KU LEUVEN

FACULTY OF
ENGINEERING SCIENCE

An in-depth analysis on the impact of Twitter features on bitcoin's price prediction

Aytug Altin

Thesis voorgedragen tot het behalen
van de graad van Master of Science
in de ingenieurswetenschappen:
computerwetenschappen, hoofdoptie
Artificiële intelligentie

Promotor:
Prof. Dr. Marie-Francine Moens

Assessor:
Damien Sileo, Mohit Kumar

Begeleider:
Liesbeth Allein

Academiejaar 2021 – 2022

Preface

I would like to thank everybody who kept me busy the last couple of months, especially my promoter prof. dr. Moens and my assistant Liesbeth Allein. I feel privileged to be able to conduct my thesis with these people. I am also thankful to be allowed to bring forward my thesis topic. Furthermore, I would like to thank the Twitter developers that granted me access to their developer API. I would also like to thank the jury for reading the text. My sincere gratitude also goes to my family and friends.

Aytug Altin

Contents

Abstract

Influential people such as the famous entrepreneur Elon Musk have altered the price of several cryptocurrencies by simply tweeting about them. Different celebrities, state mayors, and senators have influenced cryptocurrencies using Twitter making the platform a source of information to predict cryptocurrency-related trends.

We introduce a variety of Twitter features to forecast bitcoin's price. Moreover, we introduce weighted tweet sentiment features that reflect the overall opinion about bitcoin on Twitter, tweet volume features indicating the overall activity and interest in bitcoin on Twitter and, a weighted BERT feature vector that encapsulates the information found in tweets regarding bitcoin. The prediction tasks include predicting the absolute price in bitcoin and the price difference. In particular, we have used long short-term memory (LSTM) based models that take a group of these features as input alongside the absolute price or price difference.

As opposed to related research, we found that our Tweet sentiment features, obtained by the VADER sentiment model, did not produce an improvement in the prediction tasks for our data over our baseline models. They did, however, react to changes in bitcoin's price.

We introduce the first implementation of BERTTweet produced language vectors to represent the tweet information in bitcoin prediction tasks. Our results suggest that using these BERT vectors as an input to the prediction model seems to introduce noise into the data and even decreases the model's performance.

Furthermore, tweet volume appears to increase the correlation with bitcoin's price when bots and unavailable tweets are filtered from the dataset. It seems that removing redundant tweets improves the model in predicting upward and downward trends in bitcoin's price. Both a large increase and decrease in bitcoin's price seem to increase the volume of tweets regarding bitcoin on Twitter. This indicates that large fluctuations in bitcoin's price cause spikes in the number of tweets that are posted regarding bitcoin.

Our results show that by increasing the time intervals that the features represent, the correlation with bitcoin's price increases. Features representing larger time intervals have a higher chance of including data that has a higher correlation to bitcoin's price compared to features representing smaller time intervals.

Samenvatting

In dit onderzoek worden verschillende parameters getest op hun vermogen tot het correct voorspellen van de prijs van de crypto munteenheid bitcoin. De verschillende parameters zijn geëxtraheerd uit Twitter berichten (tweets). Elk parameter stelt een waarde voor een tijdsinterval voor. De data die we gebruiken bestaat uit alle tweets die de term “bitcoin” bevatten. Deze tweets zijn geplaatst tussen 8 maart 2018 en 3 november 2018. We introduceren het sentiment van tweets als een parameter. Het sentiment kan worden voorgesteld door één enkele sentiment parameter nl. de samengestelde sentiment score of door een combinatie van de individuele sentiment parameters nl. de positieve, neutrale en negatieve sentiment scores. Alle sentiment scores worden gewogen door het aantal volgers van de gebruiker die verantwoordelijk is voor het plaatsen van de tweet.

Verder introduceren we het aantal tweets voor een periode als parameter. Zo maken we een onderscheid tussen het aantal tweets en het aantal gefilterde tweets. De gefilterde tweet set bevat enkel tweets die op oktober 2021 nog beschikbaar zijn. Tweets gecreëerd door bots en niet-Engelstalige tweets worden hier ook uit gefilterd. Verder hebben we het aantal positieve, neutrale en negatieve tweets die ook kunnen gebruikt worden als parameters.

Een tweet kan worden voorgesteld door BERT vectoren. Een vector van een tweet representeert de inhoud van de tweet in een numerieke vector. We gebruiken het BERTTweet model dat specifiek getraind is voor het representeren van tweets. Er wordt een gewogen gemiddelde genomen van alle BERT vectoren binnen een tijdsperiode. Het gewicht van een tweet wordt bepaald door het aantal volgers dat de gebruiker die de tweet gecreëerd heeft.

Elk model is opgebouwd uit een long short-term memory (LSTM) netwerk gevolgd door een volledig verbonden (dense) laag. Een LSTM is een neuraal netwerk die ontworpen is om langdurige afhankelijkheid binnen waardes te leren. Door het definiëren van basismodellen die enkel de historische prijs van bitcoin als input gebruiken kunnen we een basislijn definiëren. In onze resultaten zien we dat de basismodellen de prijs van bitcoin op het voorgaande tijdstip gebruiken als hun voorspelling. Op deze manier zorgen ze dat de fout in de voorspelling zo klein mogelijk blijft. Dit introduceert een kloof tussen de actuele prijs en de voorspelde prijs, de voorspelling wordt een exacte kopie van de werkelijke prijs die met één tijdstap achterloopt. Om dit probleem te voorkomen, zullen de modellen als input het verschil in prijs van bitcoin nemen en dit verschil voor de volgende tijdstap proberen te voorspellen. Dit elimineert de mogelijkheid om de vorige prijs als voorspelling

te gebruiken om de foutmarge te houden. Elk model wordt geëvalueerd op het voorspellen van de absolute prijs en de relatieve verandering in prijs.

Een model toont verbetering ten opzichte van de basismodellen als de kloof in de absolute voorspelling kleiner wordt dan in de basismodellen en als de fout in het relatieve voorspelling lager is dan in de basismodellen. Verder kunnen de voorspelling grafieken van de relatieve voorspellingen eventueel interessante resultaten geven.

Door de tweet sentiment parameters toe te voegen als input parameters waren de modellen niet in staat om de voorspelling te verbeteren t.o.v. de basismodellen. De voorspellingen tonen aan dat het toevoegen van de parameters extra ruis introduceren boven op het basismodel. Als we de correlatie analyseren tussen de sentiment parameters en de prijs van bitcoin, concluderen we dat het sentiment van de tweets achterloopt op de verandering in de prijs van bitcoin in onze dataset. Dit suggereert dat de tweets slechts informatie geven over de veranderingen in de waarde van bitcoin. Grote veranderingen in de waarde van bitcoin worden weergegeven in de schommelingen van de sentiment scores, maar niet elke verandering in de sentiment score wordt weerspiegeld in de waarde van bitcoin. Het is mogelijk dat grote veranderingen in de waarde van bitcoin grote veranderingen in tweet sentiment veroorzaakt. Verder is er geen verschil in de voorspellingen tussen de modellen die gebruik maken van individuele sentiment scores en modellen die gebruik maken van samengestelde sentiment scores.

Uit onze resultaten blijkt dat de correlatie tussen de hoeveelheid gefilterde tweets en de waarde van bitcoin hoger is in vergelijking met de hoeveelheid originele tweets. Dit geeft aan dat het filteren van tweets waarschijnlijk de ruis in onze data vermindert. Grote verschuivingen in de prijs van bitcoin zijn te vinden in de veranderingen in het volume van de gefilterde tweets. Een toename in het volume van gefilterde tweets geeft echter niet de richting en grootte van de corresponderende verandering in de prijs van bitcoin in onze dataset. Maar niet alle veranderingen in volume zijn weerspiegeld in de prijs van bitcoin. Het ziet eruit dat Twitter op de veranderingen van bitcoin reageert door een toename van hoeveelheid tweets.

Verder kunnen we kijken naar het aantal positieve, neutrale en negatieve tweets. We vonden geen verschil in prestatie door het aantal positieve, neutrale en negatieve tweets in ons model. De proporties van het volume van deze tweets lijken stabiel te blijven over de test sets. Ook lijken er geen aanwijzingen te zijn dat één van de drie afzonderlijke volumes zich anders gedraagt dan de andere wanneer we de voorspelling grafieken inspecteren.

De BERT vectoren verkregen door het BERTTweet model worden voor het eerste gebruikt in een bitcoin voorspellingsmodel. Deze vector en de historische prijs van bitcoin worden gebruikt om de waarde van bitcoin te voorspellen. We maken bij dit model gebruik van een Dense-LSTM model. Onze resultaten laten geen meetbare verbetering zien ten opzichte van onze basismodellen. Meer nog, het gebruik van de BERT-vector als input voor het voorspellingsmodel lijkt ruis in de gegevens te introduceren en vermindert zelfs de prestaties van het model.

Bovendien introduceren we een trade-off: het verhogen van het tijdsinterval van de inputparameters verhoogt de correlatie tussen deze parameters en de waarde van bitcoin maar tegelijkertijd wordt onze training data kleiner waardoor het model minder data ter beschikking heeft om uit te leren.

List of Figures

List of Tables

List of Abbreviations

Abbreviations

LSTM	Long Short-Term Memory
CNN	Convolutional Neural Network
GRU	Gated Recurrent Unit
ANN	Artificial Neural Network
SVM	Support Vector Machine
RNN	Recurrent Neural Network
RMSE	Root Mean Square Error
MAE	Mean Absolute Error
RV	Realized Volatility
NLP	Natural Language Processing
USD	United States Dollar
UTC	Coordinated Universal Time

Chapter 1

Introduction

This chapter contains a general introduction of our work, the related work, our goals, and our contributions.

1.1 Introduction

1.1.1 Cryptocurrencies & Bitcoin

Cryptocurrencies are currencies generated by computer code. They are not attached to any government and are not related to any central bank [31]. These currencies can be bought with traditional currencies on various exchange platforms or certain cryptocurrencies can be exchanged with others on these exchange platforms. According to CoinMarketCap [3], there are more than 12500 different cryptocurrencies as of September 2021.

Bitcoin introduced in 2008 [45] is a peer-to-peer currency that is the most popular cryptocurrency [54] and has become more popular in the last five years. The number of active addresses for bitcoin increased more than three times in the last five years [2]. The number of active addresses for a cryptocurrency is the number of active participants in the transfer of the cryptocurrency's token. A similar increase is found on the second most popular cryptocurrency: Ethereum. Not only the number of active addresses for bitcoin and Ethereum has increased, but a similar trend is also visible for the number of unique addresses. Unique addresses are used when receiving a cryptocurrency and are only meant to be used once. In a survey described in [59], 53% of the users questioned during February and March 2019 acquired their first cryptocurrency between 2017 and 2019. Looking into google trends data [7], the term "cryptocurrency" started to have an increase in search interest in 2017 and peaked in January 2018. It has had its second-highest peak in May 2021.

1.1.2 Bitcoin as a Currency

The car company Tesla started accepting bitcoin at the end of March 2021 after buying 1.5 billion worth of bitcoin four months earlier [19]. On May 12, however, the company announced that it has stopped accepting bitcoin due to the fact that mining transactions for bitcoin are not fueled by sustainable energy sources. They claim that whenever approximately 50% of the bitcoin miners are fueled by sustainable energy, they may start accepting bitcoin again. These bitcoin miners are essential to proceed with transactions, they are the equivalent of modern-day banking services.

Microsoft, the multinational technology company, accepts bitcoin since 2014 [57] and started accepting bitcoin for their Microsoft Store since January 2021 [10]. AMC Theaters announced that they will accept bitcoin by the end of 2021.

On 8 June 2021 the country El Salvador passed a bitcoin law that indicates that bitcoin will be accepted as a currency in the country. The law has been put into action in September 2021[1].

1.1.3 Cryptocurrencies and Stocks

[22] describes that the market capitalization of all cryptocurrencies follows Zipf's law. The market capitalization of a cryptocurrency is calculated by multiplying the total number of coins by the current price of one unit of that cryptocurrency. For the market capitalization to follow Zipf's law, the cryptocurrency with the highest market capitalization must be double in value of the cryptocurrency with the second highest market capitalization, three times the third highest one, and so on. According to [33], the distribution of the share price of stocks also follows Zipf's law. Even if the stock market and cryptocurrencies have similarities, several dissimilarities are worth mentioning. Cryptocurrencies are decentralized and thus are not controlled by a central bank or other authorities. Stocks, on the other hand, are controlled by a centralized unit such as a central bank [13].

1.1.4 Relationship between Bitcoin and Internet Traffic

The results in [66] suggest that there is a relationship between internet attention and bitcoin's price. Moreover, there appears to be a strong Granger causal relationship between bitcoin's trading volume and what they call "internet attention". Internet attention is defined by the Google search volume index. This search volume index is calculated by normalizing the number of searches for a term with the total number of searches of other terms. If there exists a Granger causal relationship between two different time series, one time series is useful to predict the other.

Impact of Single Tweets on Cryptocurrencies

Elon Musk is known to impact the cryptocurrency market by his tweets [17]. By changing his Twitter bio by adding "#bitcoin" to it and referring to it with the tweet: "In retrospect, it was inevitable", bitcoin's price rose from $32,000 to over $38,000. Similarly, Elon Musk brought significant attention to the cryptocurrency Dogecoin. By tweeting: "I only sell Doge", the average trading volume per minute for Dogecoin went temporarily up from $1,942 per minute to $299,330.

Volume of Tweets

[14] analyses the impact of three factors on the price of bitcoin. These three include Tweet volume, Tweet sentiment, and Google Trends. [14] observed that Google's search volume index is highly correlated with both rising and falling of the price of bitcoin. The sentiment of the tweets was not consistent with the price changes and was not analyzed further in [14].

The number of tweets of the previous days related to bitcoin is a significant diver when it comes to bitcoin's realized volatility (RV) and bitcoin's trading volume [55]. RV is defined in [61] as the following: "Realized volatility is the assessment of variation in returns for an investment product by analyzing its historical returns within a defined period." Trading volume is defined as the following in [11] "Trading volume is equal to the number of coins that have exchanged hands during a defined period".

Tweet Sentiment

The results in [44] indicate that Twitter happiness sentiment is a significant predictor for bitcoin's price and several other cryptocurrencies. This Twitter happiness sentiment is calculated by giving words a happiness score. Every word is assigned a score between 1 and 9 based on their happiness. Words like love and laughter, for example, are ranked more positive than words like death and murder [8].

[34] confirmed that Tweet sentiment and tweet volume have the ability to predict bitcoin's price. Looking further into other cryptocurrencies, they did not all react to tweets in a similar fashion as bitcoin. For example, popular cryptocurrencies like Ethereum and ripple did not show predictive power with Twitter sentiment.

Sentiment analysis can be done with a model named VADER [30]. VADER is a rule-based model for sentiment analysis. The model calculates a sentiment score given a tweet. The output consists of a neutral, a negative, and a positive score. They claim that their VADER model outperforms individual human raters.

[42] compares three different features' ability to predict bitcoin's price: bitcoin's historical price, tweet volume, and tweet sentiment. They claim that the tweet sentiment is the worst among them. This conclusion is achieved by applying linear regression to both tweet sentiment and bitcoin's price and afterward analyzing correlation factors. It is important to note that these three features were implemented in separate models thus a combination of any of these features may or may not produce better results.

The impact of influential accounts compared to all users has been analyzed in [48]. 21 million tweets were collected between 1 June 2017 and 25 June 2018. The number of tweets for the 50 most influential accounts was a little over 17,000. By using a logistic model, the impact of several factors has been analyzed. They claim that the number of positive tweets of the previous day has a significant positive effect on the increase of bitcoin's price. The same is said about the number of negative tweets of the previous day. Interestingly, the number of neutral tweets did not have a significant effect on bitcoin's price. Looking into the number of positive and negative tweets of the most influential accounts, a similar effect is visible. This suggests that looking into the most influential accounts could be enough to see a correlation in the returns which is a lot fewer data compared to all tweets related to bitcoin.

Gold and Bitcoin

The ability of gold's price to predict bitcoin's price is evaluated by deep learning methods in [15]. Their motivation is that gold is one of the most important financial aspects and could potentially have an impact on the price of bitcoin. By using three different deep learning models, a convolutional neural network (CNN), an LSTM, and a gated recurrent unit (GRU), the impact of gold's price on bitcoin's price has been evaluated. Gold's price does not seem to be correlated with bitcoin's price. Twitter sentiment, however, does show a positive correlation with the LSTM model. They claim that when a positive tweet is posted regarding bitcoin, the price is expected to go up.

1.2 Related Work

1.2.1 Bitcoin prediction models

Decision Trees

KyrptoOracle [43] uses XGBoost to predict the price of bitcoin in the next minute by using VADER sentiment analysis of tweets collected. XGBoost is a decision tree-based ensemble. KyrptoOracle uses a weighted VADER score: the sentiment score of a tweet is multiplied by three different factors: the number of followers the user that posted the tweet has, the number of likes the tweet has, and the number of times the tweet has been retweeted by other users. Weighing in these factors distinguishes an unknown person with a couple of followers from a celebrity with hundreds and thousands of followers. As discussed before, Elon Musk's tweets have much more impact on the market than the average Twitter user with under a hundred followers. KyrptoOracle uses the moving average of bitcoin's price, the sentiment scores, and the previous sentiment score as input. The moving average of bitcoin's price is the average price of bitcoin for the last 100 values. KyrptoOracle obtained a Root Mean Square Error (RMSE) equal to 10 for their prediction. Notice that the number of likes and retweets may vary depending on the date. We assume that a tweet that has been posted a day ago probably has more likes and retweets than one that is posted a minute ago since it has had more time to get liked and be retweeted. It is important to take into account when the data has been extracted, or in other words, the time that passed since the tweet has been posted.

Recurrent Neural Networks and Long Short-Term Memory

The recurrent neural network (RNN) model in [50] has an overall price prediction accuracy of 77.62%. Their model implements its own sentiment analyzer and does not use the VADER model. Instead of using all tweets related to bitcoin, they choose to select tweets from specific Twitter accounts. These accounts are known for tweeting information related to bitcoin such as bitcoin-related news. This will result in fewer tweets in the data when compared to using all tweets from all users containing bitcoin-related terms. They claim an overall accuracy of 77.62%. In the graph in [50] where the predicted price and actual price are plotted over time, the predicted price seems to be lagging. It looks like the model learns to predict the next price by simply using the current value. According to [25], accuracy metrics can be misleading when these types of things occur in the graph. They suggest predicting the differences in value rather than the actual value is a better way to test if the model can forecast the future correctly.

Another way to capture social media trends is to use the relative change in followers on specific Twitter accounts or subscribers to bitcoin on Reddit [36].

Convolutional Neural Networks

The convolutional neural network (CNN) based model proposed in [21] achieved an accuracy of 75.2% on the test set it was evaluated on. The model was compared to several models such as LSTM models, random forests, XGBoost, logistic regression. The second-best model which is the LSTM has an accuracy of 67.2%. They used tweet sentiment, bitcoin price, and bitcoin blockchain features as their model's input.

Ensembles and Hybrid Models

In [41], attribute selection methods were used to select meaningful attributes and to deal with dimensionality issues. Here, a 30-day weighted moving average is used instead of an unweighted one for several features. Also, lagged features were used. Lagged features are features of the previous time intervals. For example, for a lag period of seven, the previous seven entries are used. They evaluated three ensembles, an artificial neural network (ANN), and a support vector machine (SVM). The results depend on the period they are evaluated. For the first period, one ensemble, which consists of an RNN and a Tree classifier, has yielded the best result: 62.91% accuracy. For the second period, the SVM model has yielded the best result: 59.45% accuracy. The first ensemble had an accuracy of only 49%. The second ensemble that has K-means clustering followed by an ANN or CNN that is followed by a Tree classifier did not perform well. Neither did the third, which is a mixture of ANN's, CNN's, and SVM's added together.

Using a CNN before an LSTM can efficiently improve the accuracy of the model's prediction according to [37]. The CNN model is responsible for extracting features and speeding up training since the convolution operation decreases the number of parameters in the network compared to a fully connected neural network. This means that fewer weights need to be trained. Their model tries to predict both the price of bitcoin and the direction of the price of bitcoin. Here, social media factors are not taken into account, however, different macroeconomic variables (gold's price, exchange rates, ...) and the Baidu index are considered. Baidu is a Chinese search platform and the Baidu index is obtained by calculating the number of searches for a term [62], similarly to the Google search index discussed in Section 1.1.1. Transaction information and technical indicators of bitcoin are also used. The overall results conclude that using the CNN-LSTM hybrid model yields equal or better results than a back-propagation (BP) neural network, a CNN, and an LSTM for various performance metrics. Only the precision metric for the CNN model is equal to the CNN-LSTM hybrid.

Further Details

The prediction interval is different for the different models that were discussed previously. Some models like KyrptoOracle [43] try to predict bitcoin's price for the next minute. The model in [58] tests different prediction intervals. These range from one hour up to one week. The results suggest that for their LSTM model, the prediction interval for a week yielded the best accuracy for both Ethereum's price and bitcoin's price. This model uses Telegram message volume and Google trend information. [65] compares a ten-second interval and a ten-minute interval with their model that predicts the sign of the future price change. Their results show that the ten-minute interval gives better results compared to the ten-second interval. They attribute this to the small rapid changes in bitcoin's price.

[32] showed that there is a weekly pattern in the number of daily transactions of bitcoin. This number is higher during the weekday when compared to the weekend. Looking further into the day of the week effect of bitcoin, [39] detected a positive Monday effect in the daily returns of bitcoin's price.

1.2.2 The BERT & BERTTweet models

The language representation model BERT introduced in [24] can also be used as a way to describe tweets. BERT stands for Bidirectional Encoder Representations from Transformers. BERT is designed to take context into account when representing language. The BERT model performs better than other language models. [64] claims that the BERT model is the best language model for Natural Language Processing (NLP).

[23] uses BERT to classify tweet sentiment instead of VADER that was used in some of the previously discussed models. They found that using BERT results in poor performance when it comes to calculating sentiment for tweets. The reason presented in the paper is that the BERT model is not trained for tweets, it is trained with Wikipedia articles. Since the language used in tweets is different, it may result in poorer performance. [46] introduces BERTTweet, a pre-trained language model for English tweets. According to their results, BERTTweet performs better at downstream NLP tasks such as sentiment analysis on tweets.

1.3 Research Questions

The following is an enumeration of the research questions that we want to answer and reflect on later.

RQ1. Is there a difference between using the VADER sentiment compound score and using the three individual scores (the negative, neutral, and positive sentiment scores) as input features for the prediction model?

[48] claimed that the impact of the number of neutral tweets is nonexistent compared to the number of positive and negative tweets. Notice that these are volume metrics and not sentiment values. [34] and [44] claim that sentiment can aid in predicting bitcoin's price, [42], however, claims that it does not perform well.

Hypothesis: Since three different numbers are represented by a single one, the compound score, we believe that information is lost that can be valuable. We believe that there could be a difference in the model's performance on the test set but this will remain relatively small: 1%-2% reduction in $RMSE$ when using the individual sentiment scores.

RQ2. Is there a difference in using the volume of all tweets that include the term "bitcoin" and using the volume of the filtered set? The filtered set does not contain tweets posted by bots, non-English tweets, and tweets that are removed within three years. What is the impact of using the number of positive, negative, and neutral tweets on a model's ability to predict bitcoin's price compared to using the filtered volume as input?

Looking at the individual sentiment volumes, [48] claims that the impact of the number of neutral tweets is non-existent compared to the number of positive and negative tweets. [55] illustrates that the number of tweets related to bitcoin of the previous days is a "significant diver" when it comes to bitcoin's realized volatility (RV).

Hypothesis: We expect that using the number of tweets of the filtered set will result in a decrease in $RMSE$ by 5% compared to the prediction model that takes as input the volume of all tweets. We believe that the positive, negative, and neutral volumes will improve the results by 5% compared to using the filtered volume.

RQ3. Will using a more sophisticated language representation model that is trained specifically for tweets (BERTTweet) increase the performance when compared to the baseline models and sentiment models?

As discussed in Section 1.2.2, the standard BERT model is trained with Wikipedia articles and did not perform well in the bitcoin prediction model [23]. The BERTTweet model introduced in [46], is trained on tweets instead of Wikipedia articles and does produce better results on tweet NLP tasks.

Hypothesis: We believe that the vector produced by the BERTTweet model will contain useful information about tweets. Using this vector as input for a model will result in a reduction in the $RMSE$ by 5-10% compared to the baseline models and sentiment models.

1.4 Contribution

With this work, we opt to bring clarity into the field of bitcoin prediction. By testing every feature independently on its ability to predict bitcoin's price, we can isolate its effect on the predictions. This allows us to compare these features with each other and give an in-depth analysis of the effect these features have on the results of our prediction models. We build further on the findings in [23]. They claim that the ill performance of BERT is due to the model being unable to understand the language used in tweets since it is pre-trained on Wikipedia articles. In our approach, we will introduce the BERT vector obtained from BERTTweet which is trained on tweet data. We review [23]'s claim by using the feature vectors obtained by the BERTTweet model. Furthermore, there is a lot of discussion whether tweet sentiment can aid predicting bitcoin's price in the related work. We want to take a deep look into the effect these sentiment features have on the model's ability to predict bitcoin's price.

Chapter 2

Foundations

In this chapter background, information about existing models and methods are discussed. These existing models and methods are the foundations of our models.

2.1 The LSTM Network

An LSTM cell consists of forget and control gates and a cell state [47]. Let x_t be the input vector at time t, h_t be the hidden state vector of the LSTM cell at time t and C_t be the cell state of the LSTM cell at time t. Every LSTM cell has a cell state C_t at time t. In general, the LSTM cell takes as input the previous cell state C_{t-1}, the previous hidden state h_{t-1} and a new input vector x_t [28]. The output of the LSTM is a new cell state C_t, a new hidden state h_t and a output vector o_t. This hidden state and cell state can then be passed to another LSTM cell or to the next cell at the next time step. The first part of the LSTM is the forget layer. Here, which information will be kept from the previous cell state is decided. The forget unit f_t is as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

with W_f the forget layer's weight matrix, b_f the bias and σ a sigmoid function that maps values between 0 and 1 [47]. Here, the cell decides what information is kept and what the cell will "forget". f_t consists out of values between 0 and 1 and is multiplied with the old cell state C_{t-1}. Next up, the input gate layer i_t and the new candidate values $\tilde{C}_t$ are multiplied to represent the new information. i_t and $\tilde{C}_t$ are calculated as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

with W_i and b_i the input layer's weight matrix and bias respectively, W_C and b_C the weight matrix and bias for the candidate values respectively [47]. *tanh* is a hyperbolic tangent function that maps values between -1 and 1. Now, the new cell state C_t, output vector o_t, and the new hidden state h_t can be defined as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t \cdot tanh(C_t)$$

with W_o and b_o the output layer's weight matrix and bias respectively. Figure 2.1 gives a visual summary of the LSTM cell's architecture. Figure 2.2 illustrates how the hidden state and the cell state are passed from an LSTM to the same LSTM at the next time state. Multiple LSTM cells can be concatenated to each other by using the output vector of an LSTM cell as input of another LSTM cell.

