	0.03864	0.04379	0.04481	0.03851	0.04393	0.04387	0.03828	0.04214	0.04509
2021-11-29	0.03307	0.04071	0.04346	0.04785	0.04035	0.04352	0.04653	0.03893	0.04238	0.04592
2021-11-30	0.03315	0.03932	0.04127	0.04613	0.03912	0.04145	0.04504	0.03770	0.04039	0.04437
2021-12-01	0.03317	0.03819	0.04066	0.04463	0.03809	0.04079	0.04372	0.03666	0.03890	0.04307
2021-12-02	0.03237	0.03601	0.03831	0.04198	0.03619	0.03865	0.04145	0.03542	0.03723	0.04150
2021-12-03	0.03355	0.03492	0.03773	0.04049	0.03518	0.03805	0.04013	0.03443	0.03604	0.04023
2021-12-04	0.03654	0.03747	0.04203	0.04321	0.03744	0.04198	0.04245	0.03642	0.03878	0.04271
2021-12-05	0.03661	0.04313	0.05022	0.04949	0.04270	0.04979	0.04815	0.04323	0.04707	0.05112
2021-12-06	0.03692	0.04090	0.04722	0.04662	0.04074	0.04685	0.04570	0.04167	0.04442	0.04922
2021-12-07	0.03635	0.04044	0.04511	0.04640	0.04029	0.04484	0.04546	0.04058	0.04257	0.04794
2021-12-08	0.03369	0.03802	0.04211	0.04343	0.03817	0.04210	0.04292	0.03912	0.04043	0.04615
2021-12-09	0.03486	0.03593	0.03992	0.04087	0.03632	0.04007	0.04070	0.03775	0.03858	0.04445
2021-12-10	0.03468	0.03886	0.04458	0.04412	0.03891	0.04435	0.04350	0.03988	0.04156	0.04718
2021-12-11	0.03613	0.03792	0.04362	0.04279	0.03804	0.04331	0.04229	0.03875	0.03989	0.04573
2021-12-12	0.03640	0.03973	0.04315	0.04563	0.03961	0.04285	0.04475	0.03909	0.04028	0.04625
2021-12-13	0.03801	0.03909	0.04138	0.04497	0.03901	0.04121	0.04413	0.03805	0.03879	0.04493
2021-12-14	0.03859	0.04251	0.04693	0.04885	0.04208	0.04641	0.04754	0.04118	0.04316	0.04899
2021-12-15	0.03859	0.04220	0.04510	0.04911	0.04178	0.04471	0.04775	0.04027	0.04180	0.04803
2021-12-16	0.03766	0.04073	0.04273	0.04739	0.04046	0.04251	0.04625	0.03898	0.03990	0.04641
2021-12-17	0.03782	0.04029	0.04305	0.04676	0.04004	0.04271	0.04567	0.03824	0.03902	0.04551
2021-12-18	0.03688	0.04072	0.04437	0.04704	0.04037	0.04379	0.04586	0.03811	0.03892	0.04528
2021-12-19	0.03655	0.03902	0.04195	0.04521	0.03887	0.04162	0.04429	0.03687	0.03736	0.04374
2021-12-20	0.03602	0.03677	0.03971	0.04247	0.03691	0.03961	0.04194	0.03562	0.03591	0.04214
2021-12-21	0.03711	0.03471	0.03747	0.03993	0.03508	0.03760	0.03973	0.03444	0.03463	0.04061
2021-12-22	0.03649	0.03664	0.03770	0.04272	0.03677	0.03781	0.04214	0.03508	0.03577	0.04138

2021-12-23	0.03727	0.03458	0.03604	0.04021	0.03495	0.03633	0.03997	0.03393	0.03454	0.03989
2021-12-24	0.03727	0.03664	0.03654	0.04318	0.03675	0.03678	0.04254	0.03473	0.03586	0.04090
2021-12-25	0.03671	0.03471	0.03474	0.04059	0.03505	0.03513	0.04030	0.03360	0.03459	0.03941
2021-12-26	0.03285	0.03336	0.03412	0.03877	0.03382	0.03451	0.03868	0.03262	0.03362	0.03810
2021-12-27	0.03261	0.03187	0.03277	0.03689	0.03247	0.03328	0.03701	0.03163	0.03269	0.03677
2021-12-28	0.03318	0.03023	0.03167	0.03473	0.03098	0.03225	0.03509	0.03069	0.03188	0.03547
2021-12-29	0.03319	0.03410	0.03801	0.03895	0.03447	0.03835	0.03880	0.03458	0.03737	0.04041
2021-12-30	0.03337	0.03477	0.03952	0.03938	0.03500	0.03957	0.03908	0.03452	0.03714	0.04017
2021-12-31	0.03338	0.03364	0.03775	0.03817	0.03397	0.03797	0.03797	0.03350	0.03582	0.03887
2022-01-01	0.03412	0.03325	0.03794	0.03757	0.03357	0.03808	0.03738	0.03284	0.03513	0.03801
2022-01-02	0.03299	0.03410	0.03744	0.03897	0.03427	0.03761	0.03853	0.03273	0.03509	0.03789
2022-01-03	0.02929	0.03260	0.03624	0.03711	0.03292	0.03652	0.03690	0.03176	0.03402	0.03662
2022-01-04	0.02933	0.03249	0.03684	0.03675	0.03277	0.03698	0.03651	0.03132	0.03369	0.03598
2022-01-05	0.03037	0.03198	0.03678	0.03592	0.03227	0.03686	0.03572	0.03070	0.03313	0.03509
2022-01-06	0.03035	0.03497	0.04144	0.03918	0.03495	0.04121	0.03856	0.03349	0.03685	0.03862
2022-01-07	0.03088	0.03415	0.04059	0.03797	0.03420	0.04034	0.03747	0.03268	0.03572	0.03749
2022-01-08	0.02929	0.03541	0.04267	0.03931	0.03528	0.04218	0.03857	0.03334	0.03659	0.03832
2022-01-09	0.02932	0.03347	0.04022	0.03687	0.03359	0.03995	0.03648	0.03229	0.03522	0.03695
2022-01-10	0.02783	0.03178	0.03795	0.03485	0.03210	0.03793	0.03471	0.03131	0.03404	0.03564
2022-01-11	0.02800	0.03004	0.03602	0.03271	0.03055	0.03621	0.03284	0.03038	0.03303	0.03438
2022-01-12	0.02580	0.03030	0.03526	0.03325	0.03071	0.03551	0.03321	0.02997	0.03269	0.03381
2022-01-13	0.02534	0.03136	0.03503	0.03477	0.03160	0.03529	0.03447	0.03007	0.03291	0.03386
2022-01-14	0.02537	0.03203	0.03678	0.03547	0.03214	0.03686	0.03500	0.03025	0.03347	0.03414
2022-01-15	0.02505	0.03084	0.03521	0.03416	0.03107	0.03546	0.03384	0.02944	0.03260	0.03304
2022-01-16	0.02450	0.02913	0.03344	0.03197	0.02956	0.03387	0.03193	0.02862	0.03179	0.03190
2022-01-17	0.02450	0.02766	0.03216	0.03009	0.02823	0.03268	0.03027	0.02786	0.03110	0.03080
2022-01-18	0.02454	0.02804	0.03337	0.03034	0.02848	0.03371	0.03038	0.02776	0.03131	0.03061
2022-01-19	0.02461	0.02653	0.03187	0.02844	0.02712	0.03233	0.02871	0.02705	0.03068	0.02957
2022-01-20	0.02327	0.02662	0.03261	0.02838	0.02711	0.03293	0.02855	0.02679	0.03068	0.02914
2022-01-21	0.03002	0.02750	0.03429	0.02921	0.02782	0.03440	0.02917	0.02710	0.03129	0.02945
2022-01-22	0.02869	0.03577	0.04590	0.03863	0.03604	0.04645	0.03841	0.04043	0.04695	0.04693
2022-01-23	0.02982	0.03814	0.04879	0.04111	0.03812	0.04870	0.04051	0.04149	0.04701	0.04808
2022-01-24	0.03010	0.03843	0.04681	0.04215	0.03834	0.04674	0.04137	0.04066	0.04519	0.04725
2022-01-25	0.03010	0.03737	0.04426	0.04107	0.03738	0.04431	0.04040	0.03936	0.04289	0.04569
2022-01-26	0.03008	0.03573	0.04168	0.03923	0.03592	0.04190	0.03879	0.03801	0.04075	0.04405
2022-01-27	0.02856	0.03392	0.03966	0.03710	0.03431	0.04000	0.03693	0.03671	0.03888	0.04245
2022-01-28	0.02873	0.03267	0.03774	0.03572	0.03316	0.03822	0.03568	0.03552	0.03727	0.04098
2022-01-29	0.02866	0.03224	0.03648	0.03538	0.03273	0.03702	0.03531	0.03458	0.03607	0.03980
2022-01-30	0.02857	0.03150	0.03517	0.03457	0.03202	0.03578	0.03452	0.03359	0.03492	0.03853
2022-01-31	0.02793	0.03025	0.03425	0.03301	0.03086	0.03489	0.03312	0.03257	0.03386	0.03721
2022-02-01	0.02802	0.02982	0.03334	0.03262	0.03041	0.03401	0.03270	0.03175	0.03309	0.03614
2022-02-02	0.02895	0.02872	0.03211	0.03124	0.02939	0.03285	0.03144	0.03083	0.03223	0.03490
2022-02-03	0.02913	0.03131	0.03648	0.03411	0.03166	0.03689	0.03390	0.03267	0.03509	0.03734
2022-02-04	0.03466	0.02986	0.03478	0.03252	0.03035	0.03534	0.03249	0.03167	0.03395	0.03604
2022-02-05	0.03465	0.03888	0.03919	0.04489	0.03938	0.03959	0.04504	0.04229	0.04693	0.04978
2022-02-06	0.03427	0.03717	0.03722	0.04253	0.03782	0.03772	0.04292	0.04078	0.04428	0.04793
2022-02-07	0.03479	0.03697	0.03639	0.04255	0.03755	0.03692	0.04282	0.03972	0.04243	0.04670

Appendix B. Study 2

B1. Sentiment data aggregation for Bitcoin data as an example

Date	Negative (#)	Neutral (#)	Positive (#)	Sentiment score
2021-01-01	910	3197	847	-0.012716996
2021-01-02	810	3175	858	0.009911212
2021-01-03	769	3145	1010	0.048943948
2021-01-04	794	3333	820	0.005255711
2021-01-05	692	3376	875	0.037022051
2021-01-06	976	3033	923	-0.010746148
2021-01-07	787	3241	916	0.026092233
2021-01-08	762	3389	800	0.007675217
2021-01-09	752	3344	863	0.022383545
2021-01-10	968	3166	803	-0.033421106
2021-01-11	1234	2972	736	-0.100768919
2021-01-12	990	3172	793	-0.03975782
2021-01-13	1031	3202	713	-0.064294379
2021-01-14	939	3156	869	-0.014101531
2021-01-15	863	3314	765	-0.019830028
2021-01-16	839	3344	770	-0.013930951
2021-01-17	873	3284	795	-0.015751212
2021-01-18	916	3289	747	-0.034127625
2021-01-19	519	3960	485	-0.006849315
2021-01-20	985	3092	879	-0.021388216
2021-01-21	997	3249	705	-0.058977984
2021-01-22	844	3283	825	-0.003836834
2021-01-23	825	3331	805	-0.004031445
2021-01-24	774	3391	782	0.001617142
2021-01-25	842	3361	757	-0.017137097
2021-01-26	742	3362	844	0.02061439
2021-01-27	916	3259	771	-0.029316619
2021-01-28	755	3279	886	0.026626016
2021-01-29	809	3293	853	0.008879919
2021-01-30	724	3390	841	0.023612513
2021-01-31	731	3428	810	0.015898571
2021-02-01	681	3302	978	0.059866962
2021-02-02	661	3399	905	0.049144008
2021-02-03	605	3376	968	0.073348151
2021-02-04	645	3347	965	0.064555175
2021-02-05	975	3053	925	-0.010094892
2021-02-06	691	3313	960	0.054190169
2021-02-07	312	1370	444	0.062088429
2021-02-08	677	3277	966	0.058739837
2021-02-09	786	3320	854	0.013709677
2021-02-10	832	3343	778	-0.010902483
2021-02-11	829	3239	874	0.009105625
2021-02-12	780	3254	903	0.024913915
2021-02-13	856	3338	774	-0.016505636
2021-02-14	825	3246	876	0.010309278
2021-02-15	811	3225	928	0.023569702
2021-02-16	697	3251	995	0.060287275
2021-02-17	780	3209	969	0.03812021
2021-02-18	873	3276	805	-0.013726282
2021-02-19	756	3127	1066	0.062638917
2021-02-20	882	3254	802	-0.016200891
2021-02-21	813	3371	772	-0.008272801
2021-02-22	1026	3159	758	-0.054218086
2021-02-23	742	2910	1308	0.114112903
2021-02-24	685	2789	1489	0.161998791
2021-02-25	651	2569	838	0.046081814
2021-02-26	855	3196	915	0.012082159
2021-02-27	931	3253	768	-0.032915994
2021-02-28	769	3304	878	0.022015754
2021-03-01	818	3180	953	0.027267219
2021-03-02	660	3056	1245	0.117919774
2021-03-03	705	3199	1056	0.070766129
2021-03-04	765	3287	914	0.030004027