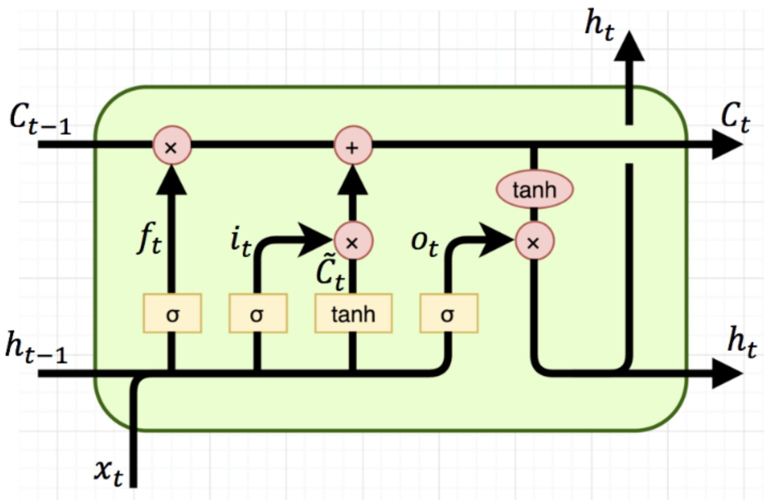


Figure 2.1: Single LSTM cell's architecture, source: [53].

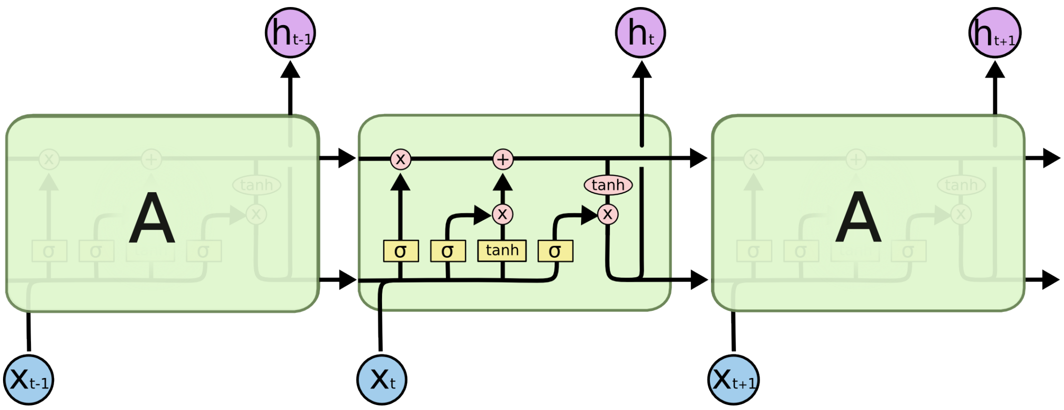


Figure 2.2: Illustration of how the hidden state and cell state are passed from an LSTM at a certain time state to the same LSTM at the next time state, source: [47].

2.2 The BERT Language Model

BERT is a language model that stands for Bidirectional Encoder Representations from Transformers [29]. The model takes both left and right context into account. This is done with a transformer. Transformers consist of encoders and decoders. Shortly, the encoders take the sentence as input and outputs a numerical representation for each word, like a vector. The decoders try to decode the output of the representation that the encoder generated. The BERT model is a multi-layer bidirectional transformer encoder. Compared to other language models, BERT applies bidirectional training of transformers. Moreover, BERT is designed to read both directions at once. Other models look at a sequence of text from left to right or from right to left. Some models combine both during training [29]. There is a base version of the model and a large version of the model. The base model has about a third fewer parameters compared to the large model.

The input of the BERT model exists of the token, segment, and positional embeddings. The token embeddings are word embeddings, the segment embeddings indicate to which sentence, A or B, the word belongs, and the positional embedding denotes the index of the token in the sequence. Figure 2.3 gives an example of these embeddings for a sentence pair. BERT is trained for two different tasks. The first

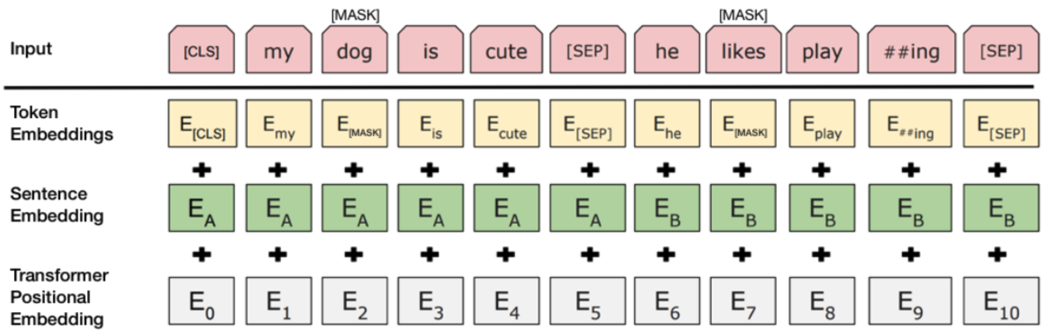


Figure 2.3: BERT input representation [24]. [CLS] indicates the start of the input sequence and [SEP] indicates the separation of sentences.

one is a prediction of a masked word. Here, approximately 15% of the words in a sentence are masked with a [MASK] token [24]. The model is trained to predict the masked word. The other task is the next sentence prediction. The model receives pairs of sentences as input [29] and must classify if the second pair follows the first. BERT's objective is to minimize the errors of both tasks during training. The idea is to make BERT get an understanding of language. These two tasks are part of the pre-training part. The other part of BERT is the fine-tuning part. Here, most of the parameters remain the same. There are a variety of different fine-tuning tasks that the model can be trained on. For example, a classification layer task can be added on top of BERT, or a named entity recognition [27]. BERT can also be used for feature extraction. The output of the first part of BERT is equal to the values of the hidden states of the last encoder layer. The idea is that these values can be seen as a vector that contextualizes the input sequence [35].

2.2.1 BERTTweet

The original BERT model introduced in [24] is pre-trained using Wikipedia articles. Since our interest lays in analyzing tweets, we will use BERTweet [46]. BERTweet is pre-trained on English tweets and it has a similar architecture as the BERT base model. BERTTweet produces better performance compared to other BERT models on tweet-related NLP tasks.

2.3 The VADER Sentiment Model

VADER (Valence Aware Dictionary for sEntiment Reasoning) is a rule-based sentiment model [30]. VADER assigns a sentiment score to a sentence by combining a sentiment lexicon approach with grammatical rules [40]. The sentiment lexicon approach assigns scores to each word in the sentence according to a predefined list. This list contains a sentiment score between -4 and 4 for each term. The higher the score, the more positive the sentiment is, the lower the score the more negative the sentiment is. Neutral words have a score equal to 0. Even emojis and acronyms are listed since they impact the sentiment of a sentence. Words like “good” and “great” have a positive score, while words like “horrible” and “bad” a more negative one. The sentiment of the complete sentence is calculated by taking the sum of each word’s sentiment and normalizing it afterward. For a sentence $W = (w_1,w_2, ..., w_n)$ of length n, with w_i the term at index i and s_i the sentiment score for w_i, the sum of the sentiment scores x of W is:

$$x = \sum_{i=1}^{n} s_i$$

The sentiment score $sent_W$ for W is as follows:

$$sent_W = \frac{x}{\sqrt{x^2 + \alpha}}$$

with α the normalization factor set to 15 [20]. It is important to include grammatical rules since the context can impact the sentiment score. Words like “not” and “but” can change the score of the entire sentence. Punctuation marks do amplify the score for a sentence, for example the sentence “I like it!!!” will have a higher VADER sentiment score than “I like it.” since the exclamation mark makes the emotion more intense [20]. VADER takes more than these two grammatical rules into account when calculating the final sentiment score of the sentence. VADER outperforms human raters in the task of classifying tweets as positive, neutral, or negative [30].

2.4 Cross-Validation

Cross-validation in machine learning is used to evaluate the model’s performance in a more robust way [56]. The idea is to split the data into multiple training and test sets, train the model on the training set, and evaluate it on the test set. This can be demonstrated with an example. Consider a dataset that is split into three parts of equal size: A, B, and C. This is called K-fold cross-validation with K the number of parts that the data is split into. First, the model is trained using parts A and B and evaluated on part C. Evaluation metrics such as error and accuracy are obtained and are saved for later. Next, the model is trained using parts B and C and evaluated on part A. At last, the model is trained using parts C and A and evaluated on part B. Table 2.1 summarizes the three different sets. In this case, K

Fold NR.	Training Set	Test Set
1	[A, B]	[C]
2	[B, C]	[A]
3	[C, A]	[B]

Table 2.1: K-fold cross-validation for a set with K=3.

is equal to three which is called the number of folds. There are different results for each training-test set combination. Analyzing the three different results for three folds is more reliable than only examining one of them.

Time Series

Using this kind of cross-validation with time series prediction models is not optimal [56]. Since the order of the elements in the data is important for time series prediction and the goal is to predict the future and not the past, a different approach is taken. It is important not to test the model on data that is older than the data in the training data, in other words: do not use the future to predict the past. Consider a dataset that is split into six parts of equal length: parts 1, 2, 3, 4, 5, and 6. Table 2.2 visualizes a three-fold cross-validation. Notice that for higher fold numbers, the

Fold NR.	Training Set	Test Set
1	[1, 2, 3]	[4]
2	[1, 2, 3, 4]	[5]
3	[1, 2, 3, 4, 5]	[6]

Table 2.2: K-fold cross-validation for a dataset consisting of time series data with K=3.

training set becomes larger.

2.5 Evaluation Metrics

In this section, we give a summary of different evaluation metrics that are used in our work. The different evaluation metrics are:

- Mean absolute error, MAE in short, is simply the mean of all differences between the prediction and the actual ground truth [52]. MAE is defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)$$

with N the number of examples, y_i the actual ground truth, and $\hat{y}_i$ the prediction for that corresponding ground truth.

- Mean square error, MSE in short, is the average of the squared differences between the ground truth values and the corresponding predicted values [52]. MSE is defined as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

with N the number of examples, y_i the actual ground truth, and $\hat{y}_i$ the prediction for that corresponding ground truth.

- Root mean square error, RMSE in short, is the square root of MSE and it is defined as follows:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

with N the number of examples, y_i the actual ground truth, and $\hat{y}_i$ the prediction for that corresponding ground truth.

- R^2 score is a goodness-of-fit measure and gives a number between 0 and 1 [26] that indicates how well a model fits the data (the higher this value the better the fit). It measures the amount of variance in the predictions of the model explained by the dataset, or in other words: it is the difference between the samples in the dataset and the predictions made by the model. R^2 score is defined as follows:

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \overline{y})^2}$$

with N the number of examples, $\overline{y}$ the mean of the ground truths, y_i the actual ground truth, and $\hat{y}_i$ the prediction for that corresponding ground truth.

- Pearson correlation coefficient, r, is an indication of both the strength, expressed in the value, and the direction, expressed in the sign, of a linear relationship [9]. r between two sets is defined as follows:

$$r = \frac{\sum_{i=1}^{N}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \overline{x})^2 \sum_{i=1}^{N}(y_i - \overline{y})^2}}$$

with N the number of examples, x_i and y_i values of each set and, $\overline{x}$ and $\overline{y}$ the mean of the values for each set.

MSE is highly biased on large values compared to RMSE [52]. In the experiments, we tried to use both error metrics as loss functions buts we found no difference in the predictions. We denote R^2 also as $R2$ and as $r2$ throughout the text.

Chapter 3

Methods

This chapter contains a description of the data collection methods, features extraction techniques, the implemented methods, and the architecture of the prediction models.

3.1 Data

In this section, we discuss how data used in our experiments is collected, how the collected data looks like, how the data is processed, and how different features are generated.

3.1.1 Collecting the Data

Tweets

The tweet dataset is retrieved from [18]. This dataset contains tweets that include the term “bitcoin”. Looking into this dataset, we noticed that some gaps lack data making the dataset unusable. The publisher of the dataset posted information on how these tweets were initially extracted. With this information, the gaps are filled using the Twitter API [12] forming a dataset that consists of tweets starting from 8 March 2018 until 3 November 2018 without any gaps. This dataset will be referred to as the original dataset $Dataset_{org}$. Table 3.1 gives a sample of tweets contained in $Dataset_{org}$. Furthermore, the tweets lack additional information. As suggested in [43] and [48], not all Twitter accounts are equally important, and thus weighing the tweets is important. Although the number of favorites, retweets, and mentions of a tweet is included in $Dataset_{org}$, they are not favorable for weighing the tweets. As discussed in Section 1.2.1, we assume that there could be some inconsistency issues with these metrics. Favorites, retweets, and mentions of a tweet are dependent on the time that has passed since they were posted. This means that the time the tweet was retrieved influences these metrics making them inconsistent. In $Dataset_{org}$, the id of each tweet is given. Using the Twitter API, information for each tweet is extracted by looking up the tweet using its id. We are particularly interested in the number of followers the user that posted the tweet has. This forms a new dataset: the updated dataset $Dataset_{up}$. The original dataset $Dataset_{org}$ contains all bitcoin

Time	Username	Tweet text
23:59:53	**r__avalos**	***"What can you do with an e-wallet? Do you know what is an online wallet? What about Bitcoin? In Asia,..."***
23:59:41	**MohammedAliHai2**	***"@APompliano Bitcoin is there to stay. Initial hiccups. when a system in equilibrium is subjected 2 a change..."***
23:59:29	**GideonOPowell**	***I know what your thinking.... Yes she#HODL 's and yes immediately after this video#coffee...***
23:59:27	Evanscash	*"#Invest trades in#bitcoin for you at a no loss! As well as crypto trading (automated), crypto casino,...*
23:59:21	BitcoinXio	*Japan's Largest Bank Experiments Using Own Crypto at Convenience Store https://news.bitcoin....*
23:59:18	theAJSingleton	*YOU ARE ALL A BUNCH OF GOD-DAMN IDIOTS!!!#BITCOIN pic.Twitter.com/...*
23:59:09	**WorxCoin**	***"Hello Worx Twitter Community, We have a mandatory wallet update to be done before block 172,801 which is in about 4 days. If ...***
23:59:08	Evanscash	*#Cryptoworld company. Return on #bitcoin #investment is under 2month starting from as low as $30 Guys what are you waiting for? https://bit.ly/...*
23:59:05	**Hydeez411**	***Was just today having a conversation with my 14 year old daughter about the importance of Bitcoin and true freedom. I'm waking her early.***
23:59:05	**WorldCoinIndex**	***Bitcoin price index https://www.worldcoinindex.com /coin/bitcoin#USD#EUR #CNY#GBP#RUB pic.twit...***

Table 3.1: Sample of ten tweets in the original dataset $Dataset_{org}$ posted on 2018-09-01, the username is the Twitter handle of the user that posted the tweet. Some tweets' text has been cut to fit the table. The tweets with bold usernames are tweets that remain in the updated dataset $Dataset_{up}$.

tweets from 8 March 2018 until 3 November 2018 and has $7,167,483$ tweets. Not all the tweets that are found in $Dataset_{org}$ can be retrieved via the Twitter API. Some tweets are deleted or the account that is responsible for posting the tweet has been removed or deleted. This means that $Dataset_{up}$ contains fewer tweets. $Dataset_{up}$ contains $5,158,502$ tweets which is 72.2% of the number of tweets in the original dataset $Dataset_{org}$. Figure 3.1 plots the volume of tweets for both $Dataset_{org}$ and $Dataset_{up}$ over time. The figure illustrates that the trend in the volume of tweets is preserved. Tweets in Table 3.1 written in bold are tweets that remain in $Dataset_{up}$, the others are removed. We have noticed that tweets that do not contain the term

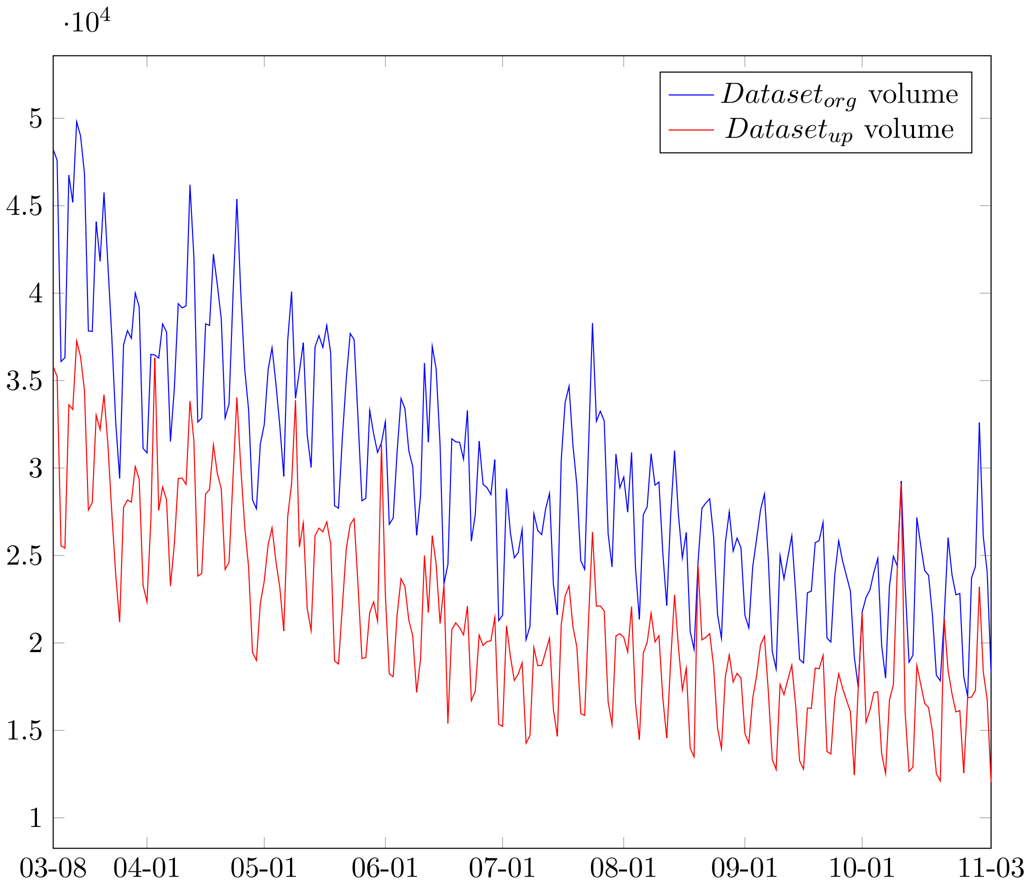


Figure 3.1: The volume of the original dataset $Dataset_{org}$ and the volume of the updated dataset $Dataset_{up}$ over time.

"bitcoin" are also found in the original dataset $Dataset_{org}$. These tweets, however, are related to cryptocurrencies in general.

Bitcoin

Bitcoin data is retrieved from [4]. This dataset contains the close price of bitcoin on a per-minute basis. Section 3.1.3 discusses these features further.

3.1.2 Data Processing

This section discusses how the updated dataset $Dataset_{up}$ is processed further before it is used in the prediction models. The resulting dataset after processing is called the filtered dataset $Dataset_{filt}$.

Sentiment Calculation

To calculate the sentiment for each tweet, there are several steps the text of a tweet needs to undergo. VADER performs sentiment analysis mainly for the English language [30], hence only English tweets are selected. When extracting additional information using the Twitter API, the language of the tweet is given which is classified by Twitter [12]. Roughly 99.7% of the tweets in $Dataset_{up}$ are classified as being written in English. Next up, the tweet's text is processed. We followed a similar approach to [43], [34], and [48]. [49] analyzed the effects of different preprocessing steps on VADER sentiment and the correlation it has with bitcoin's price. They described numerous ways to preprocess a tweet and in what order these preprocessing steps could be executed. They conclude that the best preprocessing process is different for the size of the dataset and the feature. Different effects have been encountered when the positive, neutral and, negative tweet sentiment was tested. The overall best strategy does include some processing of the text over using unprocessed text. After filtering non-English tweets, common delimiters in the tweet's text are removed. These include tabs, newlines ("\n") and returns ("\r"). Furthermore, if there are any contractions, they are transformed to their full form. For example "we've" becomes "we have". At the final step, the text is converted to lowercase and all the @usernames, URLs, white spaces, #'s, and stopwords are filtered out. These stopwords include words like "me", "then", and "over". Table 3.3 gives a sample of a tweet that undergoes these processing steps. It is important to

Processing step	Tweet text
Raw tweet	I'm having GOOD ♡\tfeelings about this bitcoin stuff \n \r
Pre-pocessing	I'm having GOOD ♡ feelings about this bitcoin stuff
Removing contractions	I am having GOOD ♡ feelings about this bitcoin stuff
Final step (removing stopwords, to lowercase,...)	good ♡ feelings bitcoin stuff

Table 3.2: Example of a tweet's text being processed for VADER sentiment.

note that emojis should not be removed since it affects VADER's sentiment scores. Afterward, for each tweet, the sentiment is calculated using the pre-trained VADER model [30].

Tweet text	Compound score
I bought bitcoin at the wrong damn time smh	-0.7964
And now it is a discussion on Bitcoin.	0.0
I hope this is true. I'll support people trying to make this true. But I don't want to get overconfident. If Bitcoin never gets shut down it will be because the community solved every problem thrown at it. This requires work, careful thought, and vigilance. Let's do it.	0.8885
Where did you get you Bitcoin is Better stamp? I want one!	0.4939
Who created bitcoin tho?	0.25
Bitcoin historic selloff below 600 or lower	-0.296

Table 3.3: Example of a tweet's text being processed for VADER sentiment.

Removing Bots

Tweets can be posted by bots and since we are interested in user sentiment, we assume that these bots are irrelevant and cause noise. A tweet is classified as a bot in a similar way that is used in [60] which is if it contains certain terms or a combination of terms. These terms can be found in Table 3.4. In the updated dataset $Dataset_{up}$ with only English words, $321,603$ tweets are classified as being posted by bot accounts. Table 3.5 gives a sample of tweets that have been tagged as being posted by bots. The resulting dataset, the filtered dataset $Dataset_{filt}$, does contain

Terms	mpgvip, freebitcoin, livescore, makeyourownlane, footballcoin, entertaining, subscribe
Bigrams	(free, bitcoin), (current, price), (bitcoin, price), (earn, bitcoin)
Trigrams	(start, trading, bitcoin)

Table 3.4: Terms and combination of terms to identify if a tweet is posted by a bot [60].

tweets that are not posted by bots and are tagged by Twitter as being written in English. $Dataset_{filt}$ is a subset of $Dataset_{up}$.

3.1.3 Features

In this section, we give a summary of all the features that are used as input in our prediction models. Each feature represents a value for a time period. This period is called the time interval i.

Bitcoin price index https://www.worldcoinindex.com/coin/bitcoin #USD #EUR #CNY #GBP #RUBpic.twitter.com/qPYQdRtd7g
Grab your F.ree bitcoins here :) make monet with bitcoin easy http://bit.ly/2maS1C3fcoin #bitcoin #free #litecoin #tradebitcoin #mineebitcoin #altcoins #blockchain **#freebitcoin** ...
Earn Bitcoin Without Investment - How To Earn Bitcoin Without Mining | Bitcoin Bot http://americanstocktrader.com/...
Start trading bitcoin and altcoin with the cheapest exchange in the space https://ift.tt/2EkDGdK
Butters demand that the statist legal system intervene to resolve their civil war over the hardforking of the glorious and infallible blockchain - http://bitcoin-wall.com - Generate Bitcoin. Take your **free Bitcoin**

Table 3.5: Example of a tweet's text that are tagged as being posted by a bot.

Bitcoin Features

Table 3.6 gives a sample of the two bitcoin features used in the experiments. The close price, c, is the price of bitcoin at the end of the interval. The difference in close price, d, is the difference between the close price at the current time step and the close price at the previous time step. All price features are expressed in United States Dollar (USD). Date and time are in Coordinated Universal Time (UTC) format.

Date & time	Close price c	Difference in close price d
2018-03-08 00:00	9904.52	0.00
2018-03-08 01:00	9939.32	34.80
2018-03-08 02:00	9870.05	-69.27
2018-03-08 03:00	9625.68	-244.37
2018-03-08 04:00	9546.51	-79.17

Table 3.6: Sample of bitcoin features per hour on 8 March 2018. The close price, c, is the closing price of bitcoin for that time interval i expressed in USD. The length of the time interval is 1 hour. The difference in close price, d, is the difference between the close price at the current time step and the close price at the previous time step expressed in USD. Date and time are expressed in UTC format.

Tweet Features

It is important to note that not every tweet is as important as another. [48] suggests that only looking into tweets of several influential accounts could be enough to predict bitcoin's price. Some models like KyrptoOracle [43] weigh the tweets by multiplying the sentiment with the number of followers, favorites, and retweets. Table 3.7 gives a sample of the Twitter features that are used in our experiments. These are the weighted average of the sentiment scores of all tweets in a time interval. The sentiment score for each tweet is calculated using the VADER model [30]. These scores are weighted by the number of followers. Let L be the set of all tweets of a time interval. Let N be the size of L. For each tweet $l \in L$, let x_l be the sentiment score of l and f_l be the number of followers of the user that posted tweet l. The weighted average of the sentiment score of L is defined as follows:

$$\frac{\sum_{l=1}^{N} x_l * f_l}{\sum_{l=1}^{N} f_l}$$

A BERT vector for a tweet is the BERT model's last layer hidden-state, more in

Time	*pos*	*neu*	*neg*	*comp*
2018-03-08 00:00	0.09714	0.79954	0.10330	-0.00716
2018-03-08 01:00	0.09529	0.72281	0.11204	-0.05392
2018-03-08 02:00	0.07579	0.78584	0.13778	-0.10633
2018-03-08 03:00	0.17965	0.78399	0.03588	0.18952
2018-03-08 04:00	0.09395	0.77213	0.13392	0.01725

Table 3.7: Sample of tweet sentiment features per hour on 8 March 2018. The positive sentiment score *pos* is the weighted average positive sentiment of all tweets during that time interval. Similarly, the neutral sentiment score *neu*, the negative sentiment score *neg*, and the compound sentiment score *comp* are defined.

Section 2.2. In the experiments, the pre-trained BERTTweet base model is used [46], the vector's length is equal to 768. Consider $\overline{BERT}$ to be the weighted average of all BERT vectors in a time interval. The weighted average is calculated similarly to the weighted average of the sentiment scores. Table 3.8 gives a sample of these vectors.

Date & Time	$\overline{BERT}$
2018-03-08 00:00	[0.27227, -0.15431, ... , -0.15249, -0.17832]
2018-03-08 01:00	[0.26831, -0.15458, ... , -0.15086, -0.17658]
2018-03-08 02:00	[0.26419, -0.15815, ... , -0.15539, -0.17841]
2018-03-08 03:00	[0.26212, -0.16070, ... , -0.15344, -0.18008]

Table 3.8: Sample of the weighted average BERT vector $\overline{BERT}$ of all tweets in a time interval. Here the time interval i is 1 hour.

Table 3.9 gives a sample of the different volume features. *volorg* is the number of tweets in the original dataset $Dataset_{org}$. $volfilt$ is the number of tweets in the filtered dataset $Dataset_{filt}$. *volpos*, *volneu*, and *volneg* are the number of positive, neutral and, negative tweets respectively in the filtered dataset $Dataset_{filt}$. A tweet is tagged positive when the compound score is greater or equal to 0.05. If the tweet's compound score is less or equal to -0.05 it is tagged as negative. Otherwise, it will be tagged as being a neutral tweet. These rules are taken from [6]. Note that the sum of *volpos*, *volneu*, and *volneg* is equal to $volfilt$.

Date	$volorg$	$volfilt$	$volpos$	$volneu$	$volneg$
2018-03-08	48214	29920	12204	12175	5541
2018-03-09	47604	30017	12443	11712	5862
2018-03-10	36091	21947	9285	8320	4342
2018-03-11	36309	21223	8871	8755	3597
2018-03-12	46743	28396	11023	11913	5460

Table 3.9: Sample of tweet volume features per day. $volorg$ is the number of tweets in the original dataset $Dataset_{org}$. $volfilt$ is the number of tweets in the filtered dataset $Dataset_{filt}$. $volpos$, $volneu$, and $volneg$ are the number of positive, neutral, and negative tweets respectively in the filtered dataset $Dataset_{filt}$.