2021-03-05	730	3358	862	0.026666667
2021-03-06	728	3241	972	0.049382716
2021-03-07	705	3298	960	0.051380214
2021-03-08	706	3293	953	0.049878837
2021-03-09	870	3192	889	0.003837609
2021-03-10	838	3274	836	-0.000404204
2021-03-11	765	3210	971	0.041649818
2021-03-12	865	3191	906	0.008262797
2021-03-13	727	3198	1029	0.06096084
2021-03-14	838	3316	797	-0.008281155
2021-03-15	710	3320	929	0.044162129
2021-03-16	724	3288	943	0.04419778
2021-03-17	664	3153	1142	0.096390401
2021-03-18	667	3444	855	0.037857431
2021-03-19	697	3146	1098	0.08115766
2021-03-20	752	3281	900	0.030002027
2021-03-21	698	3312	961	0.05290686
2021-03-22	826	3258	877	0.010280185
2021-03-23	739	3262	939	0.04048583
2021-03-24	843	3199	912	0.013928139
2021-03-25	700	3092	1170	0.094719871
2021-03-26	736	2959	1260	0.105751766
2021-03-27	654	3240	1070	0.083803384
2021-03-28	706	3151	1103	0.080040323
2021-03-29	558	2853	1569	0.203012048
2021-03-30	553	3128	1293	0.148773623
2021-03-31	653	3189	1111	0.092469211
2021-04-01	600	3296	1040	0.089141005
2021-04-02	683	3185	1070	0.07837181
2021-04-03	762	3370	815	0.010713564
2021-04-04	569	3279	1115	0.110014104
2021-04-05	614	3340	986	0.075303644
2021-04-06	708	3313	912	0.041354146
2021-04-07	831	3240	902	0.014277096
2021-04-08	746	3252	964	0.043933898
2021-04-09	682	3350	931	0.050171267
2021-04-10	718	3276	966	0.05
2021-04-11	602	3399	947	0.069725141
2021-04-12	623	3317	1022	0.080411125
2021-04-13	728	3255	966	0.048090523
2021-04-14	723	3282	938	0.043495853
2021-04-15	779	3232	951	0.034663442
2021-04-16	764	3313	872	0.02182259
2021-04-17	699	3395	870	0.034448026
2021-04-18	912	3159	885	-0.005447942
2021-04-19	844	3295	817	-0.005447942
2021-04-20	685	3208	1076	0.078687865
2021-04-21	1029	3123	792	-0.047936893
2021-04-22	994	3230	720	-0.055420712
2021-04-23	905	3159	875	-0.006074104
2021-04-24	783	3269	893	0.022244692
2021-04-25	927	3317	696	-0.046761134
2021-04-26	746	3392	822	0.015322581
2021-04-27	706	3325	909	0.041093117
2021-04-28	722	3267	970	0.050010083
2021-04-29	738	3266	947	0.042213694
2021-04-30	578	3296	1077	0.10078772
2021-05-01	917	3142	892	-0.005049485
2021-05-02	773	3252	935	0.03266129
2021-05-03	637	3359	966	0.06630391
2021-05-04	715	3155	1079	0.073550212
2021-05-05	619	3255	1070	0.091221683
2021-05-06	101	535	172	0.087871287
2021-05-07	728	3346	892	0.033024567
2021-05-08	832	3311	826	-0.001207486
2021-05-09	317	1517	351	0.015560641
2021-05-10	834	3370	736	-0.019838057
2021-05-11	694	3392	867	0.034928326
2021-05-12	1538	3014	385	-0.233542637
2021-05-13	1452	2867	621	-0.168218623

2021-05-14	1011	3205	723	-0.058311399
2021-05-15	996	3186	758	-0.048178138
2021-05-16	153	304	58	-0.184466019
2021-05-17	1197	3103	638	-0.113203726
2021-05-18	1024	3154	771	-0.051121439
2021-05-19	1053	2827	1068	0.003031528
2021-05-20	938	3220	794	-0.02907916
2021-05-21	845	3128	994	0.029997987
2021-05-22	985	3036	930	-0.011108867
2021-05-23	1047	3167	740	-0.061970125
2021-05-24	965	3234	753	-0.042810985
2021-05-25	881	3184	875	-0.001214575
2021-05-26	721	3285	953	0.046783626
2021-05-27	791	3199	954	0.032969256
2021-05-28	990	3127	840	-0.030260238
2021-05-29	815	3263	865	0.010115315
2021-05-30	804	3237	917	0.022791448
2021-05-31	758	3240	959	0.040548719
2021-06-01	763	3234	949	0.037606146
2021-06-02	704	3242	1008	0.061364554
2021-06-03	696	3279	981	0.057506053
2021-06-04	1101	2903	952	-0.030064568
2021-06-05	1247	2899	802	-0.089935327
2021-06-06	997	3097	861	-0.027447023
2021-06-07	1278	3020	653	-0.126237124
2021-06-08	1062	3089	806	-0.05164414
2021-06-09	1334	2787	819	-0.104251012
2021-06-10	932	2997	1021	0.017979798
2021-06-11	1003	3079	862	-0.028519417
2021-06-12	597	2519	1850	0.252315747
2021-06-13	874	3160	920	0.009285426
2021-06-14	763	3212	979	0.04360113
2021-06-15	1491	2685	764	-0.147165992
2021-06-16	786	3255	916	0.02622554
2021-06-17	727	2831	1408	0.137132501
2021-06-18	845	3106	1008	0.03286953
2021-06-19	840	3038	1090	0.050322061
2021-06-20	902	3080	965	0.012734991
2021-06-21	1177	3083	687	-0.099049929
2021-06-22	988	3171	784	-0.041270484
2021-06-23	838	3044	1071	0.047042197
2021-06-24	819	3171	964	0.029269277
2021-06-25	912	3135	910	-0.00040347
2021-06-26	950	3119	895	-0.011079774
2021-06-27	706	3217	1025	0.064470493
2021-06-28	743	3196	996	0.051266464
2021-06-29	630	2407	809	0.046541862
2021-06-30	730	3048	1178	0.09039548
2021-07-01	819	3110	1024	0.041389057
2021-07-02	829	3116	1012	0.03691749
2021-07-03	813	3186	953	0.028271405
2021-07-04	668	3077	1222	0.111536139
2021-07-05	798	3073	1079	0.056767677
2021-07-06	835	2998	1124	0.058301392
2021-07-07	715	3002	1245	0.106811769
2021-07-08	741	3211	997	0.051727622
2021-07-09	798	3266	886	0.017777778
2021-07-10	767	3321	878	0.022351994
2021-07-11	722	3358	873	0.030486574
2021-07-12	354	1585	860	0.18077885
2021-07-13	873	3188	895	0.004439064
2021-07-14	859	3281	807	-0.010511421
2021-07-15	866	3324	759	-0.021620529
2021-07-16	889	3292	780	-0.021971377
2021-07-17	108	519	123	0.02
2021-07-18	878	3198	889	0.002215509
2021-07-19	925	3342	703	-0.044668008
2021-07-20	953	3249	760	-0.038895607
2021-07-21	788	3276	905	0.023545985
2021-07-22	717	3335	904	0.037732042