Feature Vectors

Before defining the feature vectors, time lagging is discussed. Time lagged features are previous values of the feature. For example: for the close price feature, the previous N values can be considered as time lagged features. Table 3.10 gives the time lagged features for bitcoin's close price.

Date & Time	c_t	c_{t-1}	c_{t-2}	c_{t-3}
2018-03-08 00:00:00	9904.52	-	-	-
2018-03-08 01:00:00	9939.32	9904.52	-	-
2018-03-08 02:00:00	9870.05	9939.32	9904.52	-
2018-03-08 03:00:00	9625.68	9870.05	9939.32	9904.52
2018-03-08 04:00:00	9546.51	9625.68	9870.05	9939.32

Table 3.10: Sample of bitcoin's close price time lagged features. c_t is bitcoin's close price at time step t. The time interval i is 1 hour.

A feature vector consists of time lagged features. The window size w defines the length of the feature vector. The following feature vectors with window size w and time interval i are defined:

- The close price vector C:

$$C = \{c_i | (t - w) < i \leq t\}$$

  with c_t the close price of bitcoin at time t.

- The difference in close price vector D:

$$D = \{d_i | (t - w) < i \leq t\}$$

  with d_t the difference in close price of bitcoin at time t.

- The positive, neutral, and negative sentiment vectors POS, NEU, and NEG:

$$POS = \{pos_i | (t - w) < i \leq t\}$$

$$NEU = \{neu_i | (t-w) < i \leq t\}$$
$$NEG = \{neg_i | (t-w) < i \leq t\}$$

with pos_t, neu_t, and neg_t the weighted average of the positive, neutral, and negative sentiment score respectively at time t.

- The compound sentiment vector $COMP$:

$$COMP = \{comp_i | (t-w) < i < t\}$$

with $comp_t$ the weighted average of the compound sentiment score at time t.

- The original tweet volume vector VO:

$$VO = \{volorg_j | (t-w) < j \leq t\}$$

with $volorg_t$ the number of tweets in the original dataset $Dataset_{org}$ at time t.

- The filtered tweet volume vector VF:

$$VF = \{volfilt_j | (t-w) < j \leq t\}$$

with $volfilt_t$ the number of tweets in the filtered dataset $Dataset_{filt}$ at time t.

- The positive, neutral and negative tweet volume vectors $VPOS$, $VNEU$, and $VNEG$:

$$VPOS = \{volpos_j | (t-w) < j \leq t\}$$
$$VNEU = \{volneu_j | (t-w) < j \leq t\}$$
$$VNEG = \{volneg_j | (t-w) < j \leq t\}$$

with $volpos_t$, $volneu_t$, and $volneg_t$ the number of positive, neutral, and negative tweets in the filtered dataset $Dataset_{filt}$ at time t.

Note that for the first w entries in the dataset, the feature vectors do not exist since the previous values for these features are not present in the dataset.

3.2 Cross-Validation

Data used in the experiments is split similarly as we discussed in 2.4. The data starts on 8 March 2018 and ends on 3 November 2018. The details of the different splits can be found in Table 3.11. Notice that every fold has a different length of the training set. The size of the validation set is the last 15 day long period from the training set.

3.3 Architecture of the Models

In this section, the architecture of the different models is described in detail. First, the skeleton that is identical in all models is discussed, afterward, each model's unique details are addressed.

Fold NR.	Training Set	Test Set
1	[3, 4, 5, 6]	[7]
2	[3, 4, 5, 6, 7]	[8]
3	[3, 4, 5, 6, 7,8]	[9]
4	[3, 4, 5, 6, 7, 8, 9]	[10,11]

Table 3.11: K-fold cross-validation for the dataset used in the experiments. Numbers in the training and test set represent the number of the month that is included.

3.3.1 The Model's Skeleton

Each model takes different input features with different sizes. There is, however, a part of every model that remains unchanged. In this section, the parts of the models that remain the same are discussed, this is the model's skeleton. The skeleton consists of three unidirectional LSTM cells connected to each other followed by a fully connected dense layer. The output dimension of the dense layer is 1 which is the prediction $\hat{y}$. No activation function is used. The input dimension of the LSTM and the dense layer varies for the different models. Each LSTM cell has a dropout probability of 0.2. The objective for all the prediction models is to minimize the MSE loss $\mathcal{L}_{MSE}$ which is defined as follows:

$$\mathcal{L}_{MSE} = \frac{1}{B}\sum_{i=1}^{B}(y_i - \hat{y}_i)^2$$

with B the batch size, y_i the actual ground truth, and $\hat{y}_i$ the prediction for that corresponding ground truth. Every model is trained to either predict bitcoin's next close price c_{t+1} or bitcoin's next difference in close price d_{t+1}. In other words, the output of the model $\hat{y}$ is desired to be equal to c_{t+1} or d_{t+1}. The former models are the non-time differenced models and the latter are the time differenced models. During training, the model's parameters are optimized using the Adam algorithm with a weight decay of $1e-6$ and a learning rate of 0.001. The batch size is set 64. All input features are normalized using min-max normalization. For each element x in a set X, min-max normalization is defined as follows:

$$x_{scaled} = \frac{x - min(X)}{max(X) - min(X)}$$

with $min(X)$ and $max(X)$ being the minimum and maximum value of set X respectively. Min-max normalization is often referred to as min-max scaling. Figure 3.2 visualizes the skeleton of each model that remains unchanged. The skeleton model takes a sample feature vector $F = [\ f_t,\ f_{t-1}\ ,...,\ f_{t-w+1}\ ]$ as input with w the window size. By combining different feature vectors from Section 3.1.3, different models are defined. For each of these models, different window sizes w and different time intervals i can be selected. Note that the window size w and the number of different feature vectors determine the input dimension of the first LSTM layer.

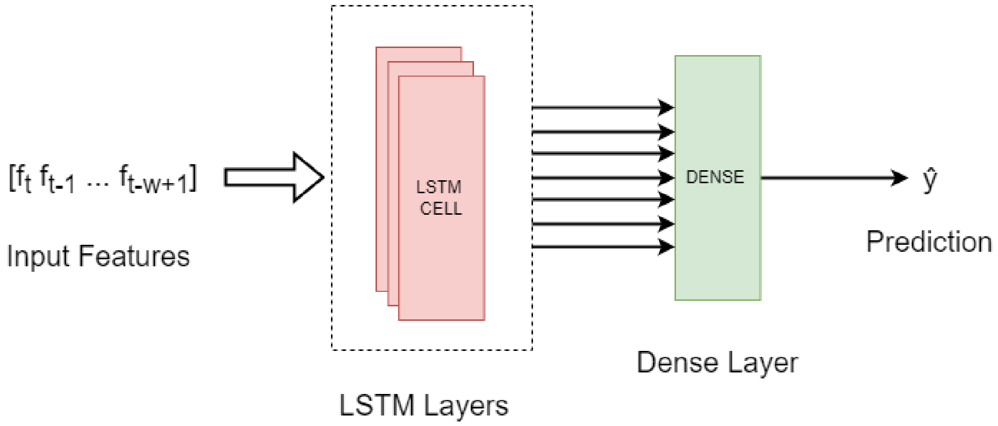


Figure 3.2: The architecture of the model's skeleton. f_i represents a value for a feature.

3.3.2 The Baseline Models

The first baseline model is the non-time differenced baseline model $Base_c$. Figure 3.3 illustrates the model's architecture. $Base_c$ takes as input the close price vector C and predicts the close price for the next time interval c_{t+1}. The second baseline model

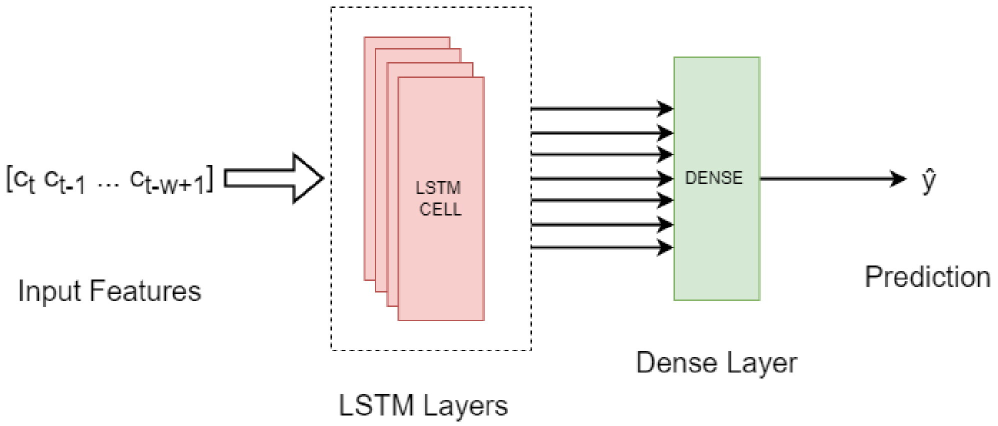


Figure 3.3: The architecture of the non-time differenced baseline model $Base_c$.

is the time differenced baseline model $Base_{td}$. Figure 3.4 illustrates the model's architecture. $Base_{td}$ takes as input the the difference in close price vector D and predicts the difference in close price for the next time interval d_{t+1}. For both baseline models, the first layer of the LSTM takes w features and has a hidden size of 128. The second and third cells take 128 features as input and have a hidden size of 128. Table 3.12 summarizes these details.

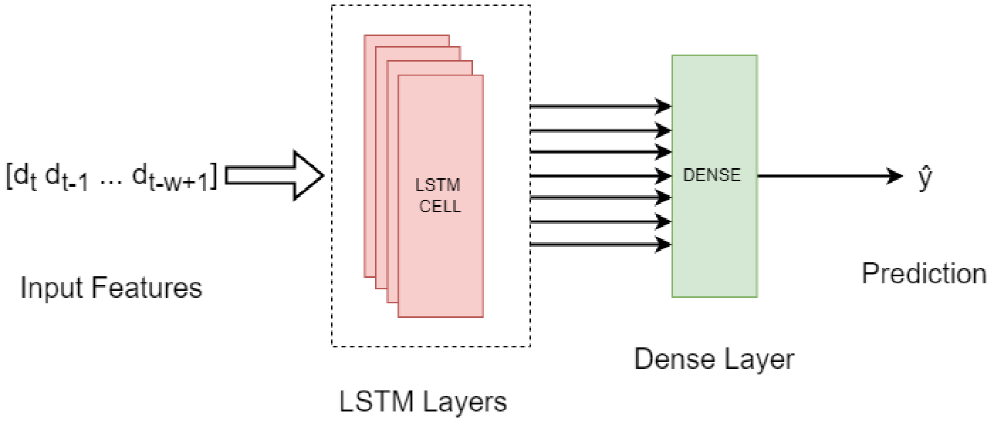


Figure 3.4: The architecture of the time differenced baseline model $Base_{td}$.

$Base_c$ & $Base_{td}$	hidden dimension LSTM Layers dropout probability activation layer	128 3 0.2 None
Optimizer	Algorithm weight decay learning rate	Adam 1e-6 0.001

Table 3.12: Summary of the parameters for the baseline models and the optimizer.

3.3.3 The Sentiment Models

In this section, the sentiment models are discussed in detail. There are two different ways that a model can use the Twitter sentiment features: by using the positive, neutral, and negative sentiment vectors POS, NEU, and NEG as input or by using the compound sentiment vector $COMP$ as input. The non-time differenced sentiment model $Sent_c$ takes as input the POS, NEU, NEG, and C vectors by concatenating them to each other and min-max normalizing them afterward. Figure 3.5 illustrates the architecture of $Sent_c$. $Sent_c$ predicts the value for c_{t+1} which is the close price at the next time interval starting. The features are min-max normalized before passing them to the models. The time differenced sentiment model $Sent_{td}$ is the time differenced version of $Sent_c$. Figure 3.6 illustrates the architecture of $Sent_{td}$. The configuration for the LSTM layers, the dense layer, and the optimizer for both $Sent_c$ and $Sent_{td}$ remain the same as the baseline models and can be found in Table 3.12. Next up, the non-time differenced compound sentiment model $Comp_c$ and the time differenced compound sentiment model $Comp_{td}$ are introduced. $Comp_c$ takes the concatenation of $COMP$ and C as input and $Comp_{td}$ takes the concatenation of $COMP$ and D as input. These features are min-max normalized. Figures 3.7 and 3.8 illustrate the architecture of $Comp_c$ and $Comp_{td}$ respectively.

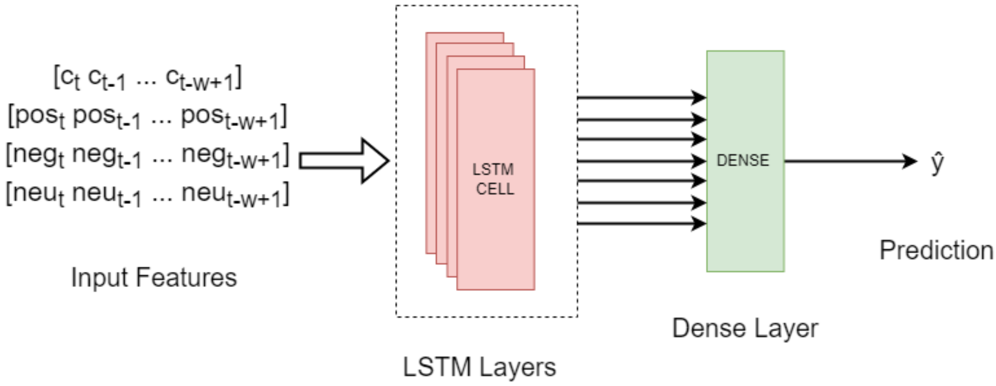


Figure 3.5: The architecture of the non-time differenced sentiment model $Sent_c$.

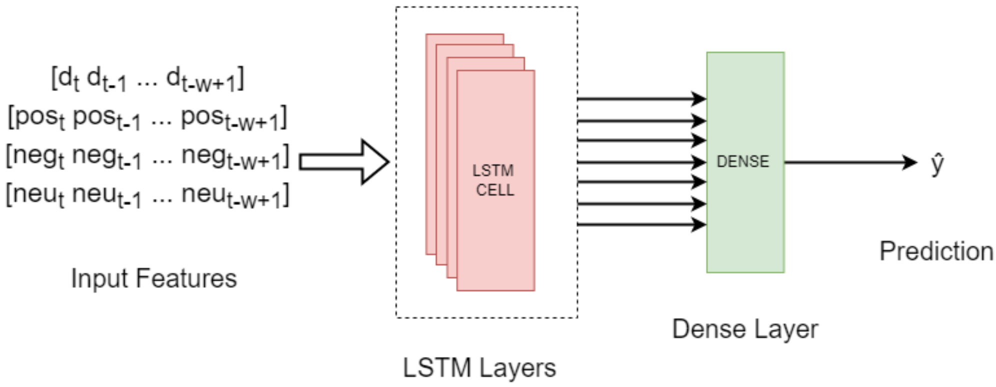


Figure 3.6: The architecture of the time differenced sentiment model $Sent_{td}$.

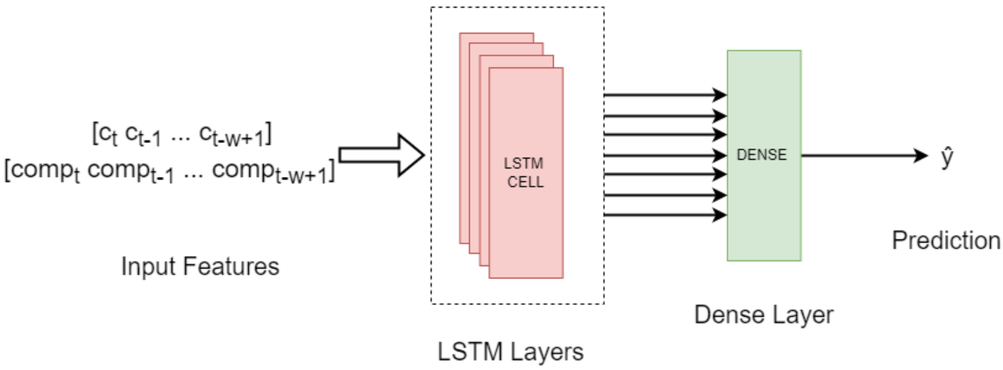


Figure 3.7: The architecture of the non-time differenced compound sentiment model $Comp_c$.

The configuration for the LSTM layers, the dense layer, and the optimizer remain

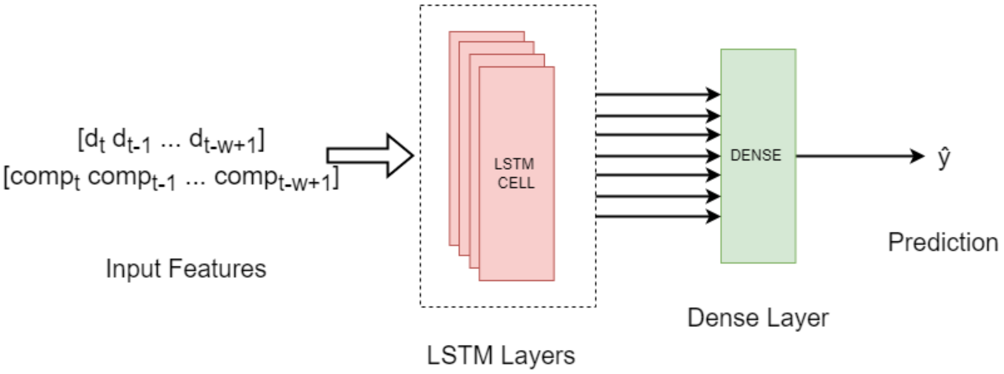


Figure 3.8: The architecture of the time differenced compound model $Comp_{td}$.

the same except for the hidden dimension which is equal to 64. Table 3.13 summarizes the details. The first

$Comp_c$ & $Comp_{td}$	hidden dimension	64
	LSTM Layers	3
	dropout probability	0.2
	activation layer	None
Optimizer	Algorithm	Adam
	weight decay	1e-6
	learning rate	0.001

Table 3.13: Summary of the parameters for $Comp_c$, $Comp_{td}$ and the optimizer.

3.3.4 The Tweet Volume-Based Models

In this section, the prediction models that use tweet volume as input are discussed in detail. First, the prediction models that take the original tweet volume vector VO as input are discussed. These are the non-time differenced original tweet volume model $VolumeOrg_c$ and time differenced original tweet volume model $VolumeOrg_{td}$. $VolumeOrg_c$ takes the concatenation of VO and C as input and $VolumeOrg_{td}$ takes the concatenation of VO and D as input. Figures 3.9 and 3.10 visualize the architecture of $VolumeOrg_c$ and $VolumeOrg_{td}$ respectively.

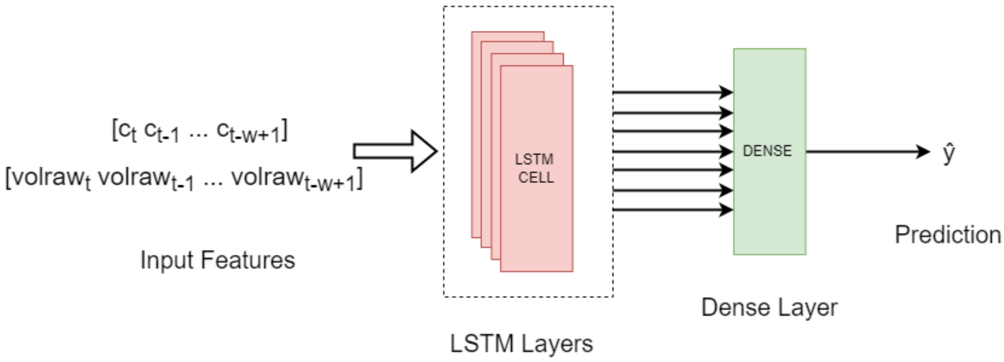


Figure 3.9: The architecture of the non-time differenced original tweet volume model $VolumeOrg_c$.

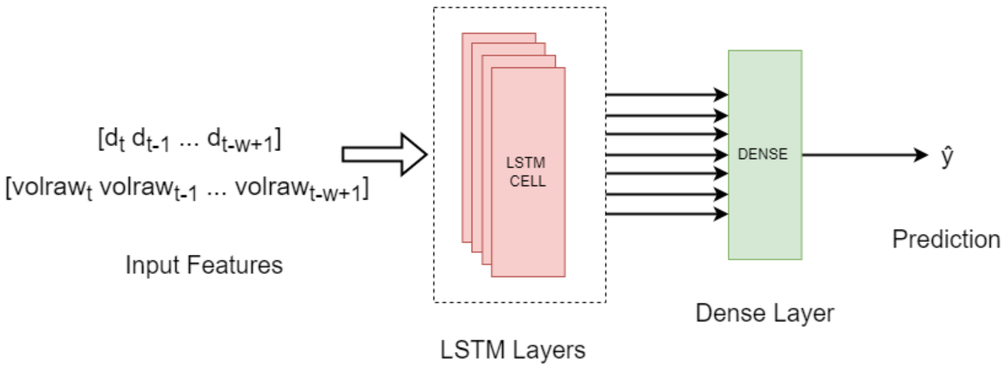


Figure 3.10: The architecture of the time differenced original tweet volume model $VolumeOrg_{td}$.

Tweet volume-based models	hidden dimension	64
	LSTM Layers	3
	dropout probability	0.2
	activation layer	None
Optimizer	Algorithm	Adam
	weight decay	1e-6
	learning rate	0.001

Table 3.14: Summary of the parameters for tweet volume-based models and the optimizer.

The configuration for the LSTM layers, the dense layer, and the optimizer remain unchanged and are summarized in Table 3.14. $VolumeOrg_{td}$ predicts c_{t+1}, $VolumeOrg_c$ predicts d_{t+1}.

Second, the non-time differenced filtered tweet volume model $VolumeFilt_c$ and the time differenced filtered tweet volume model $VolumeFilt_{td}$ are introduced. These take as input the filtered tweet volume vector VF alongside C (for $VolumeFilt_c$) or D (for $VolumeFilt_{td}$). Figures 3.11 and 3.12 visualize the architecture of $VolumeFilt_c$ and $VolumeFilt_{td}$ respectively. The configuration for the LSTM layers, the dense

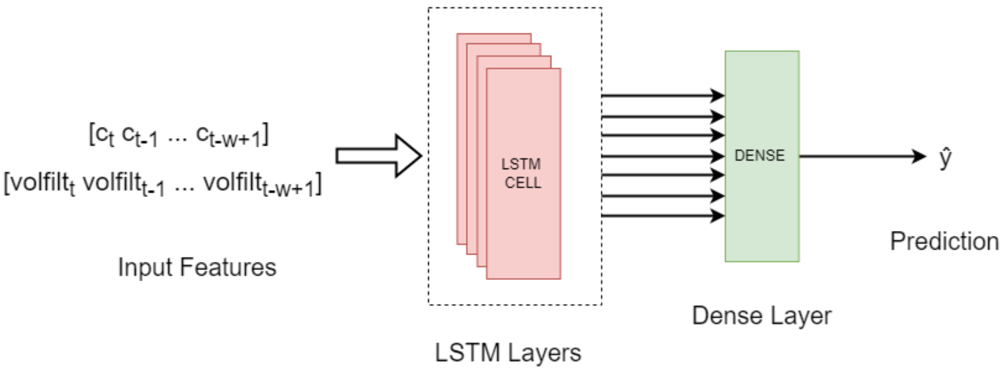


Figure 3.11: The architecture of the non-time differenced filtered tweet volume model $VolumeFilt_c$.

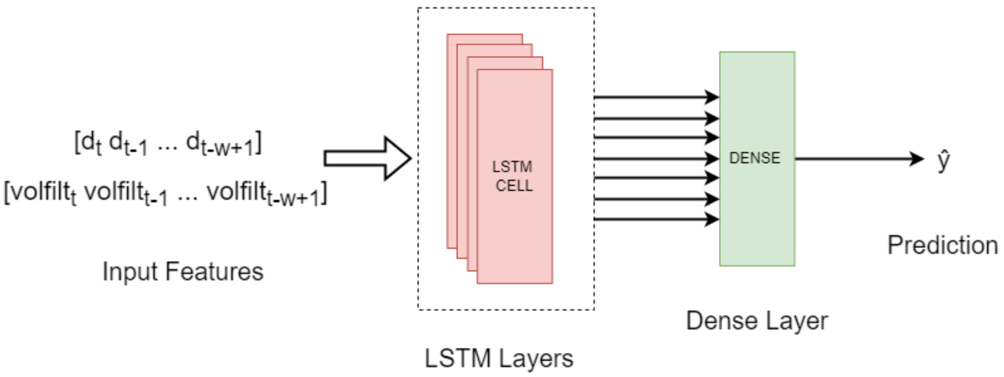


Figure 3.12: The architecture of the time differenced filtered tweet volume model $VolumeFilt_{td}$.

layer, and the optimizer are summarized in Table 3.14. Lastly, the positive, neutral, and negative tweet volume vectors $VPOS$, $VNEU$, and $VNEG$ are used in two models. Again, there is the non-time differenced version $VolumeSent_c$ and the time differenced version $VolumeSent_{td}$. $VolumeSent_c$ takes the concatenation of $VPOS$, $VNEU$, $VNEG$, and C vectors as input and predicts c_{t+1} while $VolumeSent_{td}$ takes the concatenation of $VPOS$, $VNEU$, $VNEG$, and D vectors as input and predicts d_{t+1}. Figures 3.13 and 3.14 visualize the architecture of $VolumeSent_c$ and $VolumeSent_{td}$ respectively. Table 3.14 summarizes the configuration details for both of these models.

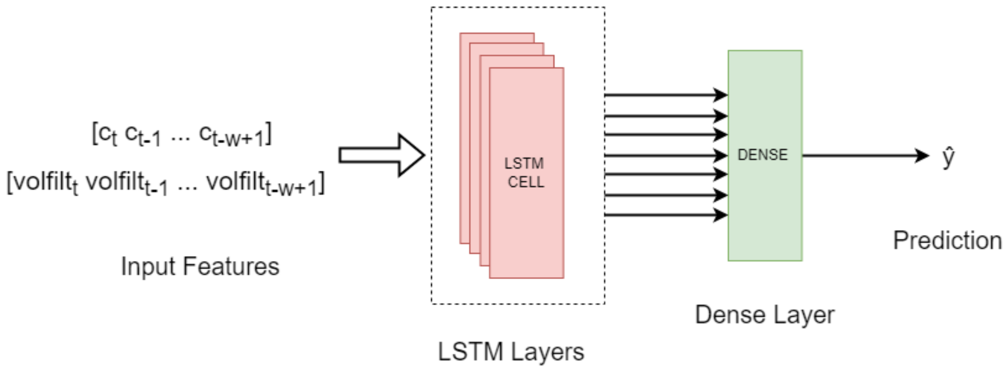


Figure 3.13: The architecture of the non-time differenced sentiment volume model $VolumeSent_c$.

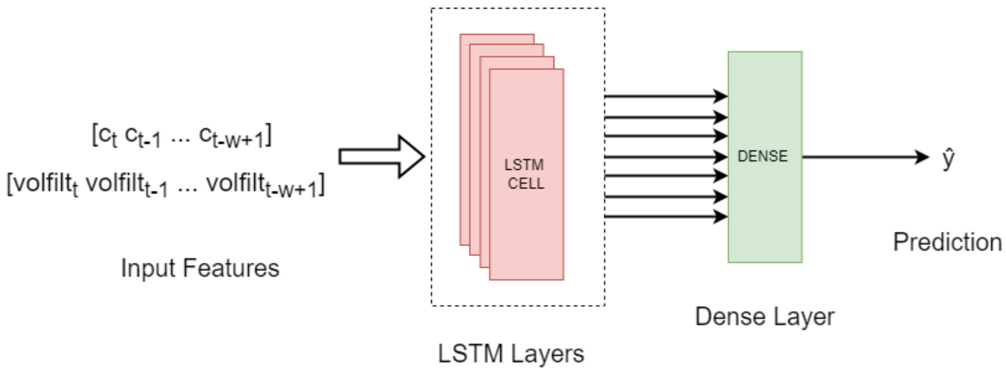


Figure 3.14: The architecture of the time differenced sentiment volume model $VolumeSent_{td}$.

3.3.5 The Dense-BERT Models

In this section, the prediction models use as input the BERT feature $\overline{BERT}$. This vector is the (weighted) average of all the BERT vectors for that time interval, more in Section 3.1.3. $\overline{BERT}$ is defined as follows:

$$\overline{BERT} = [b_1, b_2, ..., b_{768}]$$

with b_i the value at index i.

First, we introduce the non-time differenced Dense-BERT model $DenseBERT_c$. The architecture of this model is different from the previous ones, namely, it has two parts. The first part passes $\overline{BERT}$ through a dense layer. The result is a smaller intermediate vector $P = [p_1, p_2, ..., p_s]$ with length s. Ideally, s is selected to be smaller than $\overline{BERT}$'s size. The idea is that the model learns to compress $\overline{BERT}$ before using it in the LSTM layers. The second part of the model remains the same as the previous models: the intermediate vector P will be concatenated to the close price vector C and passed to the LSTM layers which are followed by another dense layer. $\overline{BERT}$ and C are min-max normalized. Figure 3.16 illustrates the architecture of $DenseBERT_c$ in detail. For the first part of the model, the input of the dense layer is equal to 768 which is the length of the $BERT$ vector. The size s of intermediate vector P is set to 10, thus the dense layer takes as input a vector of length 768 and has as output a vector of length s which is in this case equal to 10. The configuration for the second part of the model such as the LSTM layers, the dense layer, and the optimizer remain the same. Table 3.15 summarizes these details. Note that the first LSTM layer's input size is equal to the sum of s and w.

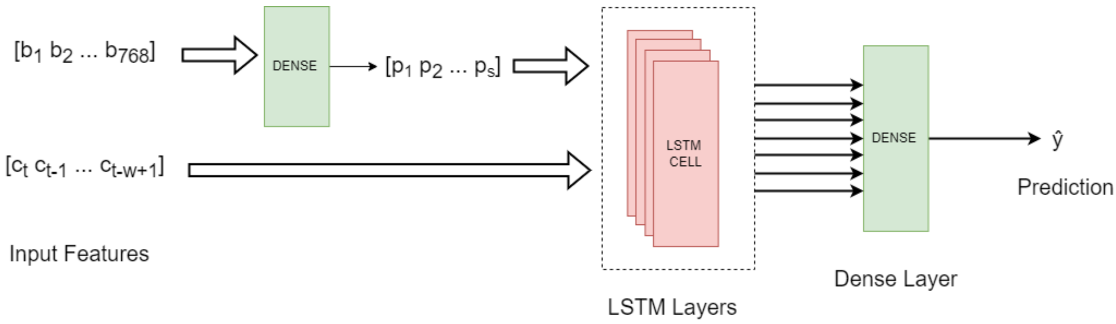


Figure 3.15: The architecture of the non-time differenced Dense-BERT model $DenseBERT_c$.

The Dense-BERT models	$BERT$ vector size	768
	dense out s	10
	hidden dimension	64
	LSTM Layers	3
	dropout probability	0.2
	activation layer	None
Optimizer	Algorithm	Adam
	weight decay	1e-6
	learning rate	0.001

Table 3.15: Summary of the parameters for the Dense-BERT models and the optimizer.

There is also a time differenced version of the Dense-BERT model: $DenseBERT_{td}$. This model is similar to $DenseBERT_c$, but instead of using the close price vector C, it uses the difference in close price vector D. Both $\overline{BERT}$ and D are min-max normalized. Figure 3.16 illustrates the architecture of $DenseBERT_{td}$. The configuration for the model remains the same and can be found in Table 3.15.

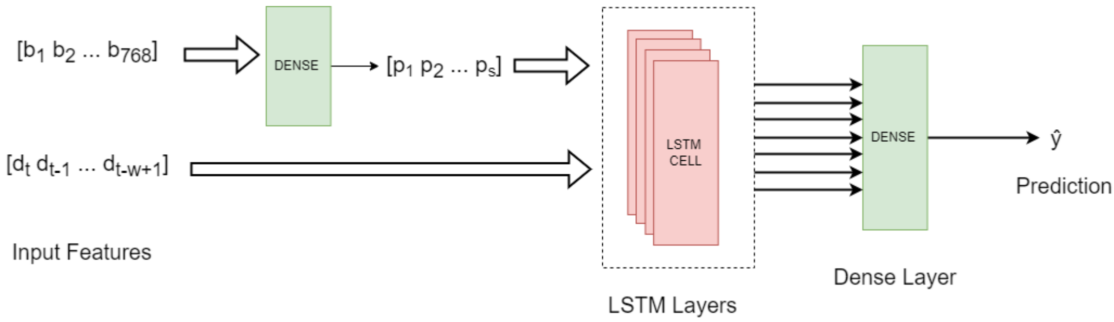


Figure 3.16: The architecture of the time differenced Dense-BERT model $DenseBERT_{td}$

3.4 Implementation Details and Used Libraries

The prediction models used in the experiments are written using the python library PyTorch [51]. The VADER model implementation is taken from the natural language toolkit (nltk) [38]. The BERT vectors are calculated using the $BERTTweet_{Base}$ model [5]. Both the prediction models and the BERTWeet models are run on the GPU "Nvidia GeForce GTX 850M". The stopwords are taken from the nltk corpus [38]. Contractions in the text are reformed using the contractions python library [63].

Chapter 4

Experiments and Results

In this chapter the experiments and results of our prediction models are discussed.

4.1 Experimental Setup

Every training experiment for each prediction model has been conducted for each window size $w \in W = \{2{,}5{,}10{,}25{,}50{,}100\}$ and for each time interval $i \in I = \{10$ $minutes$, 30 $minutes$, 1$hour$, 2$hours$, 3$hours$, 6$hours$, 12$hours$, 24$hours\}$. There are four different data folds the models get trained and tested on, these different folds are discussed in Section 2.4 and are visualized in Table 3.11. Each of these combinations results in different sizes of the datasets. A dataset split into entries with a time interval of 10 minutes will have more entries than one with a time interval of 24 hours. In Section A.1 more details are found on the different lengths of the training, validation, and test sets for each configuration. On each execution, early stopping is used: the model stops learning after 15 epochs of no improvements in the validation loss.

4.2 Baseline Results

In this section, the results of the baseline models are discussed. Afterward, criteria are established that have to be met by other prediction models to prove that they show improvement over the baseline models.

4.2.1 The Non-Time Differenced Baseline Model $Base_c$

Table 4.1 summarizes the configuration that yields the lowest MSE for the $Base_c$ model for each time interval. If different configurations have the same MSE, the MAE and $R2$ are taken into account to select the best configuration. If these are also equal to each other, both configurations are placed in the table and are separated with a "/".

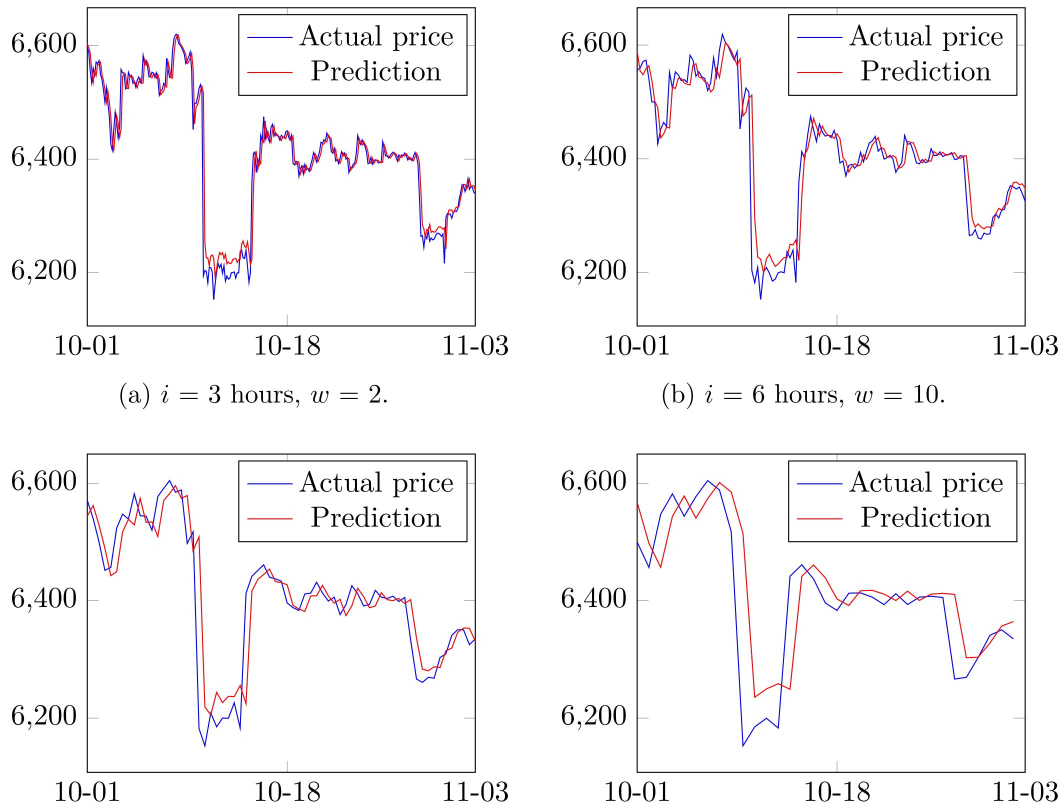
i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	5	4	17.37	465.99	21.59	0.965	69
30 Min.	25	4	11.21	445.37	21.1	0.9665	69
1 Hour	10	4	11.03	587.99	24.25	0.9555	96
2 Hours	2	4	14.1	610.75	24.71	0.9535	77
3 Hours	2	4	16.24	920.04	30.33	0.9297	80
6 Hours	10	4	21.99	1691.42	41.13	0.8687	86
12 Hours	2	4	30.34	3061.56	55.33	0.7695	68
24 Hours	2	4	48.13	6999.32	83.66	0.507	69

Table 4.1: Best results for the $Base_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size, and epochs is the number of epochs the model is trained.

.

Figure 4.1: Prediction of the $Base_c$ model on the test set for a time interval i of 3 hours and a window size w of 2. The test set belongs to fold number 4. In blue, the actual price, and in red, $Base_c$'s predicted price is plotted.

Figure 4.1 illustrates the prediction plot of the best model and the actual price of bitcoin with a time interval i equal to 3 hours. The model learns to achieve an $RMSE$ of 30.33 an $R2$ score of 0.9297, indicating that the model performs well on the prediction task. However, by looking closely into the predicted values, it is clear that the model learns to use the previous price as its prediction for the next price instead of predicting the price at that time. This causes a horizontal gap between the two curves in the figure. Figure 4.3 visualizes four different prediction plots of the $Base_c$ model trained on different time intervals next to each other. Figures 4.2b, 4.2c, and 4.2d illustrate an increase in the gap between the predicted and actual values. The larger the time interval i gets, the larger this gap becomes. The gap size is equal to the time interval i, for example in Figure 4.2a the time interval i is 3 hours and the gap size is 3 hours long because the model predicts the value for the next time interval, which is 3 hours later. In Figure 4.2b, where the time interval is equal to 6 hours, the gap size is 6 hours. The same is true for Figures 4.2c and 4.2d. As suggested in [25], accuracy metrics can be misleading when working with time



(a) i = 3 hours, w = 2.

(b) i = 6 hours, w = 10.

(c) i = 12 hours, w = 2.

(d) i = 24 hours, w = 2.

Figure 4.2: Predictions of the $Base_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Base_c$'s predicted price is plotted. i is the time interval, w is the window size.

series. Notice that the error in Table 4.1 increases for longer time intervals. The price of bitcoin, in general, varies less in 10 minutes than in 24 hours. This means that bitcoin's price 10 minutes ago is on average closer to the current price than bitcoin's price 24 hours ago. Using the previous value as a prediction is closer to the actual ground truth and will result on average in a lower MSE for shorter time intervals.

4.2.2 The Time Differenced Baseline Model $Base_{td}$

Here, we discuss the results for the time differenced baseline model $Base_{td}$. Remember that the difference in close price d_t at time t is defined as follows:

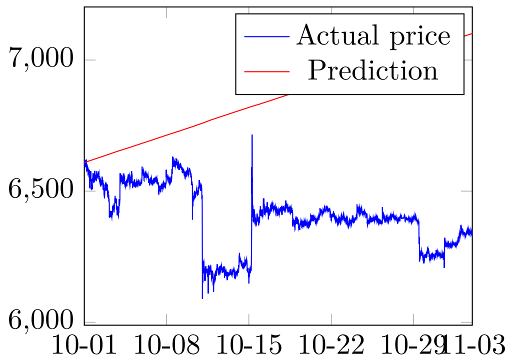
$$d_t = c_t - c_{t-1}$$

with c_t being bitcoin's close price at time t. Again, the experiment is conducted for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.2 gives the best results for each time interval. The extended version of the results for these experiments can be found in Appendix B.1.1. Here, the R2 scores are much lower than those in Table 4.1. According to [25], testing the model on time differenced values is a better way to test a model's ability to predict the future. Notice that the $RMSE$ for the 2-hour time interval is worse compared to the 3 and 6-hour time intervals. We have run the experiment again for the 2-hour time interval to determine if it is coincidental or not. An $RMSE$ of 24 was achieved indicating the bad score in Table 4.2 as being coincidental. Figure 4.3 plots three different prediction plots for the $Base_{td}$'s best model for three different time intervals alongside the actual price. Since the model tries to predict the

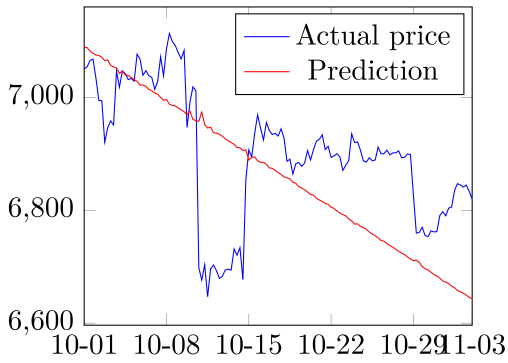
i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	100	4	4.55	91.55	9.57	-0.0002	66
30 Min.	2	4	7.4	263.41	16.23	-0.0005	66
1 Hour	2	4	9.87	466.43	21.6	0.0	76
2 Hours	25	3	29.38	2162.51	46.5	0.0	68
3 Hours	25	4	14.85	870.89	29.51	-0.0001	96
6 Hours	2	4	22.04	1698.91	41.22	-0.0137	77
12 Hours	25	4	30.34	3288.33	57.34	-0.0017	67
24 Hours	5	4	53.95	8075.82	89.87	-0.0582	75

Table 4.2: Best results for the $Base_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

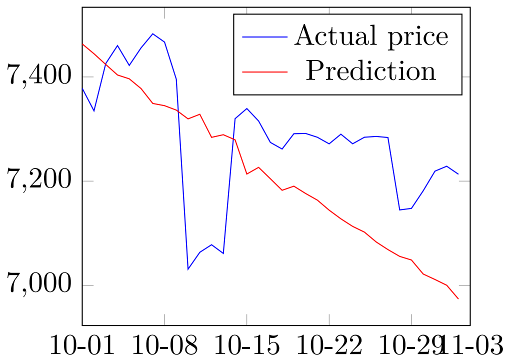
differences in price, the plot that needs to be analyzed should be the one where the real difference in price and the predicted difference in price are plotted. The previous prediction plot's values are recalculated by summing up the differences to each other to obtain the absolute price. The problem there is that the error will be accumulated with time as well. When recalculating the absolute prediction, the previous prediction is summed with the current prediction, this way the error of the previous prediction will be propagated to the next prediction. Figure 4.4 plots the time differenced prediction. Here, we see that most of the time, the model fails to predict any rise or fall in price for most of the data points.



(a) i = 10 minutes, w = 100.



(b) i = 6 hours, w = 50.



(c) i = 24 Hours, w = 50.

Figure 4.3: Predictions of the $Base_{td}$ model on the test set for three different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Base_{td}$'s predicted price is plotted. i is the time interval, w is the window size.

4.2.3 Other Notes

LSTM layers with different hidden sizes such as 64, 32, and 16 have also yielded similar results for both baseline models. The inclusion of the activation function ReLu before the dense layer in the model did not alter the results. Using RMSE as a loss function did not result in any improvements.

4.2.4 Improvement Criteria

There are different ways in which a prediction model can prove better performance than these baseline models: if the close price is used, meaning there is no time differencing applied, the new model's prediction should have a smaller gap that occurs between the predicted price and the ground truth. A lower $RMSE$ would mean that the gap on average is shorter, but the plot should always be considered to analyze different periods individually. If time differencing is applied, the error should be lower. Furthermore, the prediction plots of the models may indicate some improvement as well for different periods.

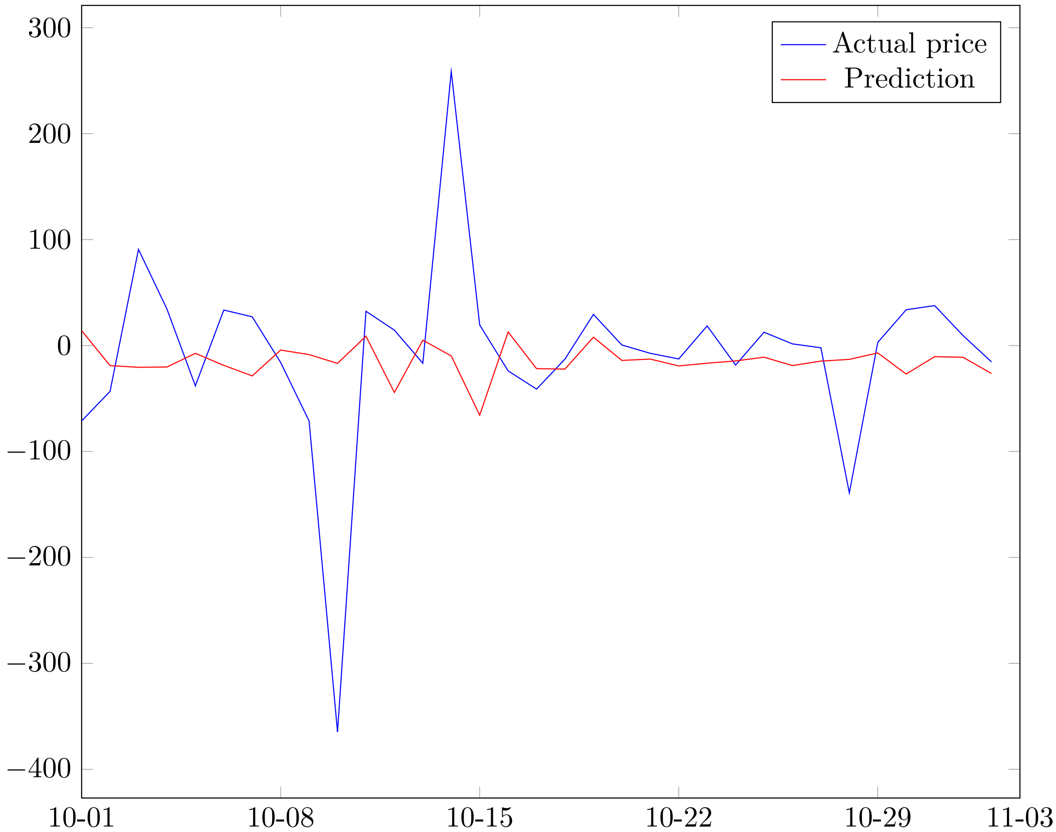


Figure 4.4: Prediction of the $Base_{td}$ model on the test set of the data fold number 4. In blue, the actual price, and in red, $Base_{td}$ model's prediction is plotted. Time interval i is 24 hours, window size w is 50.

4.3 Tweet Sentiment Results

In this section, the results of the prediction models that use Twitter sentiment scores are discussed. We have the individual scores which are the positive, neutral, and negative sentiment scores. Furthermore, there is the compound score which represents the sentiment in one value.

4.3.1 The Non-Time Differenced Sentiment Model $Sent_c$

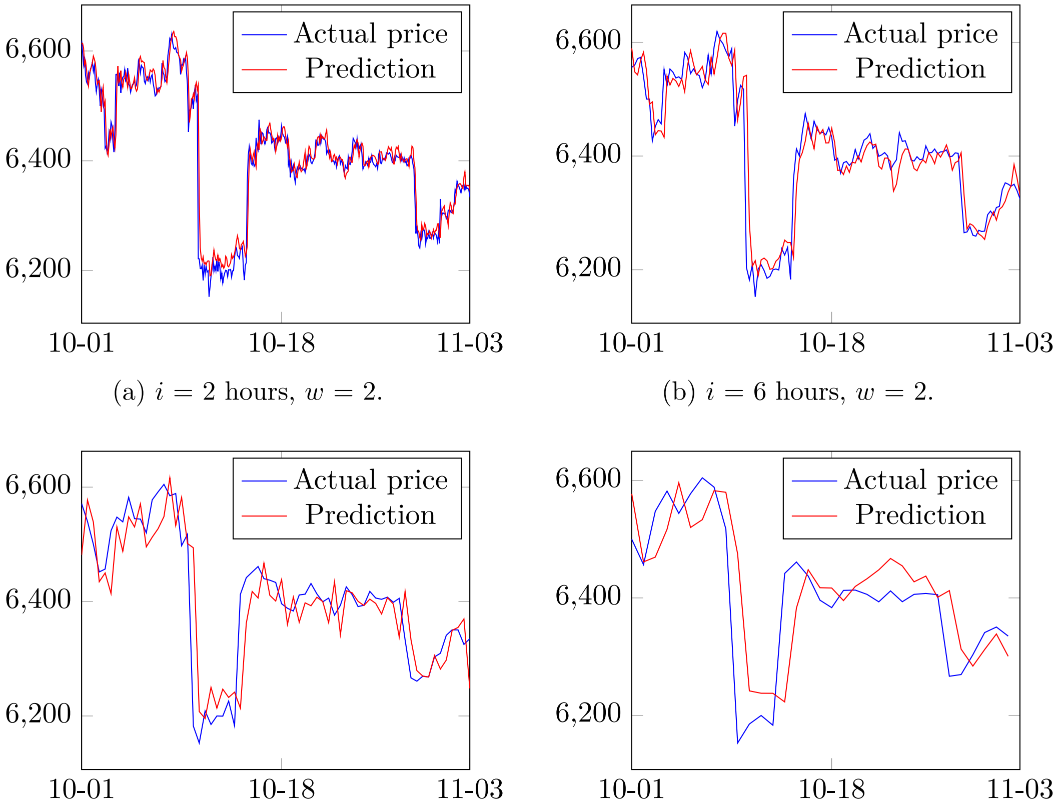
The sentiment model $Sent_c$ takes as input the sets POS, NEU, NEG, and C by concatenating them to each other and min-max normalizing them afterward. The experiment is conducted for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.3 summarizes the best results on the test set for the $Sent_c$ model for each time interval $i \in I$. Fold NR. 4 has the best results since it has the largest training size. Similar to the baseline results, a smaller time interval results in a smaller error and a higher r2 score. No model needed considerably more epochs to train. The larger time intervals, like 12 hours and 24 hours, prefer smaller window sizes since it impacts the size of the training data more drastically when compared to smaller time intervals. More information can be found in Appendix A.1. The complete set of results can be found in Appendix B.2.1. Figure 4.5 illustrates four different prediction plots for the $Sent_c$ model with different time intervals. None of the four plots show any improvement towards closing the gap between the predicted and the actual price. The prediction plots are not delayed copies of the previous price as they are on the plots in Figure 4.3.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	5	4	19.46	610.92	24.72	0.9541	76
30 Min.	10	4	13.44	599.4	24.48	0.9549	79
1 Hour	25	4	23.46	1264.74	35.56	0.9044	74
2 Hours	2	4	15.72	715.06	26.74	0.9455	97
3 Hours	2	4	19.11	1124.92	33.54	0.9141	72
6 Hours	2	4	25.29	1970.96	44.4	0.847	68
12 Hours	2	4	40.88	3877.5	62.27	0.708	73
24 Hours	2	4	56.64	7166.42	84.65	0.4952	67

Table 4.3: Best results for the $Sent_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.3.2 The Time Differenced Sentiment Model $Sent_{td}$

Table 4.4 summarizes the best results for the $Sent_{td}$ model for each time interval $i \in I$. The results are similar to the $Base_{td}$ model except when $i = 24$ hours where the $RMSE$ is more than 20% higher. The complete set of results can be found in



(a) i = 2 hours, w = 2.

(b) i = 6 hours, w = 2.

(c) i = 12 hours, w = 2.

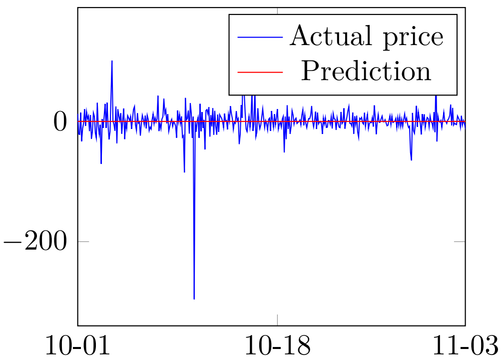
(d) i = 24 hours, w = 2.

Figure 4.5: Predictions of the $Sent_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Sent_c$'s predicted price is plotted. i is the time interval, w is the window size.

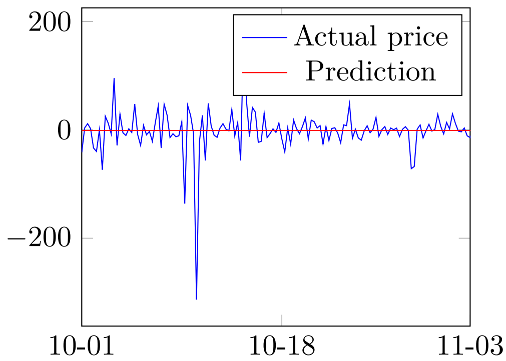
Appendix B.2.2. Figure 4.6 illustrates four different prediction plots for four different intervals. These predictions have the lowest MSE for the selected time interval. Notice that for the 2, 6, and 12-hour time intervals, the best model's prediction is a straight line. For these time intervals, there are models that are not straight lines but they have a higher $RMSE$. Figure 4.7 shows two different prediction plots for the model with a time interval of 12 hours that have a higher $RMSE$ than the straight line in Figure 4.6c but follow the trend better. This issue is discussed further in Section 5.1.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	10	4	4.55	91.56	9.57	-0.0004	67
30 Min.	2	4	7.4	263.42	16.23	-0.0005	72
1 Hour	25	4	9.87	466.43	21.6	0.0	124
2 Hours	2	4	12.84	559.72	23.66	0.0	66
3 Hours	10	4	14.85	870.94	29.51	-0.0002	78
6 Hours	100	4	21.43	1676.45	40.94	-0.0003	67
12 Hours	25	4	30.34	3288.32	57.34	-0.0017	94
24 Hours	2	4	84.58	13008.87	114.06	-0.7046	75

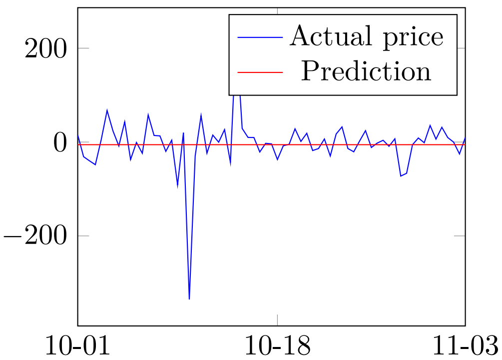
Table 4.4: Best results for the $Sent_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



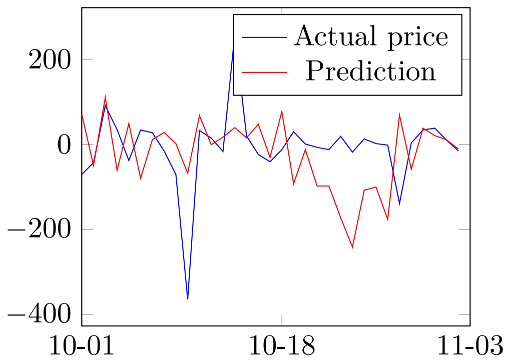
(a) i = 2 hours, w = 2.