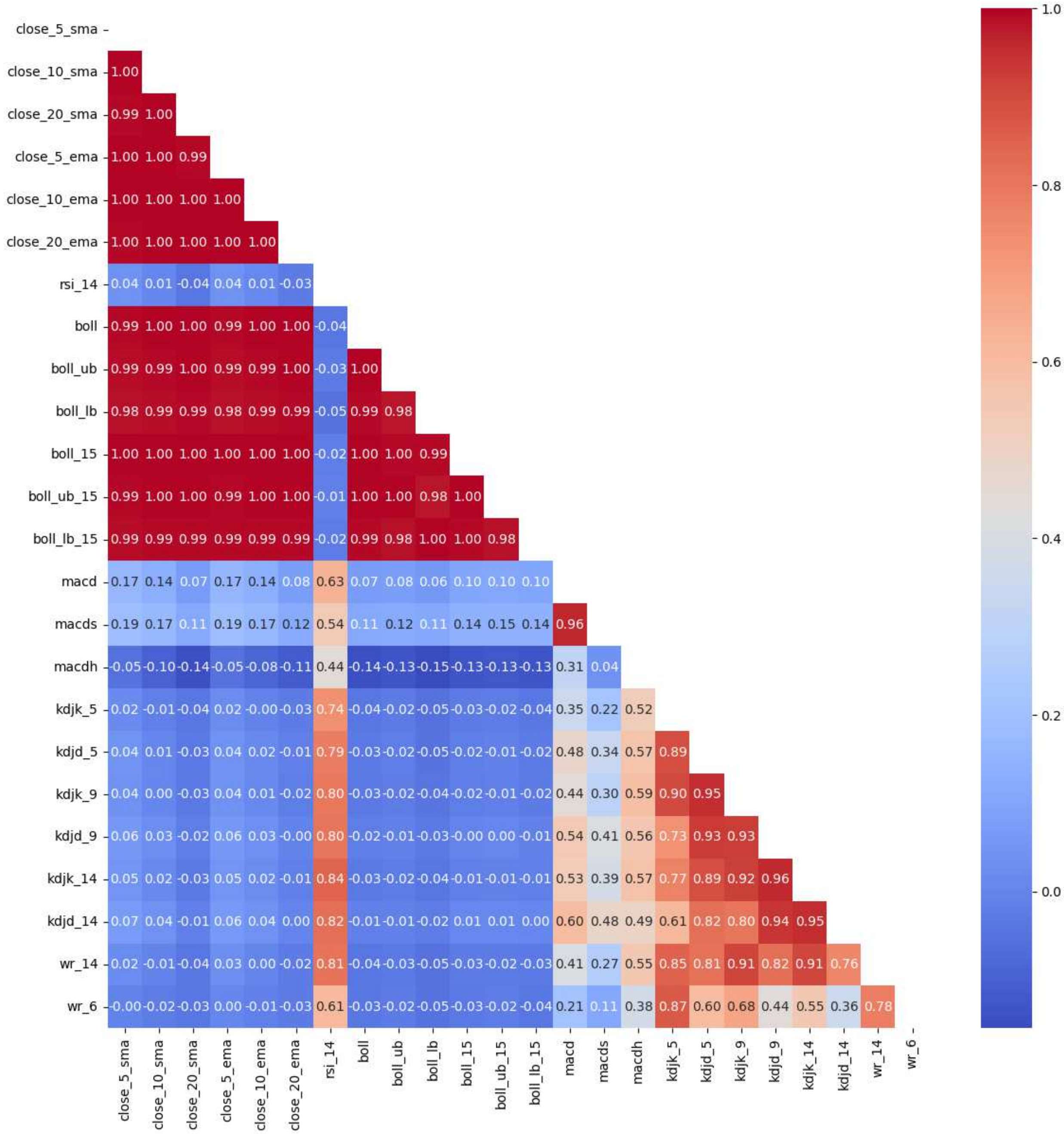
2021-07-23	293	1831	541	0.093058161
2021-07-24	602	3445	908	0.061755802
2021-07-25	713	3430	811	0.019781994
2021-07-26	905	3261	787	-0.023823945
2021-07-27	740	3323	879	0.028126265
2021-07-28	639	3306	1015	0.075806452
2021-07-29	642	3485	816	0.035201295
2021-07-30	652	3375	910	0.052258457
2021-07-31	623	3440	885	0.052950687
2021-08-01	705	3352	901	0.039532069
2021-08-02	731	3410	829	0.01971831
2021-08-03	647	3400	910	0.053056284
2021-08-04	744	3435	776	0.006458123
2021-08-05	628	3434	871	0.049260085
2021-08-06	733	3392	818	0.017196035
2021-08-07	343	1721	508	0.064152411
2021-08-08	689	3497	771	0.016542263
2021-08-09	849	3125	922	0.014910131
2021-08-10	686	3315	946	0.052557105
2021-08-11	616	3362	976	0.072668551
2021-08-12	665	3419	872	0.041767554
2021-08-13	636	3327	982	0.069969666
2021-08-14	660	3481	800	0.028334345
2021-08-15	636	3283	1022	0.078121838
2021-08-16	442	1997	516	0.025042301
2021-08-17	748	3344	863	0.02320888
2021-08-18	661	3386	901	0.048504446
2021-08-19	616	3388	943	0.066100667
2021-08-20	599	3416	936	0.068067057
2021-08-21	605	3343	1003	0.0803878
2021-08-22	547	3424	972	0.085980174
2021-08-23	590	3409	963	0.075171302
2021-08-24	694	3336	915	0.044691608
2021-08-25	636	3335	979	0.069292929
2021-08-26	703	3268	986	0.057090982
2021-08-27	661	3349	956	0.059403947
2021-08-28	678	3299	975	0.059975767
2021-08-29	667	3316	978	0.062688974
2021-08-30	820	3382	762	-0.011684126
2021-08-31	700	3317	945	0.049375252
2021-09-01	677	3384	899	0.044758065
2021-09-02	620	3063	1282	0.133333333
2021-09-03	672	3350	921	0.050374267
2021-09-04	638	3349	968	0.066599395
2021-09-05	652	3306	994	0.069063005
2021-09-06	611	3286	1048	0.088372093
2021-09-07	1056	3106	790	-0.05371567
2021-09-08	768	3370	809	0.008287851
2021-09-09	800	3285	861	0.012333199
2021-09-10	970	3166	808	-0.03276699
2021-09-11	840	3255	867	0.005441354
2021-09-12	687	3257	1022	0.067458719
2021-09-13	712	3397	851	0.028024194
2021-09-14	739	3305	910	0.034517562
2021-09-15	848	3253	855	0.001412429
2021-09-16	657	3368	935	0.056048387
2021-09-17	847	3229	857	0.002027164
2021-09-18	698	3364	902	0.04109589
2021-09-19	743	3368	839	0.019393939
2021-09-20	948	3298	710	-0.048022599
2021-09-21	908	3331	706	-0.040849343
2021-09-22	659	3367	936	0.055824264
2021-09-23	708	3256	997	0.058254384
2021-09-24	1023	3091	838	-0.037358643
2021-09-25	827	3230	889	0.012535382
2021-09-26	696	3226	1040	0.069326884
2021-09-27	723	3374	867	0.029008864
2021-09-28	779	3264	903	0.025070764
2021-09-29	753	3262	945	0.038709677
2021-09-30	606	3239	1106	0.100989699