(b) i = 6 hours, w = 100.



(c) i = 12 hours, w = 25.



(d) i = 24 hours, w = 2.

Figure 4.6: Predictions of the $Sent_{td}$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Sent_{td}$'s predicted price is plotted. i is the time interval, w is the window size.

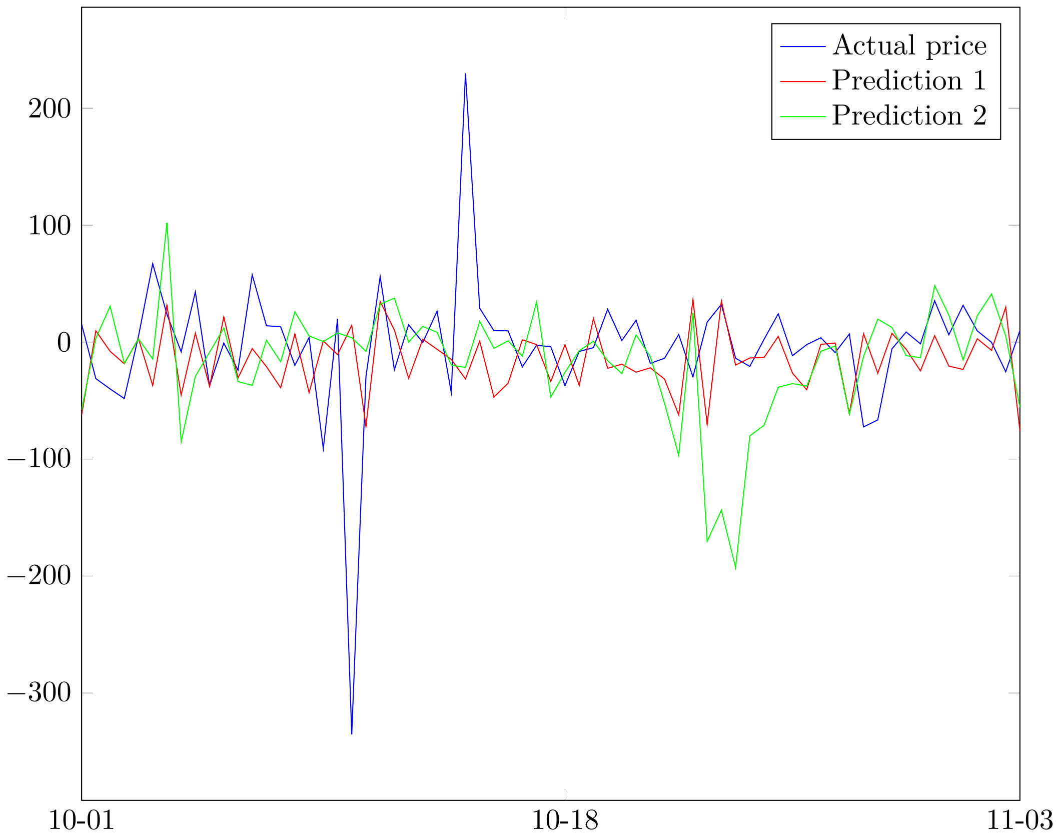


Figure 4.7: Two different predictions of the $Sent_{td}$ model on the test set with time interval of 12 hours. The test set belongs to fold number 4. In blue, the actual price, and in red and green, two different $Sent_{td}$ models' predicted price is plotted. The first prediction (in red) has a window size equal to 2 and an $RMSE$ of 67.48, the second prediction (in green) has a window size equal to 2 and an $RMSE$ of 77.61.

4.3.3 The Non-Time Differenced Compound Sentiment Model $Comp_c$

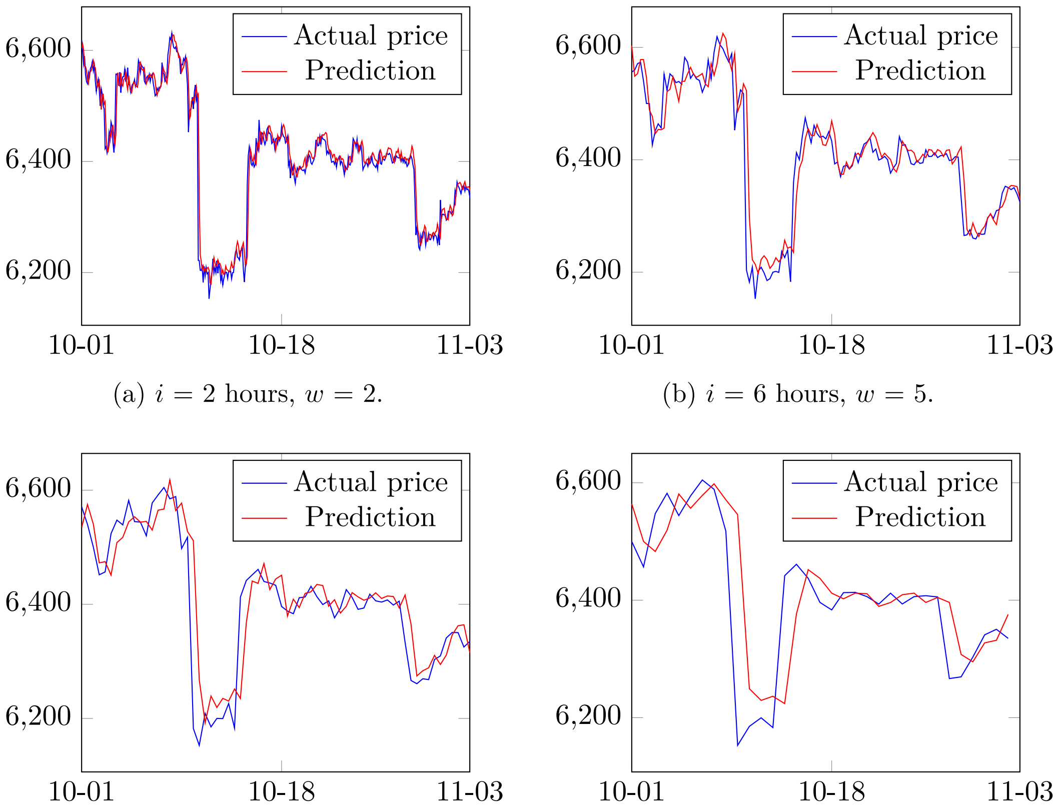
$Comp_c$ takes as input the compound sentiment vector $COMP$ and the close price vector C by concatenating them to each other and min-max normalizing them thereafter. Again, the model is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.5 summarizes the best results for each time interval. The results are worse when compared to the $Base_c$ model. The complete set of results can be found in Appendix B.2.3. Figure 4.8 illustrates four different prediction plots for the $Comp_c$ model with different time intervals. Again, none of these plots show any improvement towards closing the gap between the predicted and the actual price and the results do not show improvements.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	2	4	36.43	1519.12	38.98	0.8859	80
30 Min.	10	4	10.04	414.93	20.37	0.9688	70
1 Hour	5	4	14.08	659.13	25.67	0.9502	77
2 Hours	2	4	14.39	645.36	25.4	0.9508	75
3 Hours	2	4	17.21	1027.17	32.05	0.9216	83
6 Hours	5	4	23.11	1788.61	42.29	0.8612	66
12 Hours	2	4	34.59	3389.47	58.22	0.7448	92
24 Hours	2	4	49.36	7898.68	88.87	0.4436	74

Table 4.5: Best results for the $Comp_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.3.4 The Time Differenced Compound Sentiment Model $Comp_{td}$

$Comp_{td}$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.6 summarizes the best results for each time interval. The complete set of results can be found in Appendix B.2.4. Figure 4.9 illustrates four different prediction plots for the $Comp_{td}$ model with different time intervals. Notice that for smaller time intervals ($i < 12\ hours$), the model uses a straight line as its prediction and achieves a lower MSE compared to non-straight lines. In Figure 4.10, two predictions are plotted of the $Comp_{td}$ model where in Figure 4.10a the model prefers a straight line and in Figure 4.10b the model learns a non-straight line.



(a) i = 2 hours, w = 2.

(b) i = 6 hours, w = 5.

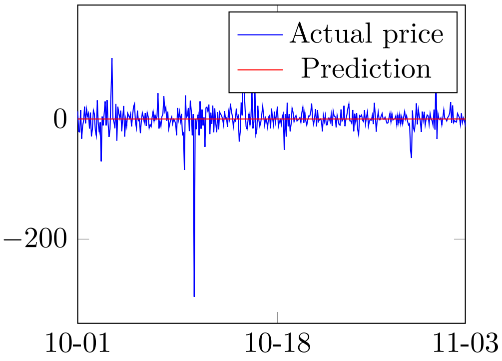
(c) i = 12 hours, w = 2.

(d) i = 24 hours, w = 2.

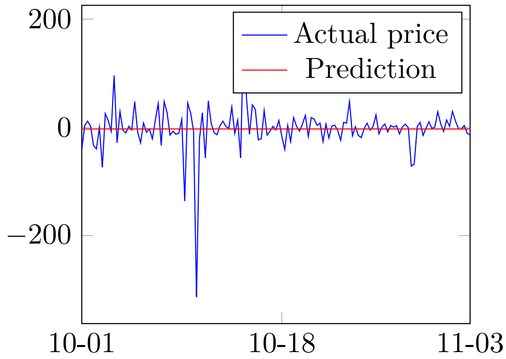
Figure 4.8: Predictions of the $Comp_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Comp_c$'s predicted price is plotted. i is the time interval, w is the window size.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	5	4	4.56	91.65	9.57	-0.0013	67
30 Min.	100	4	7.37	263.3	16.23	0.0	94
1 Hour	2/5/50	4	9.87	466.43	21.6	0.0	66
2 Hours	2/10/100	4	12.84	559.72	23.66	0.0	86
3 Hours	25	4	14.85	870.93	29.51	-0.0002	76
6 Hours	10	4	21.65	1676.91	40.95	-0.0006	69
12 Hours	50	4	29.7	3284.41	57.31	-0.0005	70
24 Hours	2	4	73.17	10657.6	103.24	-0.3965	66

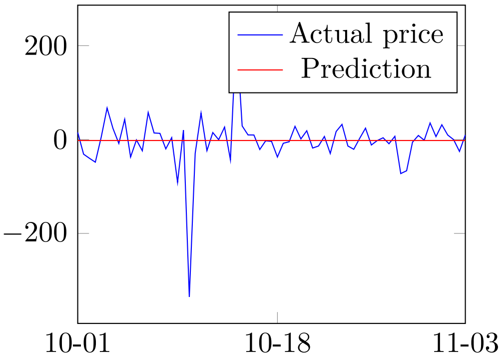
Table 4.6: Best results for the $Comp_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



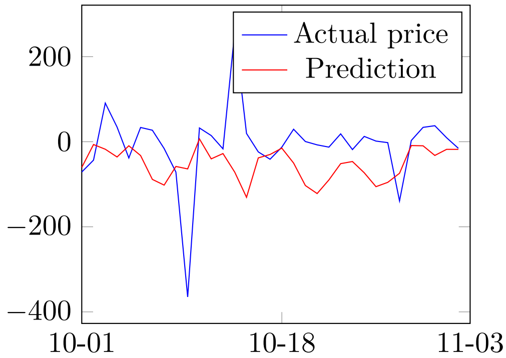
(a) i = 2 hours, w = 2.



(b) i = 6 hours, w = 10.

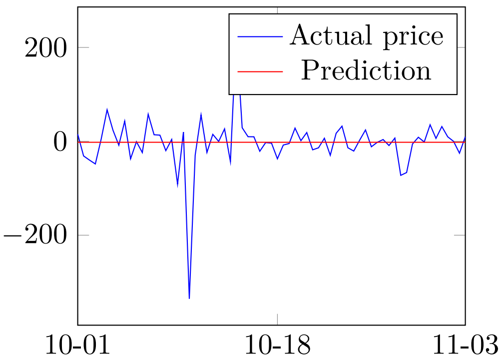


(c) i = 12 hours, w = 50.

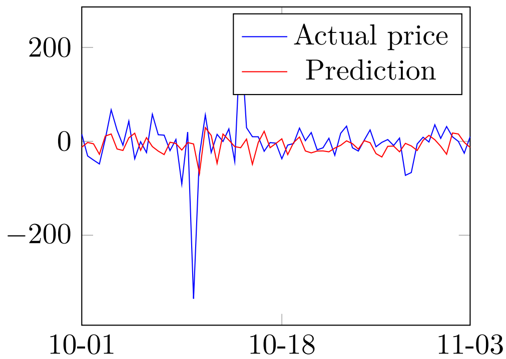


(d) i = 24 hours, w = 2.

Figure 4.9: Predictions of the $Comp_{td}$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $Comp_{td}$'s predicted price is plotted. i is the time interval, w is the window size.



(a) w = 50, RMSE = 57.31.



(b) w = 2, RMSE = 59.28.

Figure 4.10: Two different predictions of the $Comp_{td}$ model on the test set with a 12-hour time interval. The test set belongs to the data fold number 4. Window size differs w for the predictions and the set fold number is 4. In blue, the actual price, and in red, $Comp_{td}$'s predicted price is plotted.

4.4 Tweet Volume Results

In this section, the results of the prediction models that use tweet volume based features are discussed. The different types of tweet volume based features are found in Section 3.1.3.

4.4.1 The Non-Time Differenced Original Tweet Volume Model $VolumeOrg_c$

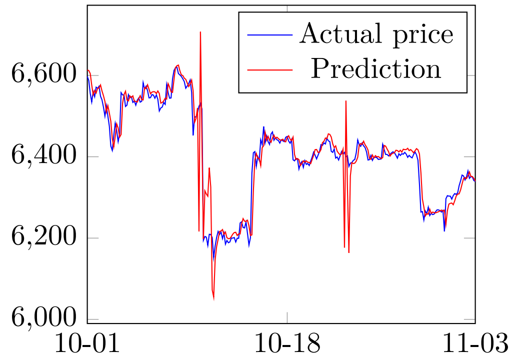
$VolumeOrg_c$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.7 summarizes the best results for each time interval. Notice that for a time interval of 6, 12, and 24 hours, the fold with the best result is fold number 3. This is different compared to the previous results where evaluation on the 4th fold NR. achieved the best results. The best $RMSE$ values for the $VolumeOrg_c$ model are similar or worse than the ones for the $Base_c$ model. The complete set of results can be found in Appendix B.3.1. Figure 4.9 illustrates four different prediction plots for the $VolumeOrg_c$ model with different time intervals. On the first prediction plot, Figure 4.8a, there are some deviations from the actual price compared to the other plots that are delayed copies of the actual price. The model seems to learn a trend that is misleading in the test set. These fluctuations seem to be the result of the high increase in $volorg$ around these periods, see Figure 4.12.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	5	4	26.43	1018.25	31.91	0.9235	99
30 Min.	2	4	13.25	449.94	21.21	0.9661	77
1 Hour	2	4	15.45	668.07	25.85	0.9495	75
2 Hours	2	4	16.9	2298.22	47.94	0.825	103
3 Hours	5	4	22.91	1911.16	43.72	0.854	85
6 Hours	2	3	56.13	6915.12	83.16	0.9268	73
12 Hours	5	3	84.83	13068.64	114.32	0.8503	70
24 Hours	2	3	108.7	24889.16	157.76	0.6991	70

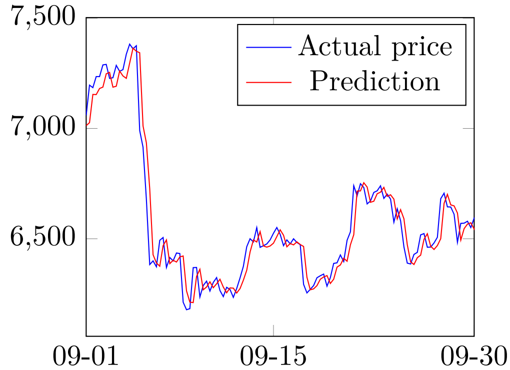
Table 4.7: Best results for the $VolumeOrg_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.4.2 The Time Differenced Original Tweet Volume Model $VolumeOrg_{td}$

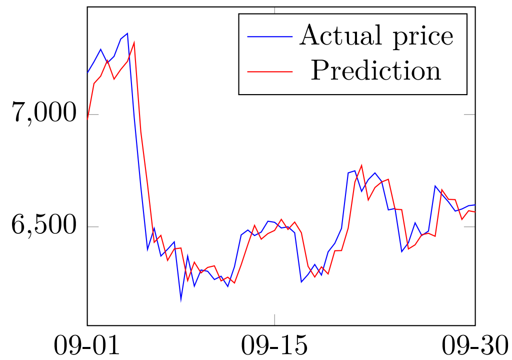
The time differenced original tweet volume model $VolumeOrg_{td}$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.8 summarizes the best results for each time interval. The complete set of results can be found in Appendix B.3.2. Figure 4.9 illustrates four different



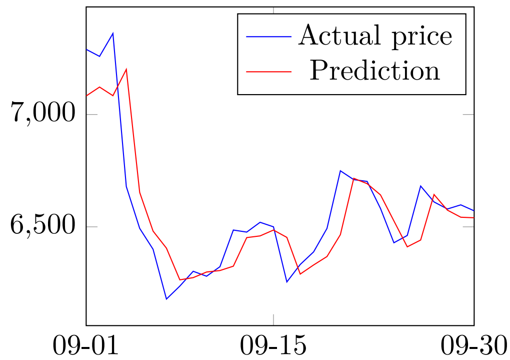
(a) i = 3 hours, w = 5, Fold NR. = 4.



(b) i = 6 hours, w = 2, Fold NR. = 3.



(c) i = 12 hours, w = 5, Fold NR. = 3.



(d) i = 24 hours, w = 2, Fold NR. = 3.

Figure 4.11: Predictions of the $VolumeOrg_c$ model on the test set for four different time intervals. Fold NR. indicates to which fold the test set belongs. In blue, the actual price, and in red, $VolumeOrg_c$'s predicted price is plotted. i is the time interval, w is the window size.

prediction plots for the $VolumeOrg_{td}$ model with different time intervals. Notice that for the 24-hour time interval, the fold number is 3.

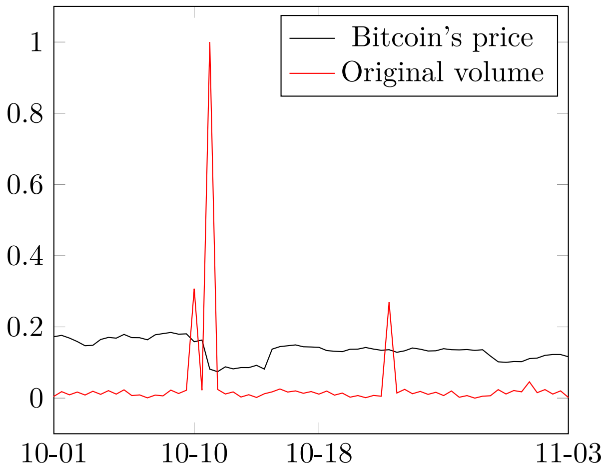
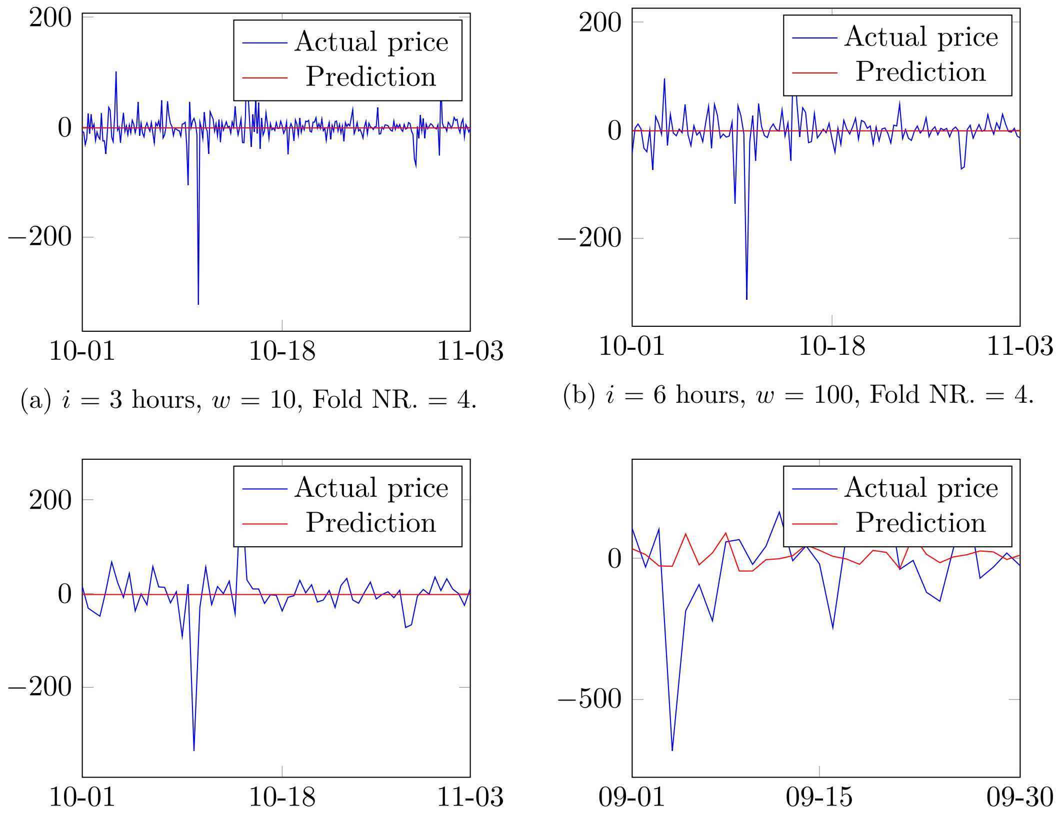


Figure 4.12: The normalized actual price (black) and the normalized original tweet volume (red) plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is the test set of the 4th fold.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	50	4	4.56	91.64	9.57	-0.0012	82
30 Min.	100	4	7.36	263.35	16.23	-0.0002	71
1 Hour	100	4	9.87	466.42	21.6	-0.0	74
2 Hours	2/10/100	4	12.84	559.72	23.66	-0.0	78
3 Hours	10	4	14.85	870.95	29.51	-0.0002	76
6 Hours	100	4	21.43	1676.44	40.94	-0.0003	68
12 Hours	50	4	29.7	3284.41	57.31	-0.0005	66
24 Hours	5	3	115.34	29117.82	170.64	-0.0297	68

Table 4.8: Best results for the $VolumeOrg_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



(a) i = 3 hours, w = 10, Fold NR. = 4.

(b) i = 6 hours, w = 100, Fold NR. = 4.

(c) i = 12 hours, w = 50, Fold NR. = 4.

(d) i = 24 hours, w = 5, Fold NR. = 3.

Figure 4.13: Predictions of the $VolumeOrg_{td}$ model on the test set for four different time intervals. Fold NR. indicates to which fol the test set belongs. In blue, the actual price, and in red, $VolumeOrg_{td}$'s predicted price is plotted. i is the time interval, w is the window size.

4.4.3 The Non-Time Differenced Filtered Tweet Volume Model $VolumeFilt_c$

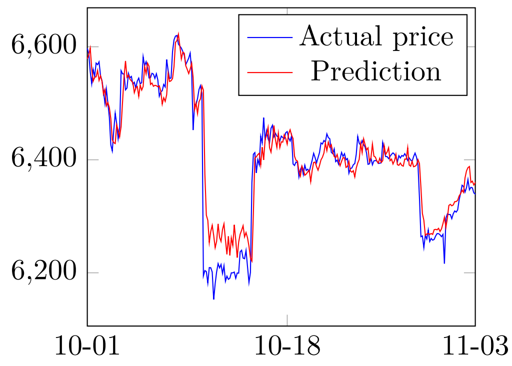
$VolumeFilt_c$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.9 summarizes the best results for each time interval. The complete set of results can be found in Appendix B.3.3. Figure 4.14 illustrates four different prediction plots for the $VolumeFilt_c$ model with different time intervals. Most of the results are similar or worse to those of the $Base_c$ model except for the 24 hours time interval. There, the $RMSE$ is more than 11% lower. Observing Figure 4.14d, the model's prediction is still lagging behind most of the time.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	5	4	31.6	1232.77	35.11	0.9074	83
30 Min.	10	4	11.99	491.18	22.16	0.963	91
1 Hour	5	4	11.56	577.94	24.04	0.9563	89
2 Hours	2	4	13.51	588.63	24.26	0.9552	73
3 Hours	2	4	16.19	907.79	30.13	0.9307	74
6 Hours	2	4	23.44	1709.61	41.35	0.8673	68
12 Hours	2	4	35.19	3566.74	59.72	0.7314	72
24 Hours	5	4	48.62	5443.18	73.78	0.6166	67

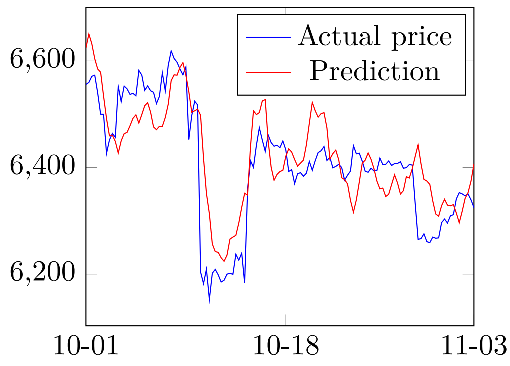
Table 4.9: Best results for the $VolumeFilt_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.4.4 The Time Differenced Filtered Tweet Volume Model $VolumeFilt_{td}$

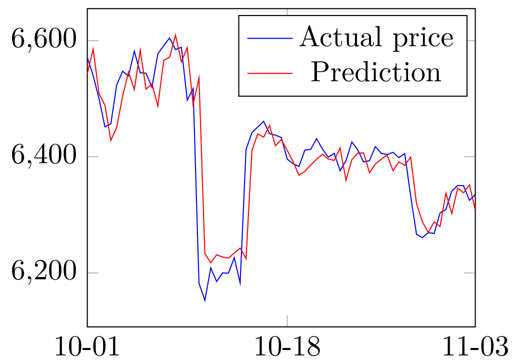
$VolumeFilt_{td}$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.10 summarizes the best results for each time interval. The complete set of results can be found in Appendix B.3.4. The results are similar to the $Base_{td}$ model, there is a slight improvement of over 5% on the $RMSE$ of the 24-hour time interval. Figure 4.15 illustrates four different prediction plots for the $VolumeFilt_{td}$ model with different time intervals. The first three prediction plots are straight lines. The fourth prediction plot does not show an indication of some trend that is followed by the model to help predict the actual price.