2021-10-01	634	2651	878	0.058611578
2021-10-02	672	3243	1045	0.075201613
2021-10-03	705	3342	906	0.040581466
2021-10-04	625	3271	1039	0.083890578
2021-10-05	606	3345	1002	0.079951545
2021-10-06	669	3304	976	0.062032734
2021-10-07	665	3339	943	0.056195674
2021-10-08	729	3261	973	0.049163812
2021-10-09	647	3463	858	0.04247182
2021-10-10	742	3341	875	0.026825333
2021-10-11	767	3347	828	0.012343181
2021-10-12	739	3306	915	0.035483871
2021-10-13	675	3367	898	0.0451417
2021-10-14	671	3358	923	0.05088853
2021-10-15	647	3299	1010	0.073244552
2021-10-16	690	3323	944	0.05124067
2021-10-17	664	3333	941	0.056095585
2021-10-18	704	3304	943	0.048273076
2021-10-19	671	3308	977	0.061743341
2021-10-20	631	3269	1053	0.085200888
2021-10-21	738	3281	936	0.039959637
2021-10-22	662	3365	928	0.053683148
2021-10-23	718	3318	900	0.036871961
2021-10-24	704	3306	947	0.049021586
2021-10-25	610	3434	897	0.058085408
2021-10-26	810	3333	816	0.001209921
2021-10-27	769	3328	863	0.018951613
2021-10-28	659	3355	937	0.056150273
2021-10-29	602	3415	937	0.067622124
2021-10-30	633	3392	950	0.063718593
2021-10-31	628	3250	1080	0.091165793
2021-11-01	565	3145	1240	0.136363636
2021-11-02	622	3256	1081	0.092558984
2021-11-03	474	2467	684	0.057931034
2021-11-04	710	3327	915	0.041397415
2021-11-05	643	3363	948	0.061566411
2021-11-06	609	3314	1034	0.085737341
2021-11-07	571	3415	981	0.082544796
2021-11-08	456	3048	1432	0.197730956
2021-11-09	662	3336	948	0.057824505
2021-11-10	992	3181	777	-0.043434343
2021-11-11	646	3443	861	0.043434343
2021-11-12	737	3280	949	0.042690294
2021-11-13	641	3131	1187	0.110102843
2021-11-14	572	3459	930	0.07216287
2021-11-15	785	3316	850	0.013128661
2021-11-16	789	3337	833	0.008872757
2021-11-17	757	3336	863	0.021388216
2021-11-18	840	3319	766	-0.015025381
2021-11-19	668	3362	902	0.047445255
2021-11-20	664	3318	966	0.061034762
2021-11-21	704	3362	893	0.038112523
2021-11-22	692	3327	926	0.047320526
2021-11-23	737	3292	924	0.037754896
2021-11-24	658	3328	951	0.059347782
2021-11-25	636	3323	999	0.073215006
2021-11-26	896	3220	827	-0.013959134
2021-11-27	836	3325	808	-0.005634937
2021-11-28	670	3344	936	0.053737374
2021-11-29	629	3477	849	0.044399596
2021-11-30	718	3323	906	0.03800283
2021-12-01	726	3382	860	0.026972625
2021-12-02	715	3263	974	0.0523021
2021-12-03	837	3347	773	-0.012911035
2021-12-04	880	3250	821	-0.011916784
2021-12-05	821	3334	796	-0.005049485
2021-12-06	733	3421	782	0.009927066
2021-12-07	840	3433	667	-0.035020243
2021-12-08	253	846	173	-0.062893082
2021-12-09	925	3293	664	-0.053461696

2021-12-10	508	1925	357	-0.054121864
2021-12-11	829	3478	648	-0.036528759
2021-12-12	851	3436	668	-0.036932392
2021-12-13	886	3394	629	-0.052352821
2021-12-14	861	3421	658	-0.041093117
2021-12-15	670	3614	666	-0.000808081
2021-12-16	875	3435	645	-0.04641776
2021-12-17	889	3438	615	-0.05544314
2021-12-18	771	3470	725	-0.009262988
2021-12-19	757	3582	622	-0.027212256
2021-12-20	807	3484	660	-0.029690972
2021-12-21	748	3438	748	0
2021-12-22	829	3408	701	-0.025921426
2021-12-23	714	3356	876	0.03275374
2021-12-24	692	3390	858	0.033603239
2021-12-25	643	3410	908	0.05341665
2021-12-26	746	3372	784	0.007751938
2021-12-27	752	3407	804	0.010477534
2021-12-28	238	757	145	-0.081578947
2021-12-29	897	3428	632	-0.053459754
2021-12-30	842	3367	749	-0.018757564
2021-12-31	813	3120	1015	0.040824576

Appendix C. Study 3

C1. Correlation heatmap of technical indicators – Bitcoin data as an example



C2. Detailed results in comparison with other methods

Out-of-sample accuracies achieved by ANN model (in place of the GRU model) using various feature sets (average and standard deviations of 10 repeated runs)