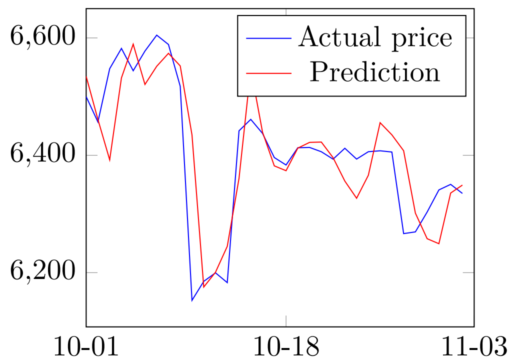
(a) i = 3 hours, w = 25.



(b) i = 6 hours, w = 100.



(c) i = 12 hours, w = 2.

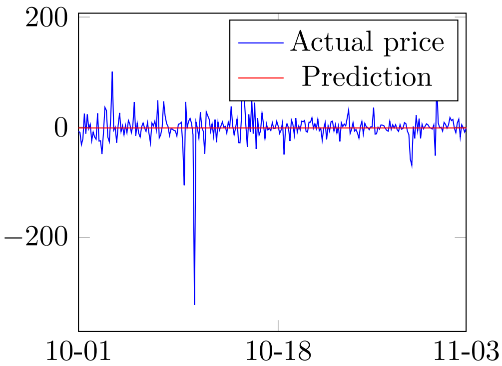


(d) i = 24 hours, w = 2.

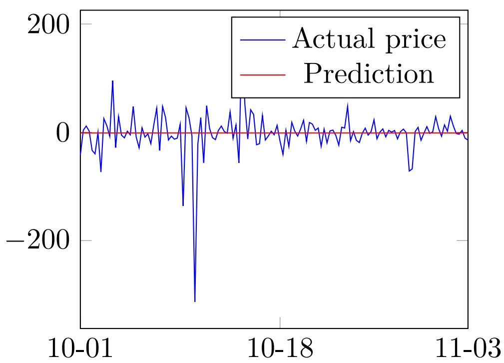
Figure 4.14: Predictions of the $VolumeFilt_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $VolumeFilt_c$'s predicted price is plotted. i is the time interval, w is the window size.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	10	4	4.55	91.56	9.57	-0.0003	99
30 Min.	5/25	4	7.38	263.29	16.23	0.0	67/78
1 Hour	5/50	4	9.87	466.42	21.6	0.0	69/79
2 Hours	2/10/100	4	12.84	559.72	23.66	0.0	69/67/67
3 Hours	25	4	14.85	870.94	29.51	-0.0002	124
6 Hours	100	4	21.43	1676.42	40.94	-0.0003	81
12 Hours	50	4	29.7	3284.43	57.31	-0.0005	72
24 Hours	10	4	46.97	7059.04	84.02	0.075	81

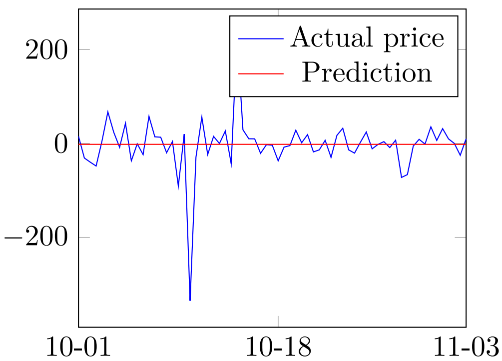
Table 4.10: Best results for the $VolumeFilt_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



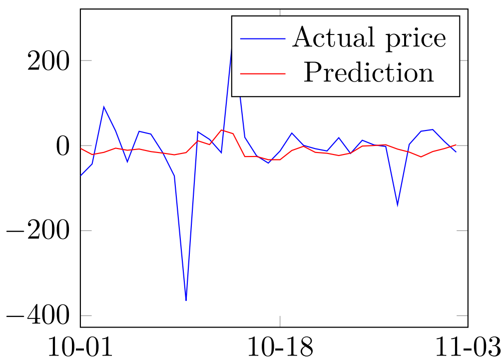
(a) $i = 3$ hours, $w = 25$.



(b) $i = 6$ hours, $w = 100$.



(c) $i = 12$ hours, $w = 50$.



(d) $i = 24$ hours, $w = 10$.

Figure 4.15: Predictions of the $VolumeFilt_{td}$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $VolumeFilt_{td}$'s predicted price is plotted. i is the time interval, w is the window size.

4.4.5 The Non-Time Differenced Sentiment Volume Model $VolumeSent_c$

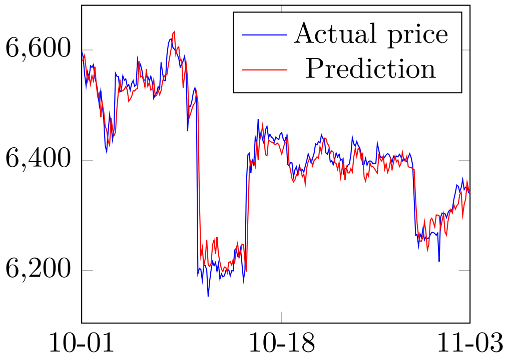
$VolumeSent_c$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.11 summarizes the best results for each time interval. The results are similar to the $Base_c$ model and there is no sign of improvement. The complete set of results can be found in Appendix B.3.5. Figure 4.16 illustrates four different prediction plots for the $VolumeSent_c$ model with different time intervals. These prediction plots are most of the time delayed copies of the actual price with some fluctuations. Looking further into the prediction plots in Figures 4.16c and 4.16d, they seem to be exact copies of the previous price up until 18 October. Afterward, the model matches the actual prediction for two or three data points. This is not the case for the prediction plots of the $Base_c$ model.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	10	4	31.72	1444.79	38.01	0.8915	112
30 Min.	2	4	10.76	383.85	19.59	0.9711	76
1 Hour	2	4	19.25	797.09	28.23	0.9397	67
2 Hours	2	4	15.76	703.22	26.52	0.9464	75
3 Hours	10	4	22.04	1223.73	34.98	0.9065	111
6 Hours	2	4	24.31	1764.19	42.0	0.8631	110
12 Hours	2	4	39.56	3776.58	61.45	0.7156	70
24 Hours	2	4	49.39	6702.82	81.87	0.5278	78

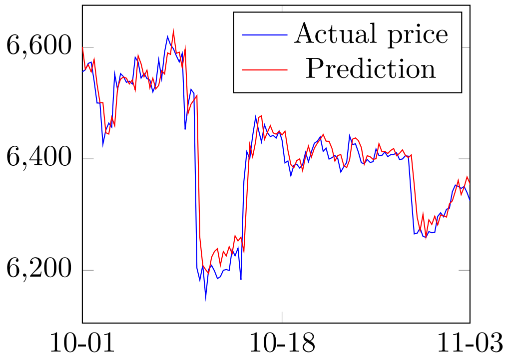
Table 4.11: Best results for the $VolumeSent_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.4.6 The Time Differenced Sentiment Volume Model $VolumeSent_{td}$

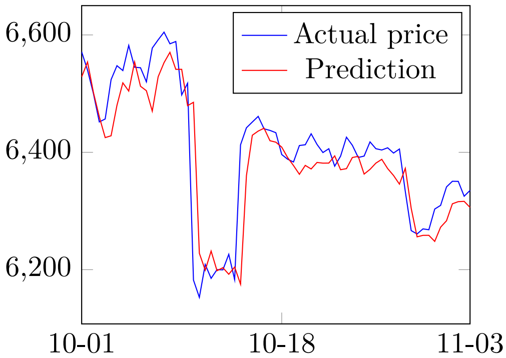
$VolumeSent_{td}$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.12 summarizes the best results for each time interval. These results have similar $RMSE$ values as $Base_{td}$. However, for the 2 and 24-hour time intervals, the $VolumeSent_{td}$ model performs better. The complete set of results can be found in Appendix B.3.6. Figure 4.17 illustrates four different prediction plots for the $VolumeSent_{td}$ model with different time intervals. The first three prediction plots are straight lines, the fourth, however, indicates that there is some trend that the model has learned. This is discussed further in Section 5.2.



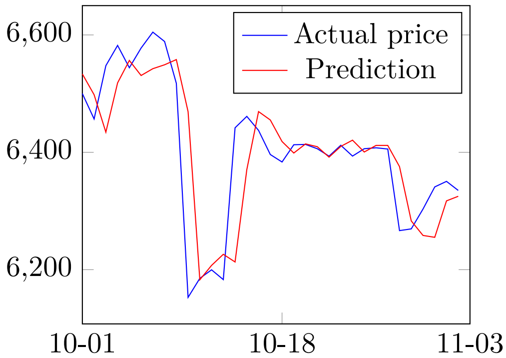
(a) i = 3 hours, w = 10.



(b) i = 6 hours, w = 2.



(c) i = 12 hours, w = 2.

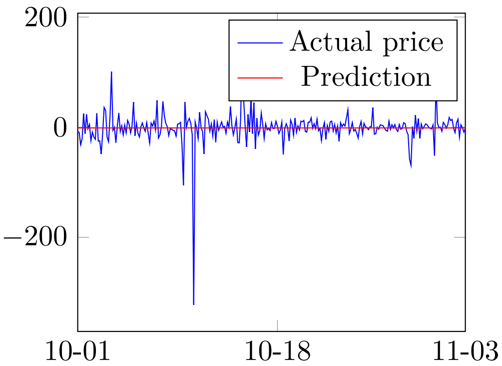


(d) i = 24 hours, w = 2.

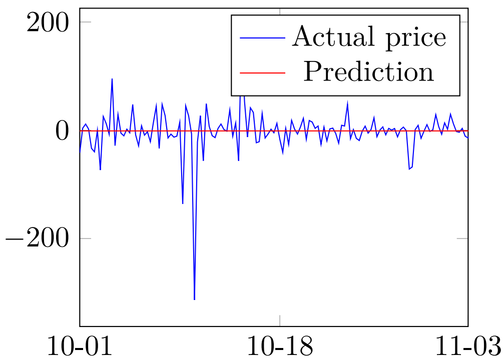
Figure 4.16: Predictions of the $VolumeSent_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $VolumeSent_c$'s predicted price is plotted. i is the time interval, w is the window size.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	10	4	4.55	91.54	9.57	-0.0001	178
30 Min.	10/25	4	7.38	263.29	16.23	-0.0	67/66
1 Hour	50/100	4	9.87	466.42	21.6	-0.0	69/69
2 Hours	2/5/10	4	12.84	559.72	23.66	-0.0	98/70/72
3 Hours	25	4	14.85	870.92	29.51	-0.0002	101
6 Hours	100	4	21.43	1676.44	40.94	-0.0003	80
12 Hours	50	4	29.7	3284.38	57.31	-0.0004	82
24 Hours	10	4	56.2	6968.35	83.48	0.0869	91

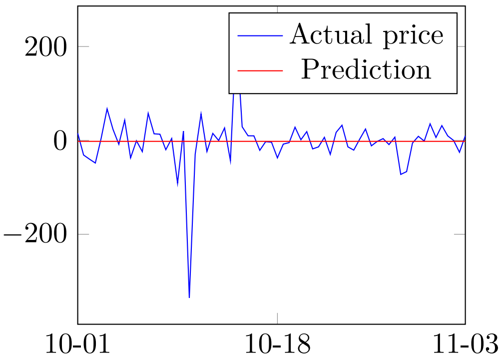
Table 4.12: Best results for the $VolumeSent_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



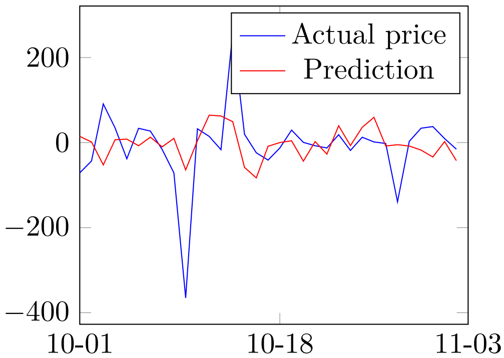
(a) i = 3 hours, w = 25.



(b) i = 6 hours, w = 100.



(c) i = 12 hours, w = 50.



(d) i = 24 hours, w = 10.

Figure 4.17: Predictions of the $VolumeSent_{td}$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $VolumeSent_{td}$'s predicted price prediction is plotted. i is the time interval, w is the window size.

4.5 BERT Vector Results

In this section, the models use the BERT feature $\overline{BERT}$ as input. This vector is the (weighted) average of all the BERT vectors for that time interval, more in Section 3.1.3.

4.5.1 The Non-Time Differenced Dense-BERT Model $DenseBERT_c$

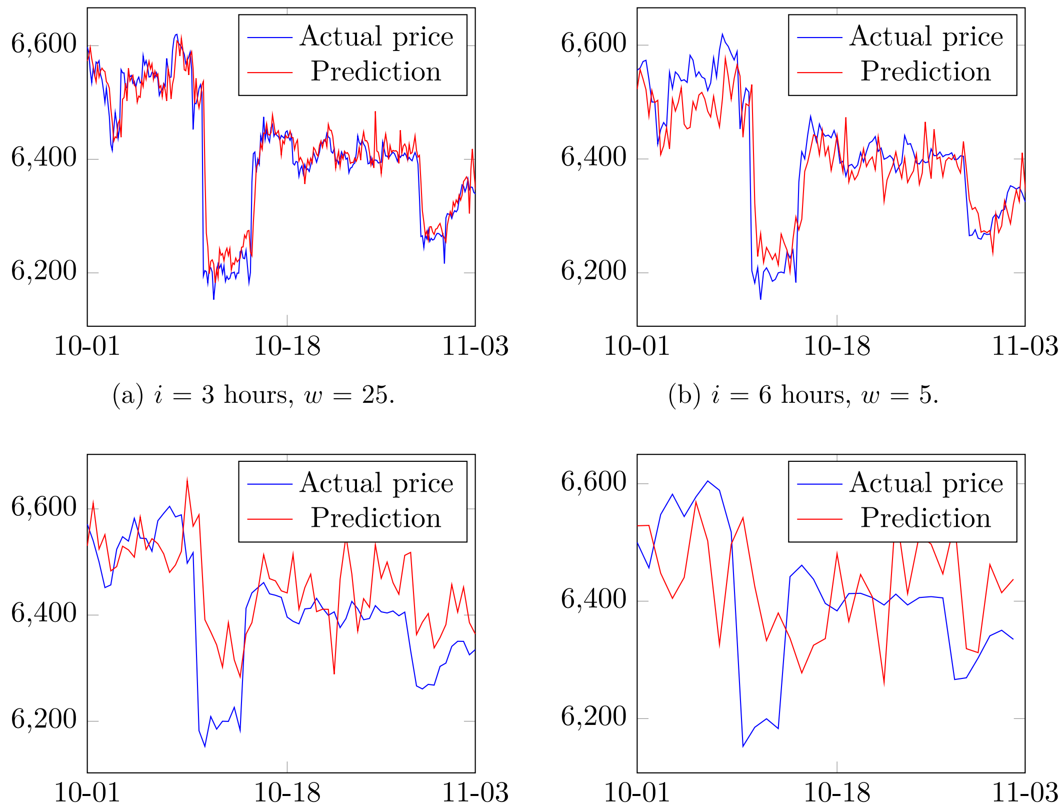
$DenseBERT_c$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.13 summarizes the best results for each time interval. There is no improvement over the $Base_c$ model in the result metrics, the $RMSE$ is worse for almost each time interval. The complete set of results can be found in Appendix B.4.1. Figure 4.18 illustrates four different prediction plots for the $DenseBERT_c$ model with different time intervals. These prediction plots deviate largely from the actual price in Figures 4.18c and 4.18d.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	100	4	46.72	2852.59	53.41	0.7858	99
30 Min.	10	4	19.48	907.59	30.13	0.9317	82
1 Hour	25	4	17.73	954.84	30.9	0.9278	85
2 Hours	25	4	18.81	1011.45	31.8	0.923	68
3 Hours	25	4	23.84	1491.67	38.62	0.8861	72
6 Hours	2	4	41.99	3323.41	57.65	0.7421	69
12 Hours	2	4	77.66	10117.09	100.58	0.2382	67
24 Hours	5	4	109.64	19393.54	139.26	-0.3661	87

Table 4.13: Best results for the $DenseBERT_c$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.

4.5.2 The Time Differenced Dense-BERT Model $DenseBERT_{td}$

$DenseBERT_{td}$ is trained for each window size $w \in W$, for each time interval $i \in I$, and for each of the four different data folds. Table 4.14 summarizes the best results for each time interval. These are similar to the results of $Base_{td}$ because $DenseBERT_{td}$ also prefers straight lines. The complete set of results can be found in Appendix B.4.2. Figure 4.19 illustrates four different prediction plots for the $DenseBERT_{td}$ model with different time intervals. Straight lines are seen on the first three prediction plots and the fourth is close to being a straight line. The models fail to follow the trend of bitcoin's price. These results are discussed further in detail in Section 5.3.



(a) i = 3 hours, w = 25.

(b) i = 6 hours, w = 5.

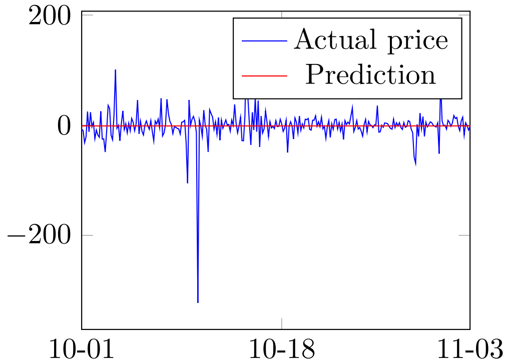
(c) i = 12 hours, w = 2.

(d) i = 24 hours, w = 5.

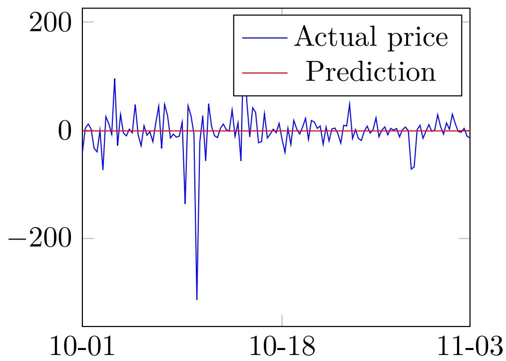
Figure 4.18: Predictions of the $DenseBERT_c$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $DenseBERT_c$'s predicted price is plotted. i is the time interval, w is the window size.

i	w	Fold NR.	MAE	MSE	RMSE	R2	Epochs
10 Min.	10	4	4.56	91.63	9.57	-0.0011	147
30 Min.	10/25	4	7.37	263.29	16.23	0.0	82/71
1 Hour	5/50/100	4	9.87	466.42	21.6	0.0	82/71/68
2 Hours	2/10/100	4	12.84	559.72	23.66	0.0	111/90/88
3 Hours	25	4	14.85	870.92	29.51	-0.0002	80
6 Hours	100	4	21.43	1676.48	40.94	-0.0003	76
12 Hours	50	4	29.7	3284.42	57.31	-0.0005	66
24 Hours	25	4	47.47	7654.44	87.49	-0.003	71

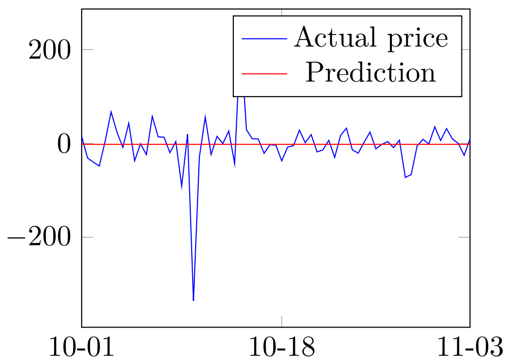
Table 4.14: Best results for the $DenseBERT_{td}$ model for each time interval $i \in I$. The fold NR. indicates the fold number from Table 3.11, w is the window size and, epochs is the number of epochs the model is trained.



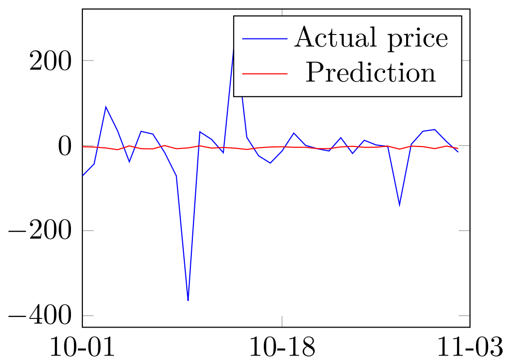
(a) i = 3 hours, w = 25.



(b) i = 6 hours, w = 100.



(c) i = 12 hours, w = 50.



(d) i = 24 hours, w = 25.

Figure 4.19: Predictions of the $DenseBERT_{td}$ model on the test set for four different time intervals. The test set belongs to fold number 4. In blue, the actual price, and in red, $DenseBERT_{td}$'s predicted price is plotted. i is the time interval, w is the window size.

Chapter 5

Discussion

In this section, the results are discussed and the research questions from Section 1.3 are revisited.

5.1 Tweet Sentiment's Ability to Predict Bitcoin's Price

The first research question RQ1 is as follows: "Is there a difference between using the VADER sentiment compound score and using the three individual scores (the negative, neutral, and positive sentiment scores) as input features for the prediction model?" Recall that [34] and [44] claim that sentiment helps predict bitcoin's price and [42] claims the contrary. There is, however, no direct comparison between the compound sentiment score and the individual sentiment scores. We have expected a slight improvement of 1%-2% in $RMSE$ of the prediction models that use the individual sentiment scores. In this section, we will discuss if tweet sentiment can predict bitcoin's price and look into the difference between the prediction models that use the individual sentiment scores, $Sent_c$ and $Sent_{td}$, and the prediction models that use the compound sentiment score, $Comp_c$ and $Comp_{td}$. At last, we look into the differences in the prediction of both types of prediction models and we formulate an answer to our research question RQ1.

5.1.1 Does Twitter Sentiment Predict Bitcoin's Price?

In order for the non-time differenced models to show improvement over the baseline model, the gap between the prediction and the actual price must be smaller. If the $RMSE$ of the prediction models is considered, the sentiment models, $Sent_c$ and $Comp_c$, are unable to show any improvement over the baseline model $Base_c$. However, there is a difference between the prediction plots of the $Base_c$ model and the prediction plots of the sentiment models. The $Base_c$ model's prediction curve follows the shape of the actual price while the sentiment models' prediction curves have more fluctuations. They deviate more from the shape of the actual price. These differences in shape become clear when comparing Figure 4.2c with

Figure 4.5c and 4.8c. A possible explanation is that the sentiment features introduce noise or the trend in the test set is different from the trend in the training and/or validation set. Figure 5.1 plots both the normalized average compound score and the normalized close price of bitcoin. Both are min-max normalized. Figure 5.1 does not indicate a correlation between the tweet sentiment and the close price. However, on 11 October where there is a significant decrease in bitcoin's price (marked with the green line in Figure 5.1), there is a decrease in the compound score. Note that the change in the sentiment scores is delayed by one data point. Similarly, other equally sized changes in bitcoin's price are also represented in the fluctuations of the compound score, but not every change in the compound score is reflected in bitcoin's price. This may indicate that a change in bitcoin's price causes a change in the compound sentiment score but further research is needed. The size of the change in the compound score, however, does not indicate the size of the difference in bitcoin's price. Figure 5.2 plots the individual sentiment scores alongside bitcoin's

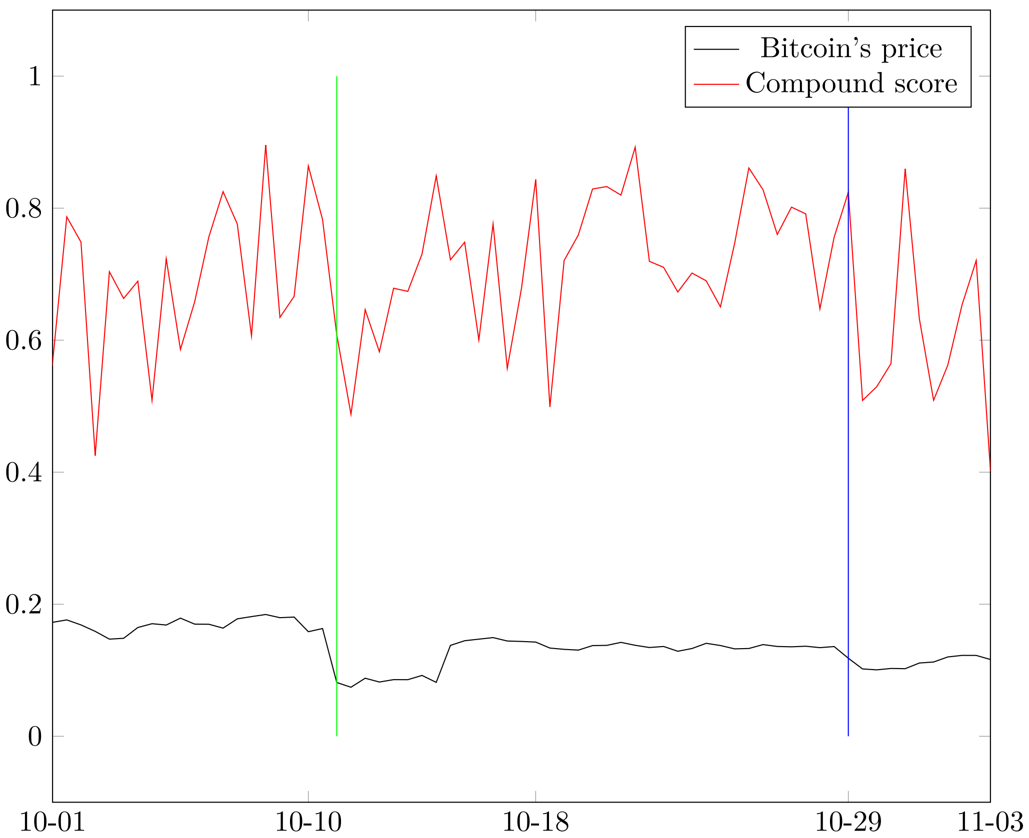


Figure 5.1: The normalized actual price and the normalized compound score are plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The green and blue lines mark two different points where a sudden change in bitcoin price has taken place. The plotted period is the test set of the 4th data fold.

price. While bitcoin's price is dropping on 11 October, the negative sentiment score increases, the positive sentiment score decreases, and the neutral sentiment score remains stable. Interestingly, from 16 October until 29 October, when the price remains stable, the neutral sentiment score is on average higher than the negative and positive sentiment score. We believe that these individual sentiment scores have the potential to improve the predictions since they can provide more detailed information regarding the sentiment. More research should be conducted analyzing these individual scores further.