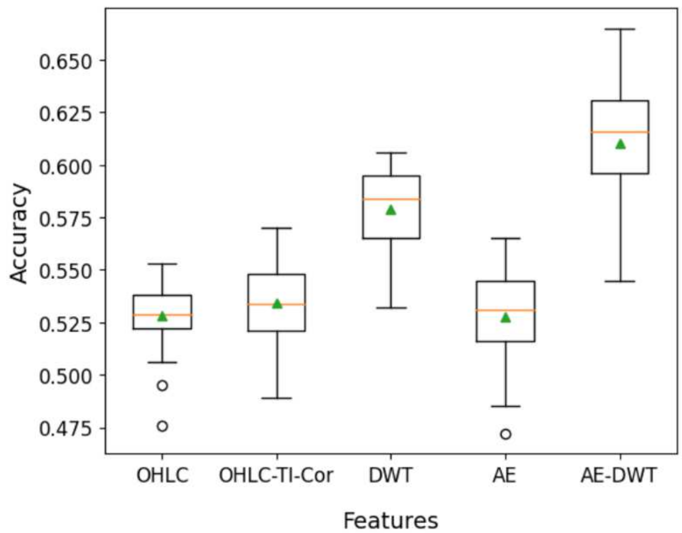
	OHLC	OHLC-TI-Cor	DWT	AE	AE-DWT
BinanceCoin	0.532 (0.024)	0.515 (0.011)	0.599 (0.016)	0.528 (0.012)	**0.625 (0.017)**
Bitcoin	0.522 (0.025)	0.545 (0.015)	0.567 (0.016)	0.557 (0.008)	**0.623 (0.009)**
BitcoinCash	0.522 (0.022)	0.534 (0.014)	0.572 (0.011)	0.504 (0.009)	**0.625 (0.007)**
Cardano	0.529 (0.014)	0.546 (0.017)	0.602 (0.014)	0.545 (0.016)	**0.631 (0.015)**
Chainlink	0.476 (0.014)	0.489 (0.013)	0.582 (0.017)	0.472 (0.012)	**0.589 (0.062)**
Dash	0.525 (0.012)	0.518 (0.025)	0.550 (0.022)	0.523 (0.023)	**0.550 (0.027)**
Decentraland	0.522 (0.013)	0.523 (0.013)	0.593 (0.011)	0.517 (0.019)	**0.606 (0.008)**
EOS	0.552 (0.021)	0.551 (0.008)	0.590 (0.017)	0.535 (0.011)	**0.663 (0.011)**
EnjinCoin	0.511 (0.015)	0.533 (0.017)	0.543 (0.015)	0.520 (0.014)	**0.561 (0.039)**
Ethereum	0.506 (0.019)	0.504 (0.021)	0.557 (0.009)	0.502 (0.016)	**0.597 (0.022)**
EthereumClassic	0.547 (0.018)	0.545 (0.014)	0.580 (0.016)	0.542 (0.017)	**0.648 (0.011)**
Filecoin	0.524 (0.009)	0.545 (0.011)	0.564 (0.016)	0.536 (0.013)	**0.625 (0.017)**
IOTA	0.538 (0.012)	0.553 (0.016)	0.595 (0.012)	0.535 (0.006)	**0.579 (0.050)**
Litecoin	0.514 (0.028)	0.526 (0.012)	0.573 (0.011)	0.556 (0.010)	**0.616 (0.015)**
Maker	0.547 (0.014)	0.561 (0.008)	0.601 (0.009)	0.548 (0.011)	**0.608 (0.014)**
Monero	0.551 (0.010)	0.548 (0.013)	0.584 (0.016)	0.553 (0.013)	**0.596 (0.010)**
NEO	0.549 (0.012)	0.558 (0.003)	0.594 (0.013)	0.565 (0.011)	**0.612 (0.015)**
OMG_Network	0.523 (0.015)	0.527 (0.024)	0.606 (0.015)	0.524 (0.009)	**0.639 (0.013)**
Stellar	0.525 (0.024)	0.534 (0.017)	0.601 (0.016)	0.531 (0.013)	**0.651 (0.048)**
Tezos	0.533 (0.013)	0.570 (0.014)	0.604 (0.014)	0.533 (0.012)	**0.665 (0.007)**
Theta	0.533 (0.021)	0.537 (0.014)	0.586 (0.009)	0.516 (0.012)	**0.626 (0.020)**
Tron	0.538 (0.007)	0.511 (0.013)	0.532 (0.013)	0.509 (0.010)	**0.545 (0.029)**
Waves	0.536 (0.010)	0.521 (0.025)	0.587 (0.013)	0.513 (0.016)	**0.602 (0.018)**
XRP	0.553 (0.016)	0.552 (0.006)	0.565 (0.012)	0.548 (0.011)	**0.635 (0.041)**
Zcash	0.495 (0.019)	0.509 (0.015)	0.541 (0.016)	0.485 (0.018)	**0.548 (0.035)**
Average	0.528 (0.018)	0.534 (0.019)	0.579 (0.021)	0.528 (0.022)	**0.611 (0.033)**

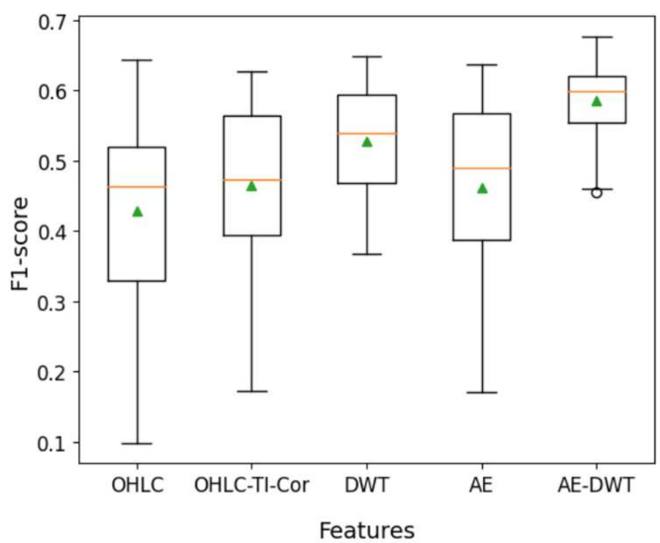
Out-of-sample f1-scores achieved by ANN model (in place of the GRU model) using various feature sets (average and standard deviations of 10 repeated runs)

	OHLC	OHLC-TI-Cor	DWT	AE	AE-DWT
BinanceCoin	0.463 (0.136)	0.362 (0.106)	0.539 (0.069)	0.388 (0.117)	**0.599 (0.027)**
Bitcoin	0.385 (0.108)	0.323 (0.136)	0.368 (0.073)	0.384 (0.068)	**0.520 (0.049)**
BitcoinCash	0.486 (0.062)	0.510 (0.047)	0.532 (0.041)	0.514 (0.038)	**0.610 (0.020)**
Cardano	0.380 (0.102)	0.473 (0.079)	0.527 (0.071)	0.430 (0.056)	**0.577 (0.040)**
Chainlink	0.415 (0.076)	0.536 (0.084)	0.609 (0.054)	0.322 (0.122)	**0.636 (0.041)**
Dash	0.644 (0.024)	0.594 (0.083)	0.598 (0.043)	0.569 (0.195)	**0.610 (0.024)**
Decentraland	0.228 (0.180)	0.343 (0.107)	0.517 (0.056)	0.350 (0.077)	**0.552 (0.022)**
EOS	0.237 (0.115)	0.385 (0.059)	0.438 (0.053)	0.246 (0.079)	**0.596 (0.020)**
EnjinCoin	0.295 (0.171)	0.412 (0.157)	0.385 (0.137)	0.490 (0.126)	**0.501 (0.172)**
Ethereum	0.474 (0.080)	0.403 (0.107)	0.542 (0.042)	0.515 (0.054)	**0.574 (0.049)**
EthereumClassic	0.329 (0.060)	0.394 (0.067)	0.494 (0.074)	0.423 (0.059)	**0.621 (0.034)**
Filecoin	0.189 (0.119)	0.397 (0.092)	0.447 (0.066)	0.423 (0.057)	**0.555 (0.066)**
IOTA	0.632 (0.030)	0.615 (0.058)	0.613 (0.034)	0.637 (0.015)	**0.657 (0.021)**
Litecoin	0.460 (0.056)	0.554 (0.035)	0.601 (0.022)	0.558 (0.025)	**0.615 (0.025)**
Maker	0.145 (0.089)	0.173 (0.042)	0.421 (0.041)	0.171 (0.086)	**0.455 (0.045)**
Monero	0.605 (0.054)	0.628 (0.017)	0.649 (0.025)	0.594 (0.018)	**0.650 (0.023)**
NEO	0.640 (0.013)	0.625 (0.013)	0.620 (0.026)	0.638 (0.017)	**0.620 (0.029)**
OMG_Network	0.481 (0.076)	0.584 (0.051)	0.595 (0.016)	0.568 (0.030)	**0.636 (0.033)**
Stellar	0.547 (0.068)	0.561 (0.040)	0.565 (0.022)	0.565 (0.028)	**0.62 0(0.125)**

Tezos	0.517 (0.062)	0.572 (0.047)	0.595 (0.025)	0.562 (0.025)	**0.677 (0.008)**
Theta	0.517 (0.056)	0.439 (0.093)	0.542 (0.046)	0.391 (0.163)	**0.599 (0.066)**
Tron	0.628 (0.019)	0.565 (0.062)	0.569 (0.037)	0.576 (0.056)	**0.589 (0.040)**
Waves	0.098 (0.108)	0.303 (0.184)	0.469 (0.041)	0.214 (0.127)	**0.492 (0.043)**
XRP	0.520 (0.022)	0.497 (0.039)	0.530 (0.040)	0.567 (0.024)	**0.611 (0.046)**
Zcash	0.418 (0.173)	0.396 (0.139)	0.444 (0.111)	0.453 (0.086)	**0.460 (0.147)**
Average	0.429 (0.156)	0.466 (0.116)	0.528 (0.076)	0.462 (0.128)	**0.585 (0.058)**

Boxplots of out-of-sample accuracies and F1-scores achieved by the ANN model (in place of the GRU) using various feature sets





Out-of-sample accuracies achieved by GRU model using various feature sets – PCA in place of the Autoencoder (average and standard deviations of 10 repeated runs)

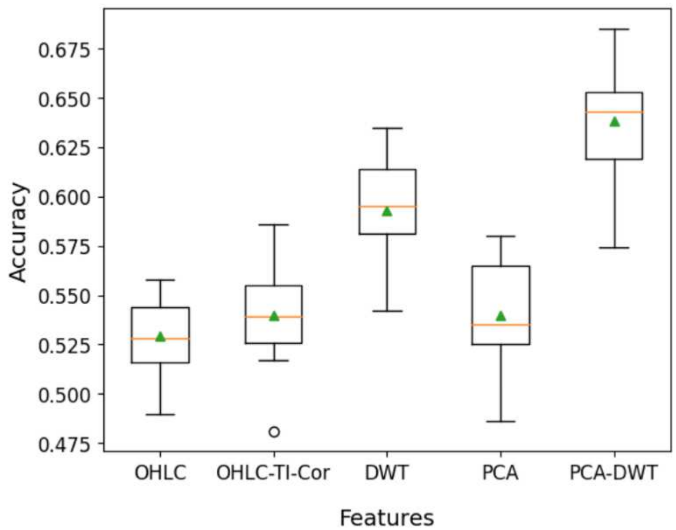
	OHLC	OHLC-TI-Cor	DWT	PCA	PCA-DWT
BinanceCoin	0.506 (0.019)	0.506 (0.014)	0.620 (0.010)	0.494 (0.012)	**0.661 (0.010)**
Bitcoin	0.513 (0.018)	0.557 (0.018)	0.604 (0.016)	0.563 (0.025)	**0.704 (0.009)**
BitcoinCash	0.541 (0.017)	0.557 (0.010)	0.613 (0.013)	0.538 (0.016)	**0.678 (0.017)**
Cardano	0.540 (0.019)	0.528 (0.014)	0.634 (0.016)	0.534 (0.012)	**0.683 (0.011)**
Chainlink	0.495 (0.010)	0.478 (0.014)	0.643 (0.014)	0.492 (0.013)	**0.654 (0.050)**
Dash	0.509 (0.028)	0.524 (0.012)	0.619 (0.013)	0.535 (0.001)	**0.679 (0.015)**
Decentraland	0.527 (0.007)	0.524 (0.010)	0.593 (0.016)	0.529 (0.018)	**0.633 (0.020)**
EOS	0.545 (0.013)	0.566 (0.012)	0.67 (0.0120)	0.584 (0.009)	**0.726 (0.010)**
EnjinCoin	0.532 (0.026)	0.538 (0.015)	0.604 (0.017)	0.537 (0.013)	**0.691 (0.015)**
Ethereum	0.508 (0.007)	0.513 (0.018)	0.579 (0.009)	0.528 (0.016)	**0.635 (0.016)**
EthereumClassic	0.528 (0.014)	0.524 (0.009)	0.580 (0.035)	0.527 (0.011)	**0.580 (0.064)**
Filecoin	0.537 (0.016)	0.570 (0.014)	0.555 (0.014)	0.567 (0.011)	**0.621 (0.020)**
IOTA	0.541 (0.019)	0.548 (0.013)	0.622 (0.012)	0.543 (0.020)	**0.665 (0.024)**
Litecoin	0.516 (0.008)	0.514 (0.009)	0.628 (0.023)	0.556 (0.011)	**0.702 (0.022)**
Maker	0.552 (0.012)	0.558 (0.009)	0.633 (0.008)	0.542 (0.009)	**0.668 (0.013)**
Monero	0.548 (0.019)	0.541 (0.013)	0.652 (0.008)	0.536 (0.019)	**0.721 (0.007)**
NEO	0.548 (0.007)	0.551 (0.005)	0.606 (0.006)	0.550 (0.005)	**0.639 (0.009)**
OMG_Network	0.533 (0.028)	0.535 (0.011)	0.618 (0.011)	0.539 (0.007)	**0.656 (0.011)**
Stellar	0.519 (0.011)	0.530 (0.022)	0.639 (0.046)	0.514 (0.006)	**0.700 (0.064)**
Tezos	0.513 (0.012)	0.556 (0.014)	0.661 (0.015)	0.544 (0.018)	**0.709 (0.020)**
Theta	0.517 (0.026)	0.520 (0.019)	0.665 (0.007)	0.509 (0.018)	**0.738 (0.013)**
Tron	0.526 (0.010)	0.521 (0.003)	0.572 (0.046)	0.532 (0.017)	**0.585 (0.065)**
Waves	0.543 (0.003)	0.548 (0.009)	0.601 (0.044)	0.544 (0.000)	**0.619 (0.069)**
XRP	0.546 (0.011)	0.589 (0.013)	0.582 (0.013)	0.577 (0.015)	**0.682 (0.016)**
Zcash	0.523 (0.014)	0.523 (0.013)	0.574 (0.013)	0.522 (0.017)	**0.620 (0.019)**
Average	0.528 (0.016)	0.537 (0.023)	0.615 (0.013)	0.537 (0.022)	**0.666 (0.041)**