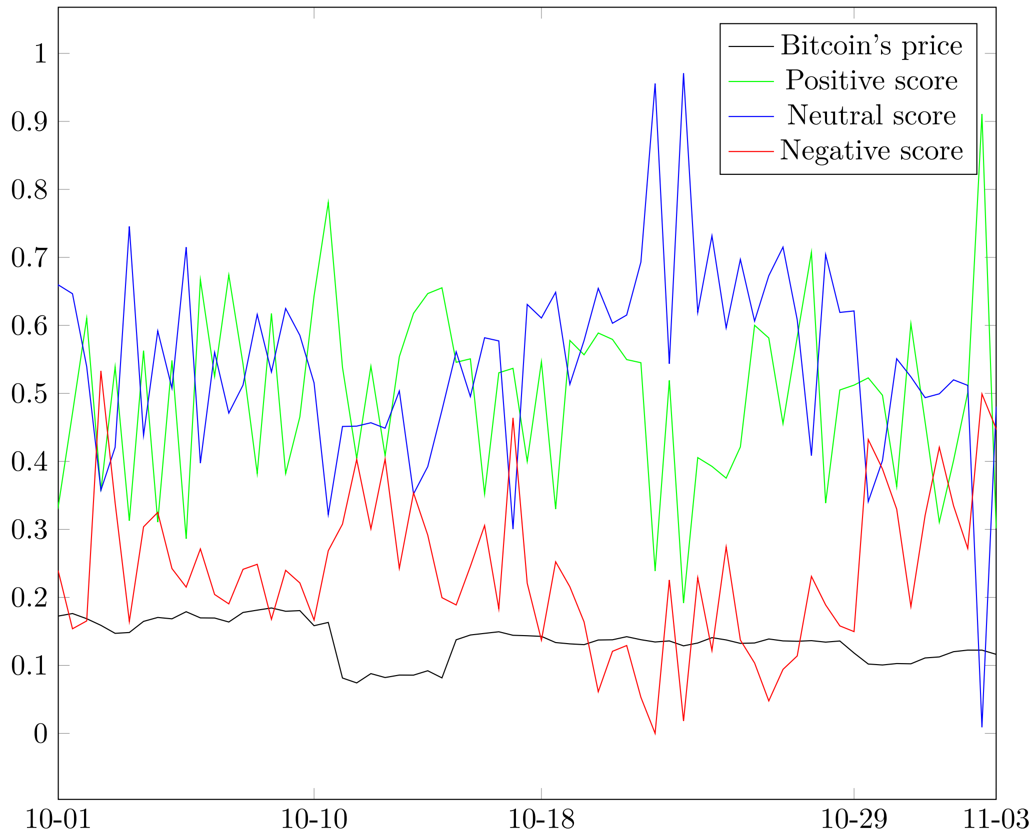


Figure 5.2: The normalized actual price, the normalized positive sentiment score, the neutral sentiment score, and the negative sentiment score are plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is the test set of the 4th fold.

Feature	10Min		6h		24h	
	r	p	r	p	r	p
Positive score	-0.0432	0.0001>	-0.079	0.0142	-0.1123	0.0825
Neutral score	0.0469	0.0001>	0.1134	0.0004	0.1604	0.0128
Negative score	-0.0182	0.0007	-0.0689	0.0326	-0.0902	0.1635
Compound score	-0.0339	0.0001>	-0.0432	0.18	-0.0592	0.3616

Table 5.1: The values for the Pearson correlation coefficient, r, and corresponding p-value, p, with the tweet sentiment features and bitcoin's close price for three different time intervals.

Table 5.1 gives the Pearson correlation coefficient, r, and corresponding p-values, p, for the correlation between the sentiment features and bitcoin's close price. These values indicate a poor linear correlation. Recall from Section 2.5, that the Pearson correlation coefficient, r, an indication is of both the strength, expressed in the value, and the direction, expressed in the sign, of a linear relationship. The p-value, p, states the significance of the correlation, The closer to 0, the more certain the correlation indicated by r is [9].

What Do these Tweets Look Like?

Table 5.2 gives the most impactful tweets on 10 October which is a day before a significant decrease in bitcoin's price. The users that posted these tweets have the most followers on that day which defines the impact of a tweet. Recall that a weighted sentiment score is taken with the weight being the number of followers. All but one tweet is posted by news organizations or some other group. One tweet has been posted by an individual which is in this case a celebrity. Notice that these tweets give no indication of the fall that bitcoin's price will go through the next day. Figure 5.1 shows that the dip in the compound score happens after the price has dropped. The most impactful tweets on the day after the crash, 11 October, have a lower compound score in comparison. They are simply reports of the crash of bitcoin's price and have therefore a more negative sentiment score. Tweets like "Nearly $13 billion wiped off of cryptocurrency market as major coins plunge" and "Bitcoin slumps more than 5%, puts 'digital gold' status in jeopardy" are in the top ten most impactful tweets.

5.1.2 What about Predicting the Difference in Price?

Looking further into the time differenced models, the sentiment models, $Sent_{td}$ and $Comp_{td}$, are unable to show any improvement over the $Base_{td}$ model. Figure 5.3 illustrates three different prediction plots for each of these models. Even if the $Base_{td}$ model has the lowest RMSE between them, the other models have a better fit when their corresponding plots are compared. The $Base_{td}$'s prediction with the lowest $RMSE$ tends to have fewer fluctuations and tends to follow the trend of the price less than the sentiment models' prediction plots with the lowest $RMSE$.

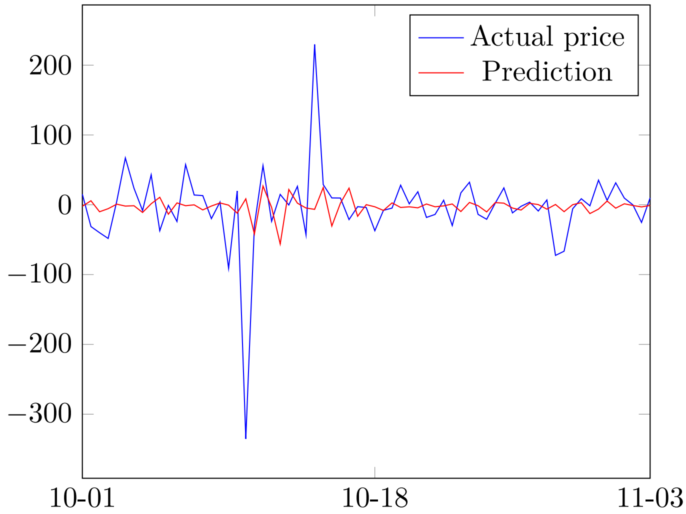
Username	Followers	Tweet text	*comp*
WIRED	10259172	*In 2001, a 25-year-old unemployed college dropout named Bram Cohen created BitTorrent. Now, his latest creation is a digital currency, and startup, called Chia aimed at making cryptocurrency acceptable to the financial industry*	0.65970
business	7613765	*Bitcoin looks like the way forward in a country battered by revolution and corruption*	0.36120
business	7613757	*Bitcoin looks like the way forward in a country battered by revolution and corruption*	0.36120
souljaboy	5404045	*Love bitcoin. Love Robinhood. Too dope*	0.85550
CNBC	4345255	*Bitcoin should be having a good year. Its decline is a warning signal, researcher says. #CNBCCrypto*	0.12800
MarketWatch	4124722	*Bitcoin needs a blessing from the SEC, says analyst*	0.49390
MarketWatch	4124719	*Crytpocurrency enthusiasts aren't going to like how AARP defines bitcoin*	0.59940
MarketWatch	4124717	*Dr. Doom says bitcoin represents the 'mother of all bubbles'*	-0.40190
MarketWatch	4124716	*Dr. Doom" Nouriel Roubini said bitcoin represents the "mother of all bubbles"*	-0.40190
MarketWatch	4124715	*Bitcoin whales aren't responsible for volatility, research firm finds*	0.31820

Table 5.2: Ten tweets posted by users with the most followers on 10 October with username the handle of the user that posted the tweet, followers the number of followers of that user, Tweet text the filtered text of the tweet, and *comp* the compound score for that tweet.

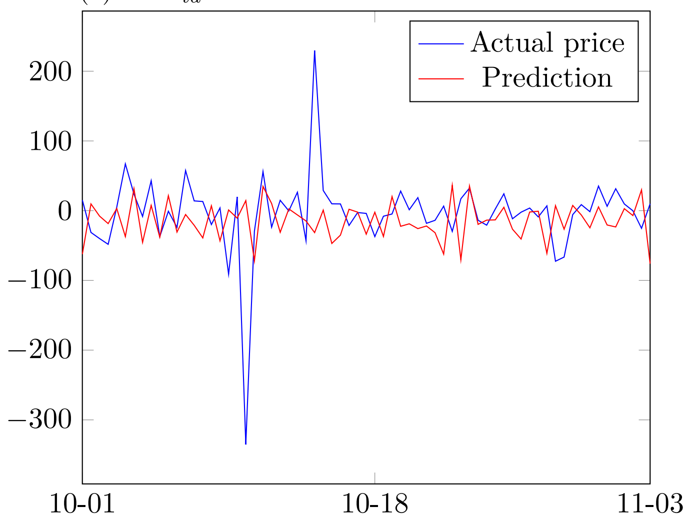
The $Base_{td}$ model's prediction curve in Figure 5.3a matches the changes in price less compared to the sentiment model's prediction curves in Figures 5.3b and 5.3c. This is especially noticeable in the prediction plots starting from 18 October. There, the $Base_{td}$ model's prediction curves for the fourth set have little to no tendency to follow the actual price while $Sent_{td}$ and $Comp_{td}$ have more similarities with the fluctuation of the actual price. However, if other prediction plots of the $Base_{td}$ model that have a slightly worse $RMSE$ are considered, the curve is different. Some prediction plots of the $Base_{td}$ model follow the trend similar to the sentiment models (see Figure C.3a in Appendix C.2). A model having the lowest $RMSE$ among others does not necessarily mean that it has the best prediction plot fit. Some models predict a straight line in order to keep the $RMSE$ as low as possible while different predictions try to follow the shape of the actual price. These prediction can predict extremely off values at some prediction points which will have a negative impact on the MSE while other prediction points are closer to the real value.

5.1.3 Compound or Individual Sentiment Scores

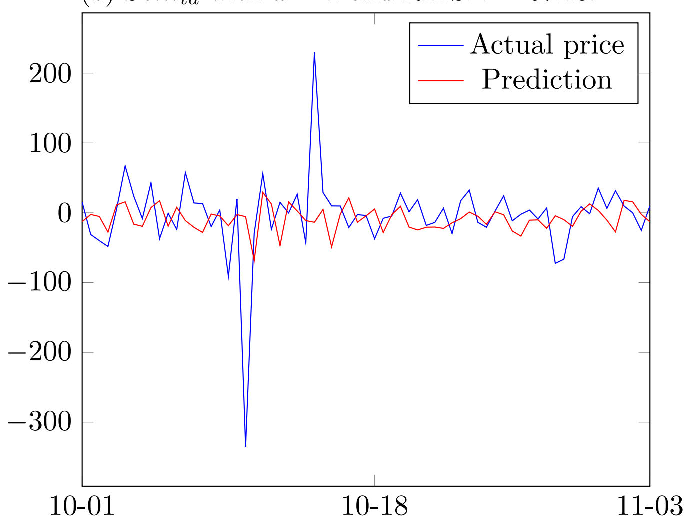
The results of both the $Sent_c$ and $Sent_{td}$ models are unable to show any indication of improvement over the $Comp_c$ and $Comp_{td}$ models in the result metrics and in the prediction plots. *This means that in our experiments, using the individual sentiment scores does not show an improvement over only using the compound score which rejects our initial hypothesis. However, the plots regarding the individual sentiment scores seem to indicate that these scores provide more detailed information regarding the sentiment.* Lastly, we found that the claim that sentiment helps predict bitcoin's price is not as straightforward as it is suggested in [34] and [44].



(a) $Base_{td}$ with $w = 5$ and $RMSE = 59.18$.



(b) $Sent_{td}$ with $w = 2$ and $RMSE = 67.48$.



(c) $Comp_{td}$ with $w = 5$ and $RMSE = 59.28$.

Figure 5.3: Predictions of the $Base_{td}$, $Sent_{td}$, and $Comp_{td}$ models on the test set of fold number 4 with time interval = 12 hours. In blue, the actual price, and in red, the model's price prediction is plotted. w is the window size.

5.2 Impact of Number of Tweets

Recall RQ2: "Is there a difference in using the volume of all tweets that include the term 'bitcoin' and using the volume of the filtered set? ... What is the impact of using the number of positive, negative, and neutral tweets on a model's ability to predict bitcoin's price compared to using the filtered volume as input?" According to [48], the number of positive and negative tweets can help predict bitcoin's price. Furthermore, [55] illustrates that the number of tweets related to bitcoin of the previous days is a "significant diver" when it comes to bitcoin's realized volatility (RV). We expect that using the number of tweets of the filtered set will result in a decrease in $RMSE$ by 5% compared to the prediction model that takes as input the volume of all tweets. We believe that the positive, negative, and neutral volumes will improve the results by 5% compared to using the filtered volume. In this section, we discuss the impact of the volume features on a prediction model's ability to forecast bitcoin's price. There are three ways that a prediction model takes the volume into account: the original tweet volume, the filtered tweet volume, and the individual sentiment volume. These features are discussed in Section 3.1.3. We take a look into the correlation between these tweet volume-based features and bitcoin's price. The predictions will be analyzed on their corresponding plots both for the non-time differenced and time differenced models. In the end, the research question RQ2 is answered.

5.2.1 Is Tweet Volume Correlated with Bitcoin's Price?

Table 5.3 gives the Pearson correlation coefficient, r, and its corresponding p-value, p, between the tweet volume features and bitcoin's close price for three different time intervals. Notice that for larger time intervals, r increases. This may indicate that the models will be able to use the correlation between the volume and bitcoin's price more effectively in higher time intervals. The volume of the original dataset, $volorg$, has a much lower value for r than the volume of the filtered dataset, $volfilt$. Figure 5.4 plots the normalized original tweet volume, the normalized filtered tweet volume, and the normalized close price of bitcoin. The original tweet volume remains stable for the most part compared to the fluctuation of the filtered tweet volume. On the large drop in bitcoin's price on 10 October, both volume metrics have an increase. On 15 October, there is a similar increase in bitcoin's price, this change is not reflected in the original tweet volume but is in the filtered tweet volume. Note that most of the large changes in bitcoin's price can be found in the fluctuation of the filtered tweet volume. The variations in the filtered tweet volume, however, are most of the time not reflected in bitcoin's price. Notice that the volume of the filtered tweets follows a certain pattern, every weekend, the volume decreases significantly, a couple of days later on Tuesday or Wednesday, the volume increases. In Appendix C.1.2, Figure C.2 plots the normalized filtered tweet volume for each month, there the same pattern is visible. Both the rise and fall in bitcoin's price results in an increase in the filtered tweet volume. Figure 5.5 plots the normalized positive sentiment volume, the normalized neutral sentiment volume, and the normalized negative sentiment

	10Min		6h		24h	
Feature	Pearson's r	P-value	Pearson's r	P-value	Pearson's r	P-value
$volorg$	0.0304	0.0001>	0.0936	0.0036	0.1469	0.0228
$volfilt$	0.3446	0.0001>	0.4706	0.0001>	0.5892	0.0
$volpos$	0.3147	0.0001>	0.4296	0.0001>	0.5822	0.0
$volneu$	0.3426	0.0001>	0.5321	0.0001>	0.635	0.0
$volneg$	0.2162	0.0001>	0.3208	0.0001>	0.4034	0.0

Table 5.3: The values for the Pearson correlation coefficient, r, and corresponding p-value, p, with the tweet volume features and bitcoin's close price for three different time intervals. $volorg$ is the number of tweets in the original dataset $Dataset_{org}$. $volfilt$ is the number of tweets in the filtered dataset $Dataset_{filt}$. $volpos$, $volneu$, and $volneg$ the number of positive, neutral, and negative tweets respectively in the filtered dataset $Dataset_{filt}$.

volume. Remember that the sum of these individual sentiment volumes is equal to the filtered tweet volume. Figure 5.5 shows that in general, the positive sentiment volume umbrellas the negative sentiment volume that on his turn umbrellas the neutral sentiment volume. There is no indication of the direction of bitcoin's price changes found in these individual sentiment volumes on both 10 and 15 October. Figure 5.6 shows two different prediction plots of the $VolumeOrg_c$ model where the prediction precedes the actual price in some parts. However, these predictions are far away from being considered accurate in general. There are also different parts where the model predicts the inverse of a shift in price. Note that these models have a much higher $RMSE$ than the best plots of the models of the corresponding data folds.

5.2.2 Predicting the Difference in Price

Figure 5.7 plots four different prediction plots belonging to different time differenced models. Three of them use a tweet volume metric as their input and the other one is a prediction of the $Base_{td}$ model. The $VolumeOrg_{td}$ model does not perform well on the 4th fold set. Comparing $VolumeFilt_{td}$ and $VolumeSent_{td}$ models' prediction plots, $VolumeSent_{td}$ model's prediction has more fluctuation compared to the $VolumeFilt_{td}$ model's prediction that is steadier. The differences, however, remain visually small. It may seem that the $VolumeFilt_{td}$ model and the $VolumeSent_{td}$ model fit the data better compared to the best $Base_{td}$ model, but there are $Base_{td}$ models that have a similar fit (see Figure C.3b in Appendix C.2). As noted before, the model with the lowest $RMSE$ does not necessarily mean that it has a shape that matches the actual price the best.

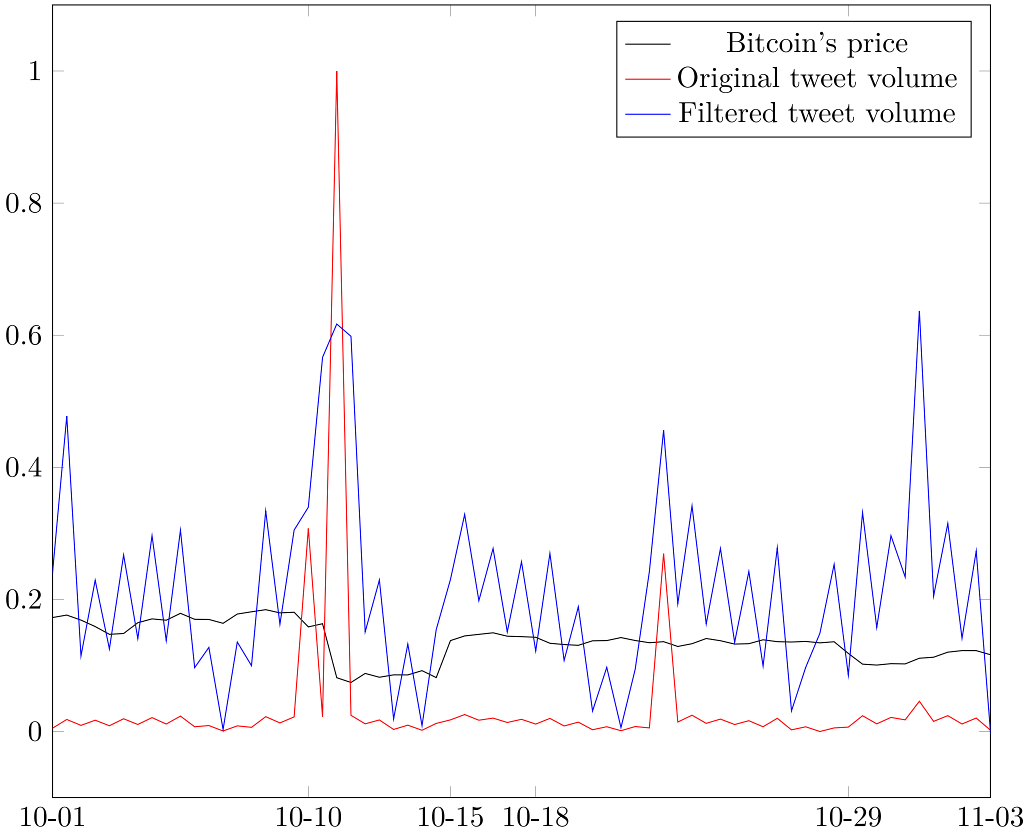


Figure 5.4: The normalized actual price, the normalized original tweet volume and, the normalized filtered tweet volume plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is the test set of the 4th fold.

5.2.3 Is there an Improvement in Performance?

First of all, it is clear that an increase in correlation between bitcoin's price is true when the volume of the filtered set is considered compared to the original volume. To answer the first part of research question RQ2: *The volume of all tweets is less correlated with bitcoin's price compared to the volume of the filtered set. The expected decrease in RMSE by 5% is only true for six out of sixteen different time intervals across both the time differenced and non-time differenced models, thus we reject the first part of our initial hypothesis. In addition, we conclude that it looks beneficial to filter out tweets that are posted by bots and it seems that these tweets cause noise in our data.*

Furthermore, looking at the individual volumes, their proportion to the filtered volume remains stable. There seems to be no indication that one of the three individual volumes behaves differently than the others when the prediction plots are inspected. *The improvement in performance expected in our initial hypothesis did*

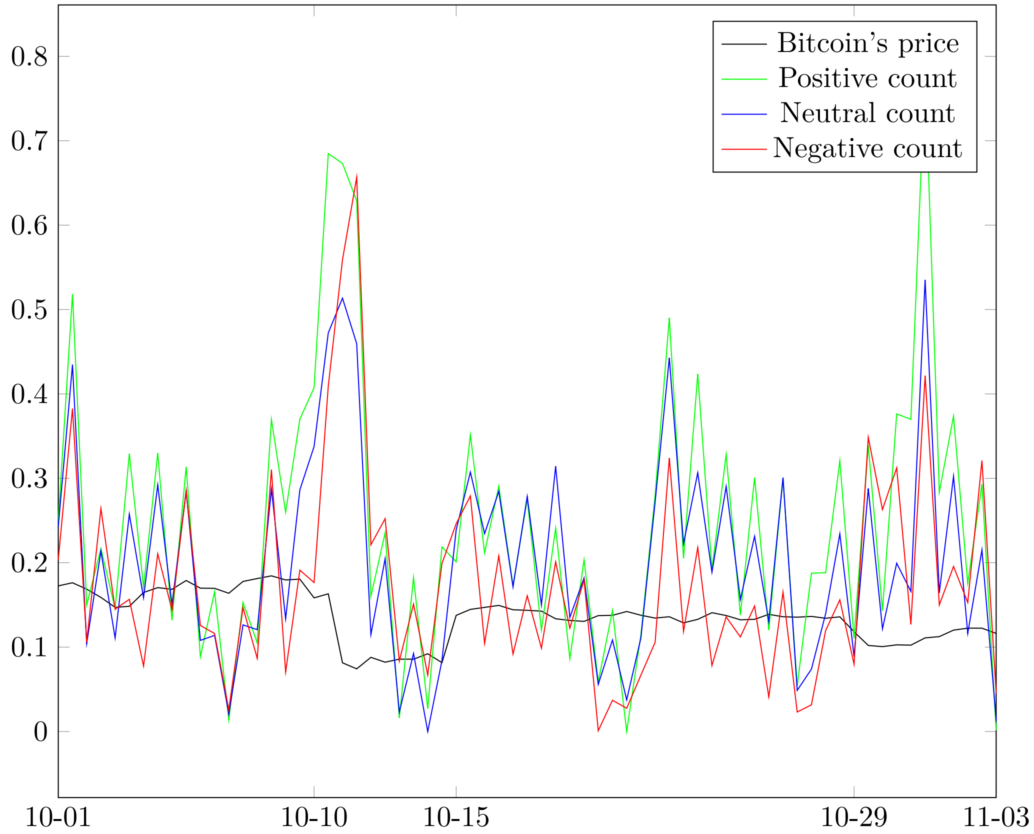
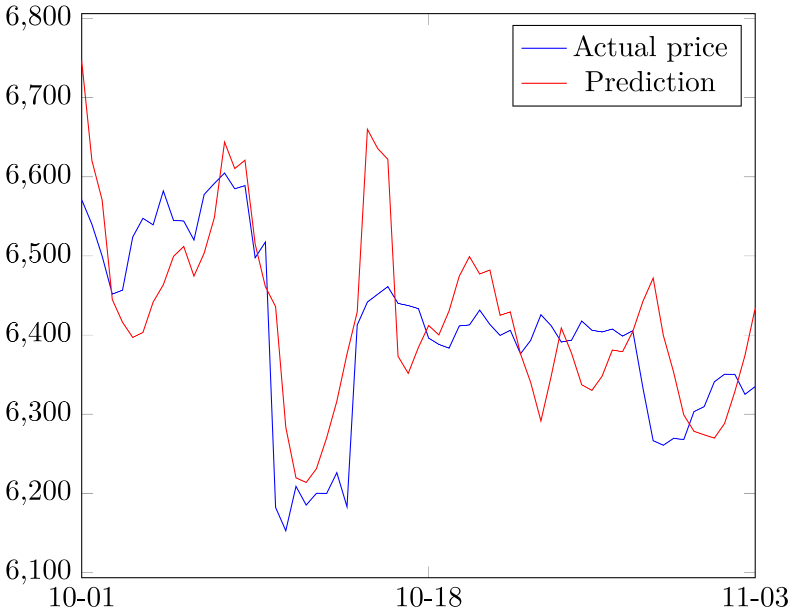
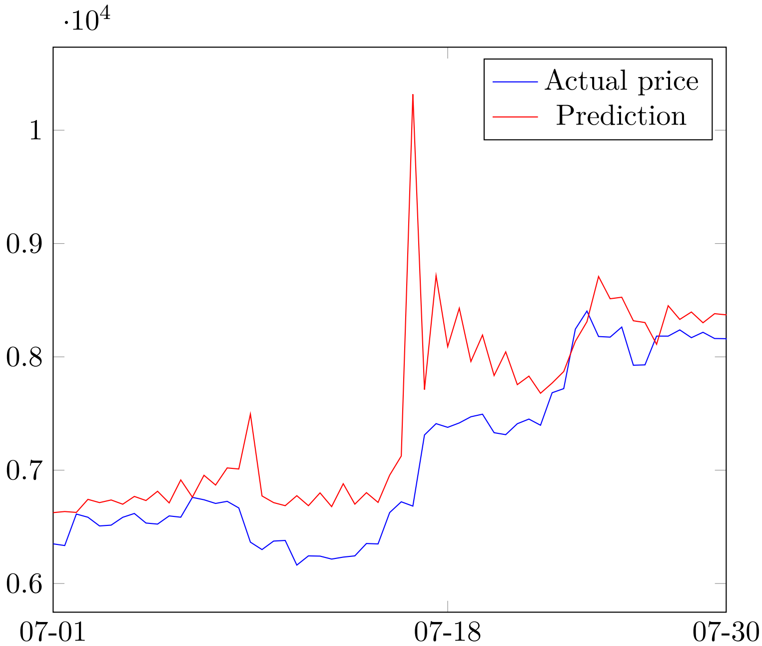


Figure 5.5: The normalized actual price, the normalized positive sentiment volume, the normalized neutral sentiment volume, and the normalized negative sentiment volume plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is the test set of the 4th fold.

not appear in the prediction models. In most cases, the RMSE was worse in the individual sentiment models which is the opposite of what we expected. Hereby, we reject the second part of our initial hypothesis.

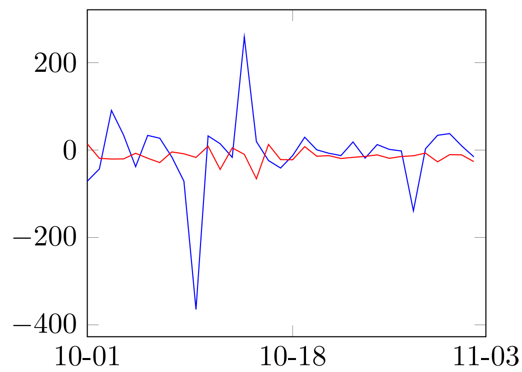


(a) i = 12 hours, w = 50, Fold NR. = 4, $RMSE$ = 129.