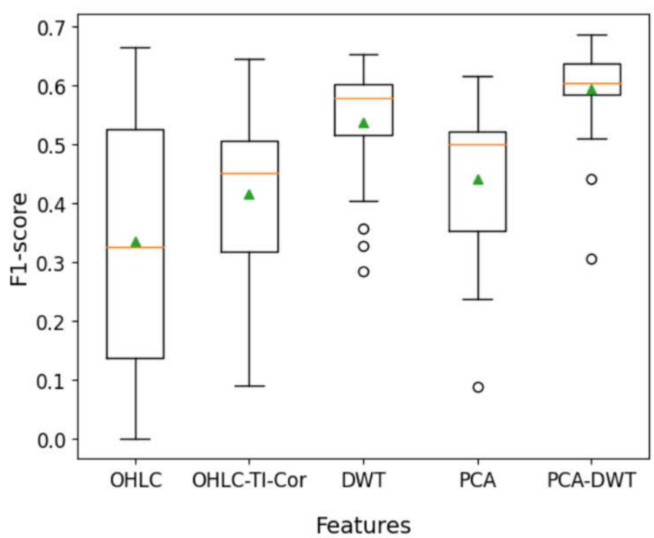
Out-of-sample f1-scores achieved by GRU model using various feature sets – PCA in place of the Autoencoder (average and standard deviations of 10 repeated runs)

	OHLC	OHLC-TI-Cor	DWT	PCA	PCA-DWT
BinanceCoin	0.450 (0.251)	0.370 (0.129)	0.575 (0.034)	0.426 (0.202)	**0.641 (0.031)**
Bitcoin	0.403 (0.167)	0.440 (0.108)	0.576 (0.030)	0.448 (0.079)	**0.671 (0.020)**
BitcoinCash	0.480 (0.085)	0.464 (0.055)	0.602 (0.030)	0.496 (0.035)	**0.675 (0.024)**
Cardano	0.420 (0.200)	0.429 (0.061)	0.617 (0.035)	0.470 (0.094)	**0.673 (0.014)**
Chainlink	0.517 (0.228)	0.422 (0.190)	0.671 (0.020)	0.360 (0.282)	**0.665 (0.029)**
Dash	0.460 (0.302)	0.643 (0.045)	0.638 (0.033)	0.659 (0.034)	**0.694 (0.015)**
Decentraland	0.073 (0.175)	0.215 (0.192)	0.449 (0.091)	0.320 (0.214)	**0.509 (0.073)**
EOS	0.356 (0.182)	0.452 (0.036)	0.634 (0.019)	0.493 (0.020)	**0.690 (0.014)**
EnjinCoin	0.388 (0.132)	0.478 (0.095)	0.584 (0.046)	0.502 (0.092)	**0.671 (0.024)**
Ethereum	0.300 (0.258)	0.382 (0.172)	0.548 (0.048)	0.501 (0.151)	**0.597 (0.065)**
EthereumClassic	0.116 (0.156)	0.134 (0.170)	0.511 (0.109)	0.173 (0.226)	**0.354 (0.281)**
Filecoin	0.114 (0.120)	0.442 (0.090)	0.333 (0.108)	0.393 (0.078)	**0.519 (0.048)**
IOTA	0.656 (0.045)	0.608 (0.053)	0.621 (0.032)	0.617 (0.051)	**0.674 (0.058)**
Litecoin	0.612 (0.066)	0.595 (0.077)	0.650 (0.024)	0.545 (0.039)	**0.705 (0.025)**
Maker	0.214 (0.204)	0.235 (0.113)	0.547 (0.032)	0.239 (0.179)	**0.622 (0.039)**
Monero	0.666 (0.029)	0.644 (0.053)	0.688 (0.010)	0.687 (0.016)	**0.743 (0.009)**
NEO	0.647 (0.011)	0.609 (0.036)	0.628 (0.012)	0.644 (0.006)	**0.651 (0.031)**
OMG_Network	0.462 (0.240)	0.480 (0.111)	0.612 (0.026)	0.588 (0.041)	**0.656 (0.022)**
Stellar	0.662 (0.056)	0.634 (0.050)	0.651 (0.026)	0.677 (0.011)	**0.716 (0.027)**
Tezos	0.592 (0.206)	0.607 (0.060)	0.670 (0.028)	0.564 (0.061)	**0.712 (0.026)**
Theta	0.513 (0.259)	0.489 (0.172)	0.653 (0.022)	0.561 (0.205)	**0.738 (0.014)**
Tron	0.682 (0.005)	0.679 (0.016)	0.652 (0.050)	0.634 (0.053)	**0.677 (0.024)**

Waves	0.314 (0.000)	0.305 (0.076)	0.373 (0.255)	0.330 (0.000)	**0.369 (0.301)**
XRP	0.500 (0.110)	0.545 (0.045)	0.508 (0.043)	0.531 (0.054)	**0.652 (0.019)**
Zcash	0.225 (0.213)	0.140 (0.173)	0.485 (0.101)	0.198 (0.179)	**0.587 (0.060)**
Average	0.433 (0.180)	0.458 (0.155)	0.579 (0.090)	0.482 (0.145)	**0.634 (0.098)**

Boxplots of out-of-sample accuracies and F1-scores achieved by the GRU model using various feature sets (PCA in place of the Autoencoder)





C3. Algorithm: **Trading Strategy Execution**

1. Set the variable "holding" to False, indicating that we are not currently holding any asset.
2. Iterate over each predicted label and actual price:
   a. Get the actual Close prices of the cryptocurrency from the OHLC data.
   b. For each predicted label and corresponding actual price, do the following:
      i. If the prediction is to buy (y=1), and we are not currently holding any cryptocurrency:
         - Buy the cryptocurrency and record the price at which the cryptocurrency is bought.
         - Set the variable "holding" to True to indicate that we are now holding the cryptocurrency.
      ii. If the prediction is to sell (y=0), and we are currently holding the cryptocurrency:
         - Calculate the return and record it in the list of returns.
         - Set the variable "holding" to False to indicate that we have sold the cryptocurrency and are no longer holding it.
3. Calculate the average of the returns.
4. Calculate the standard deviation of the returns.
5. Calculate the Sharpe ratio for daily returns by subtracting the daily risk-free rate from the mean return, and then dividing by the standard deviation of the returns.