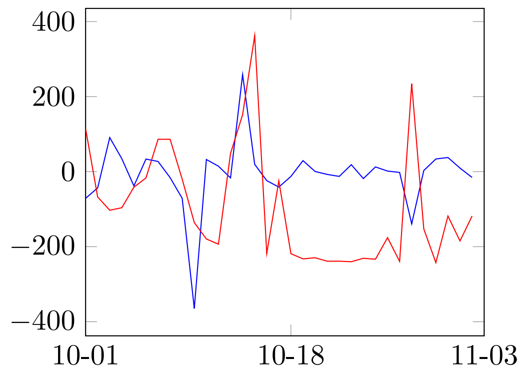


(b) i = 6 hours, w = 2, Fold NR. = 1, $RMSE$ = 640.

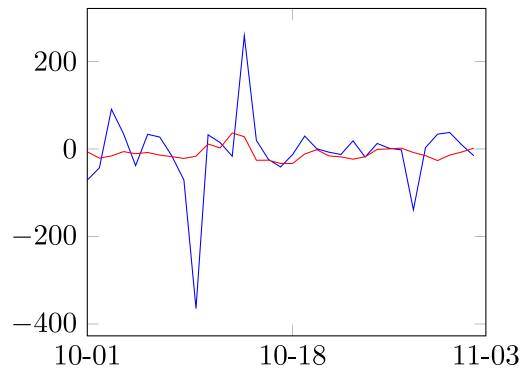
Figure 5.6: Predictions of the $VolumeFilt_c$ model on the test set of different data folds. In blue, the actual price, and in red, $VolumeFilt_c$'s predicted price is plotted. i is the time interval, w is the window size.



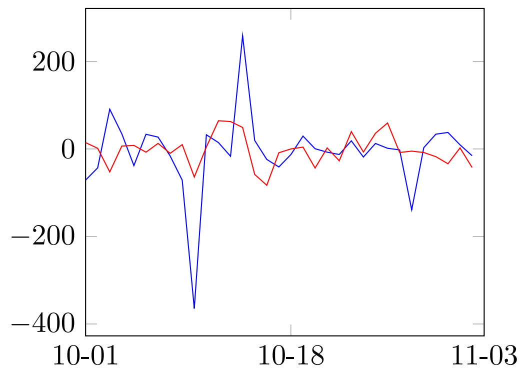
(a) $Base_{td}$ with $w = 5$ and $RMSE = 89.87$.



(b) $VolumeOrg_{td}$ with $w = 100$ and $RMSE = 195.51$.



(c) $VolumeFilt_{td}$ with $w = 10$ and $RMSE = 84.02$.



(d) $VolumeSent_{td}$ with $w = 10$ and $RMSE = 83.48$.

Figure 5.7: Predictions of the $Base_{td}$, $VolumeOrg_{td}$, $VolumeFilt_{td}$, and $VolumeSent_{td}$ models on the test set of fold number 4 with time interval = 24 hours. In blue, the actual price, and in red, the model's price prediction is plotted. w is the window size.

5.3 Representing Sentiment with BERT Vectors

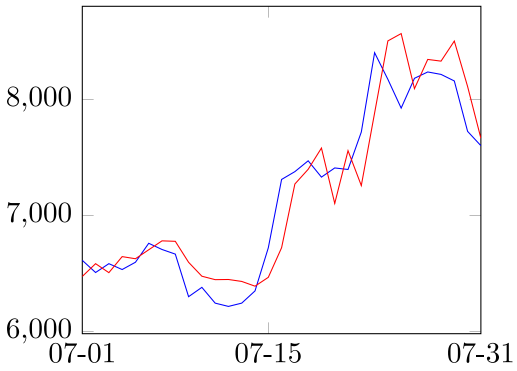
Remember RQ3: "Will using a more sophisticated language representation model that is trained specifically for tweets (BERTTweet) result in an increase in performance when compared to the baseline models and the sentiment models?" BERT itself did not perform well as a tweet sentiment feature for predicting bitcoin's price [23]. Since BERTTweet [46] has never been used to represent tweet sentiment to predict bitcoin's price while BERTTweet is trained using tweets, different results may be possible. We have implemented the first model that uses BERT vectors produced by the BERTTweet model. We expect a reduction in $RMSE$ by 5-10% compared to the baseline models and the VADER sentiment models. In this section, we answer RQ3 by analyzing the performance of the prediction models that use as input these BERT vectors.

5.3.1 The Performance of the Prediction Models

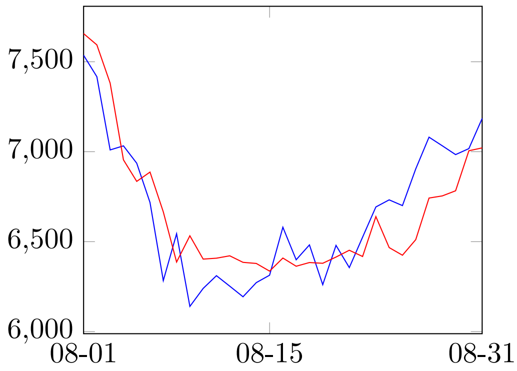
Figure 5.8 plots the best predictions for the $DenseBert_c$ model for each data fold's test set. Notice that as the training size increases (higher fold number means more training data), the prediction becomes less similar to being a lagging copy of the actual price. One possible explanation is that because there is more data available, the model is able to learn a trend different from copying the previous price. Interestingly, the best prediction for the 4th data fold in Figure 5.8d deviates largely from the actual price. Figure 5.9 plots the predictions for both the training set and the validation set. Here can be seen that even if the model fits the training set well, there is no similar fit found for the validation set. The same can be said about the corresponding prediction for the test set in Figure 5.8d. The model is unable to generalize what it has learned in the training set. One possible explanation could be that the trend in the data differs from time to time, meaning what the model has learned in the training data is not relevant in the validation set and in the test set. Looking further into the time differenced Dense-BERT model $DenseBert_{td}$, the $RMSE$ for the best model for each time interval is similar to the $Base_{td}$ model. The best performing $DenseBert_{td}$ models are straight lines. Figure 5.10 illustrates for each individual set the best prediction plot for the 24-hour time interval. These prediction plots are not convincing on the model's ability to predict the change in price using $\overline{BERT}$ as input. The prediction plots, however, fluctuate and show some indication of following the change in price but we can find similar prediction plots for the $Base_{td}$ model (see Figure C.3b in Appendix C.2).

5.3.2 Is Using a BERT Vector Beneficial?

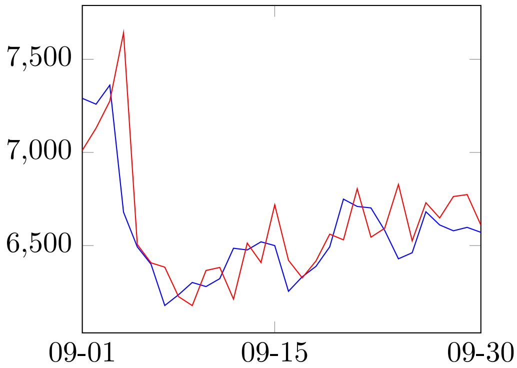
The results of the $DenseBert_c$ and $DenseBert_{td}$ models show no indication of improvement in performance compared to the baseline models $Base_c$ and $Base_{td}$. *Our initial hypothesis that expected an improvement in performance is hereby rejected. Moreover, using the BERT vector as input and compressing it with a dense layer seems to introduce noise into the data and even decreases the model's performance.*



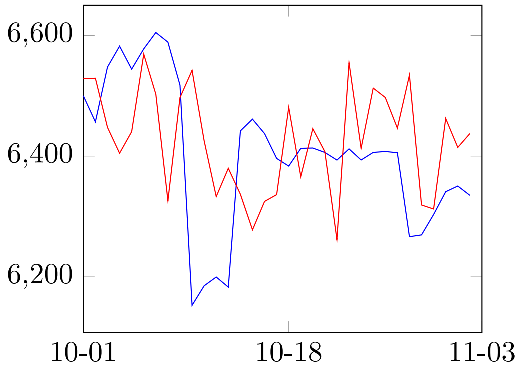
(a) $w = 5$ and fold NR. 1.



(b) $w = 100$ and fold NR. 2.

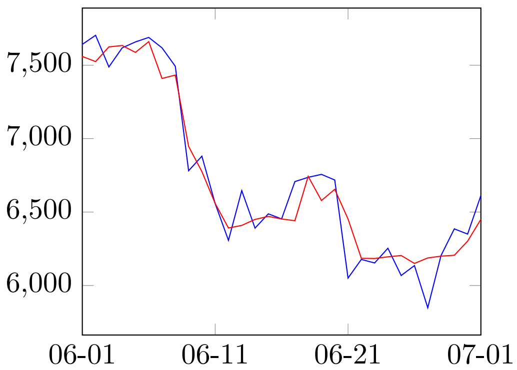


(c) $w = 10$ and fold NR. 3.

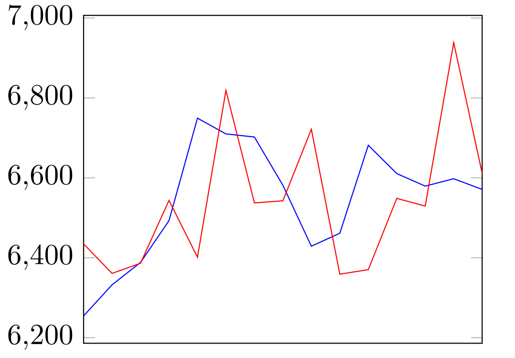


(d) $w = 5$ and fold NR. 4.

Figure 5.8: Best predictions (lowest RMSE) plots of the $DenseBert_c$ model on the test set for fold number with time interval = 24 hours. In blue, the actual price, and in red, the model's price prediction is plotted. w is the window size, fold NR. is the data fold number.



(a) Part of the training set.



(b) Validation set

Figure 5.9: Prediction plot of the $DenseBert_c$ model for a part of the training and the whole validation set with time interval = 24 hours, window size = 4, Fold NR. 4.

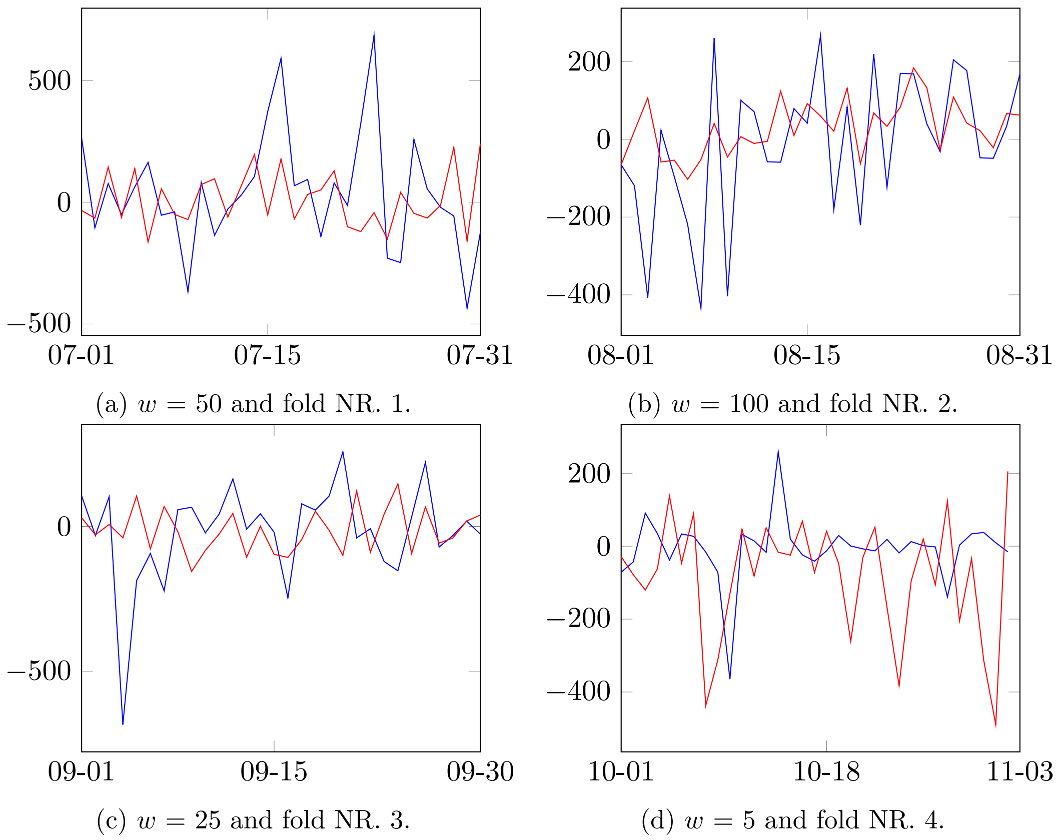


Figure 5.10: Best predictions plots (lowest RMSE) of the $DenseBert_{td}$ model on the test set of different folds. The time interval is equal to 24 hours. In blue the actual price is plotted and in red, the model's price prediction is plotted. w is the window size, fold NR. is the data fold number.

5.4 Other Remarks

5.4.1 Impact of Different Time Intervals

For shorter time intervals ranging from 10 minutes up to 3 hours, the prediction plots are copies of the previous close price or are straight lines for the time differenced versions. Whenever the time interval is equal to or larger than 6 hours, the prediction becomes interesting. For the models where time differencing is not applied, the prediction plot starts to deviate from being a lagged copy of the previous close price. The same can be said about the time differenced models, there, the prediction stops being a straight line. This also occurs for some plots where the time interval is equal to 3 hours. The Pearson correlation coefficient, r, for all features in Tables 5.1 and 5.3 increases with larger time intervals. We think that this is because there is probably more information found in features with larger time intervals that can be correlated to bitcoin's price. For example, the top tweets in a 10 minute period are more likely to be less interesting than the top tweets of a whole day.

5.4.2 The Trade-Off between Correlation and Training Size

For most of the results, the experiments where the 4th data fold is used resulted in the best results. This fold has the largest training data. Looking at all the results in Appendix B, smaller training data generally results in a lower $RMSE$. This introduces a trade-off: increasing the time interval, increases the correlation of features between bitcoin's price, but it simultaneously decreases the size of the training data which makes training the model harder since there are fewer data points available to learn from.

Chapter 6

Conclusion & Future Work

The final chapter contains the overall conclusion and suggestions for future work.

6.1 Conclusion

Several Twitter features are analyzed in depth by using them in prediction models. We distinguish between the tweet sentiment features, tweet volume features, and BERT features. These features are used alongside bitcoin's historical price as input in our LSTM-base models.

We introduce a time differenced prediction task where the model predicts the difference in price using the history of the previous differences in price. This eliminates the model's possibility to use the previous price as its current prediction to keep the error low. Every model has a non-time differenced and a time differenced version. We establish two different criteria that the prediction models have to meet to show that they can truly predict bitcoin's price. A prediction model shows improvement over the baseline models:

- if the gap in the non-time difference model decreases compared to the baseline models and
- if the $RMSE$ in the time differenced model is lower compared to the baseline models.

Ideally, both criteria are met for a model to prove better performance. Inspecting the prediction plots of the time differenced models may reveal interesting results.

By adding tweet sentiment features, the models were not able to improve their prediction compared to the baseline. The prediction plots reveal that these features introduce noise on top of the baseline model. Analyzing the correlation between the sentiment features and bitcoin's price, we conclude that tweet sentiment is lagging behind the change in bitcoin's price in our dataset. This suggests that the tweets are merely a report of the change in bitcoin's price after it already happened. Large changes in bitcoin's price are represented in the fluctuations of the compound score, but not every change in the compound score is reflected in bitcoin's price. It appears to be that big changes in bitcoin's price causes alterations in the sentiment of the tweets. Furthermore, the positive sentiment score reacts proportionally to a rise in bitcoin's price and the negative sentiment score reacts inversely proportional. The neutral sentiment score is the highest among them for periods where bitcoin's price remains stable. These individual sentiment scores seem to provide more detailed information regarding the sentiment that the prediction models could benefit from.

Second, we used BERT vectors obtained by the BERTTWeet model's last hidden layer in our prediction tasks. This is the first time that these vectors were used in a bitcoin prediction model. The weighted average of the BERT vector and bitcoin's historical price is passed to our Dense-LSTM model which we train to forecast bitcoin's price. Our results suggest that using these BERT vectors as an input to the prediction model seems to introduce noise into the data and even decreases the model's performance.

Third, using the volume of a filtered set of tweets increases the correlation between bitcoin's price compared to the original tweet volume. This indicates that filtering tweets presumably reduces noise in our data. Major shifts in bitcoin's price can be found in the changes in the volume of the filtered tweets. An increase in the filtered tweet volume, however, does not indicate the direction and size of the corresponding change in bitcoin's price in our dataset. On top of that, not all changes in volume are reflected in bitcoin's price. Both a large increase and decrease in bitcoin's price seem to increase the volume of tweets regarding bitcoin on Twitter. This indicates that large fluctuations in bitcoin's price cause spikes in the number of tweets that are posted regarding bitcoin. Interestingly, the proportions of the number of positive, neutral, and negative tweets remain unchanged during large fluctuations in bitcoin's price.

Lastly, we introduce a trade-off: increasing the time interval of the features increases the correlation of these features between bitcoin's price, but it simultaneously decreases the size of the training data which makes training the model harder since there are fewer data points available to learn from.

6.2 Future Work

We feel like our BERTTweet input feature still has room for further improvement. Several other models can be introduced using this vector such as a CNN-LSTM. Furthermore, we would like to refine the BERTTweet model on the sentiment classification task by using our data. We could calculate the ground truth sentiment via the VADER model and refine BERTTweet to predict these. This way, the feature vector could be more meaningful for our prediction models.

A combination of different features is also interesting to analyze. Each feature could have different window sizes or we can even introduce a (weighted) moving average for different features. More sentiment data from different platforms can have different impacts on our prediction models. Some interesting sources can be news articles, Telegram group messages, Reddit activity, and Google searches.

We have seen different patterns in tweet activity in our data such as the weekly incline and decline of the volume of tweets. Normalizing these patterns could get rid of noise in our data and even reveal other trends that could be useful for our prediction models.

We have seen that reducing the number of tweets can remove noise that is not useful for the prediction models. Further research can be done on ways to reduce the number of tweets even more to obtain a higher correlation. We could make a pool of influential users (as suggested in [48]) and combine them with accounts that have a minimum number of followers. Another possibility is to only take into account the sentiment of the top N tweets posted by the accounts with the highest followers within a period. We can also look into the tweets with the most likes or retweets additionally to the followers.

Furthermore, it could be interesting to train the models to predict bitcoin's price in more than one step. This is interesting for the shorter time intervals to forecast sudden large drops or rises in bitcoin's price while they are happening. For example, if the price drops 50% within one hour, a model with a 15-minute interval can predict the drop while it is happening compared to a model that predicts the next price for the next 6 hours. The model could learn to predict large changes that happen in a short time from previous occurrences.

We have seen that even if models have a high error, they may follow the shape of the ground truth more favorably in some parts of the curve compared to models with a lower error. Also, some predictions that have a better score appear to be straight lines. It could be interesting to alter the loss function to penalize this undesired behavior. We can add a second loss function that only checks if the direction of the prediction is correct to favor prediction curves that match the shape of the ground truth.

6.3 Code and Data

For reproducibility purposes and to encourage others to build further our findings, we provided our code and data in [16].

Appendices

Appendix A

The First Appendix

This appendix gives more detailed information regarding different experiment configurations.

A.1 Dataset Configuration

In this section, more information about the datasets can be found. Selecting different time intervals will make the size of the training, validation, and test set vary. For example, a 30-day long set can be split into 30 entries of one-day intervals, 60 entries of 12-hour intervals, 90 entries of 6-hour intervals, 180 entries of 3-hour intervals, 270 entries of 2-hour intervals, 540 entries of 1-hour intervals, 1080 entries of 30-minute intervals and 3240 entries of 10-minute intervals. The number of entries in a set is inversely proportional to the length of the time interval. Similarly different sets have different sizes as discussed in Section 2.4. Notice that only the size of the training set is affected by the fold number, the validation set size is always 15 days long and the test set size is always month-long. The fluctuations in test size are because different months have different lengths.
An increase in the window size will decrease the training set size. This is due to time lagging features. If the window size equals 10, the first 10 rows cannot be used since their 10 previous values are not available and are therefore dropped. An increase in the size of the window will account for a decrease in the training set size. For example, a training set with a window size of 10 that has 374 items will have 359 items when the window size is set to 25. This is 15 items less which is the difference between the two window sizes. Only the first part of the set is affected by the size of the window since the removal is only done at the head of the dataset. In Table A.8, there is no entry for the first fold with a window size of 100. This is not possible since the size of the first set is not large enough to time lag the features with a window size of 100.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	14399	2160	4464
2	2	18863	2160	4464
2	3	23327	2160	4320
2	4	27647	2160	4895
5	1	14396	2160	4464
5	2	18860	2160	4464
5	3	23324	2160	4320
5	4	27644	2160	4895
10	1	14391	2160	4464
10	2	18855	2160	4464
10	3	23319	2160	4320
10	4	27639	2160	4895
25	1	14376	2160	4464
25	2	18840	2160	4464
25	3	23304	2160	4320
25	4	27624	2160	4895
50	1	14351	2160	4464
50	2	18815	2160	4464
50	3	23279	2160	4320
50	4	27599	2160	4895
100	1	14301	2160	4464
100	2	18765	2160	4464
100	3	23229	2160	4320
100	4	27549	2160	4895

Table A.1: The size for the training, validation, and test for a time interval of 10 minutes for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	4799	720	1488
2	2	6287	720	1488
2	3	7775	720	1440
2	4	9215	720	1631
5	1	4796	720	1488
5	2	6284	720	1488
5	3	7772	720	1440
5	4	9212	720	1631
10	1	4791	720	1488
10	2	6279	720	1488
10	3	7767	720	1440
10	4	9207	720	1631
25	1	4776	720	1488
25	2	6264	720	1488
25	3	7752	720	1440
25	4	9192	720	1631
50	1	4751	720	1488
50	2	6239	720	1488
50	3	7727	720	1440
50	4	9167	720	1631
100	1	4701	720	1488
100	2	6189	720	1488
100	3	7677	720	1440
100	4	9117	720	1631

Table A.2: The size for the training, validation, and test for a time interval of 30 minutes for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	2399	360	744
2	2	3143	360	744
2	3	3887	360	720
2	4	4607	360	815
5	1	2396	360	744
5	2	3140	360	744
5	3	3884	360	720
5	4	4604	360	815
10	1	2391	360	744
10	2	3135	360	744
10	3	3879	360	720
10	4	4599	360	815
25	1	2376	360	744
25	2	3120	360	744
25	3	3864	360	720
25	4	4584	360	815
50	1	2351	360	744
50	2	3095	360	744
50	3	3839	360	720
50	4	4559	360	815
100	1	2301	360	744
100	2	3045	360	744
100	3	3789	360	720
100	4	4509	360	815

Table A.3: The size for the training, validation, and test for a time interval of 1 hour for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	1199	180	372
2	2	1571	180	372
2	3	1943	180	360
2	4	2303	180	407
5	1	1196	180	372
5	2	1568	180	372
5	3	1940	180	360
5	4	2300	180	407
10	1	1191	180	372
10	2	1563	180	372
10	3	1935	180	360
10	4	2295	180	407
25	1	1176	180	372
25	2	1548	180	372
25	3	1920	180	360
25	4	2280	180	407
50	1	1151	180	372
50	2	1523	180	372
50	3	1895	180	360
50	4	2255	180	407
100	1	1101	180	372
100	2	1473	180	372
100	3	1845	180	360
100	4	2205	180	407

Table A.4: The size for the training, validation, and test for a time interval of 2 hours for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	799	120	248
2	2	1047	120	248
2	3	1295	120	240
2	4	1535	120	271
5	1	796	120	248
5	2	1044	120	248
5	3	1292	120	240
5	4	1532	120	271
10	1	791	120	248
10	2	1039	120	248
10	3	1287	120	240
10	4	1527	120	271
25	1	776	120	248
25	2	1024	120	248
25	3	1272	120	240
25	4	1512	120	271
50	1	751	120	248
50	2	999	120	248
50	3	1247	120	240
50	4	1487	120	271
100	1	701	120	248
100	2	949	120	248
100	3	1197	120	240
100	4	1437	120	271

Table A.5: The size for the training, validation, and test for a time interval of 3 hours for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	399	60	124
2	2	523	60	124
2	3	647	60	120
2	4	767	60	135
5	1	396	60	124
5	2	520	60	124
5	3	644	60	120
5	4	764	60	135
10	1	391	60	124
10	2	515	60	124
10	3	639	60	120
10	4	759	60	135
25	1	376	60	124
25	2	500	60	124
25	3	624	60	120
25	4	744	60	135
50	1	351	60	124
50	2	475	60	124
50	3	599	60	120
50	4	719	60	135
100	1	301	60	124
100	2	425	60	124
100	3	549	60	120
100	4	669	60	135

Table A.6: The size for the training, validation, and test for a time interval of 6 hours for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	199	30	62
2	2	261	30	62
2	3	323	30	60
2	4	383	30	67
5	1	196	30	62
5	2	258	30	62
5	3	320	30	60
5	4	380	30	67
10	1	191	30	62
10	2	253	30	62
10	3	315	30	60
10	4	375	30	67
25	1	176	30	62
25	2	238	30	62
25	3	300	30	60
25	4	360	30	67
50	1	151	30	62
50	2	213	30	62
50	3	275	30	60
50	4	335	30	67
100	1	101	30	62
100	2	163	30	62
100	3	225	30	60
100	4	285	30	67

Table A.7: The size for the training, validation, and test for a time interval of 12 hours for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	Training Set Size	Validation Set Size	Test Set Size
2	1	99	15	31
2	2	130	15	31
2	3	161	15	30
2	4	191	15	33
5	1	96	15	31
5	2	127	15	31
5	3	158	15	30
5	4	188	15	33
10	1	91	15	31
10	2	122	15	31
10	3	153	15	30
10	4	183	15	33
25	1	76	15	31
25	2	107	15	31
25	3	138	15	30
25	4	168	15	33
50	1	51	15	31
50	2	82	15	31
50	3	113	15	30
50	4	143	15	33
100	1	1	15	31
100	2	32	15	31
100	3	63	15	30
100	4	93	15	33

Table A.8: The size for the training, validation, and test for a time interval of 24 hours for varying window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

Appendix B

The Second Appendix

In this appendix, extensive results of the experiments are found.

B.1 Baseline Models Results

B.1.1 The Non-Time Differenced Baseline Model $Base_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	91.54	13413.04	115.81	0.9746	68
2	2	158.49	29489.8	171.73	0.819	127
2	3	62.06	4751.73	68.93	0.9513	75
2	4	39.41	1895.07	43.53	0.8577	93
5	1	179.84	55078.42	234.69	0.8958	77
5	2	132.08	19698.36	140.35	0.8791	88
5	3	91.78	11349.08	106.53	0.8836	67
5	4	17.37	465.99	21.59	0.965	69
10	1	146.07	35526.71	188.49	0.9328	69
10	2	125.65	19530.85	139.75	0.8801	76
10	3	71.26	6074.32	77.94	0.9377	74
10	4	35.09	1524.72	39.05	0.8855	72
25	1	136.08	25896.88	160.93	0.951	82
25	2	130.93	20801.72	144.23	0.8723	74
25	3	78.01	10439.18	102.17	0.893	68
25	4	42.42	2077.63	45.58	0.844	92
50	1	198.78	46015.14	214.51	0.913	67
50	2	122.23	19222.5	138.65	0.882	70
50	3	81.58	15728.94	125.42	0.8387	83
50	4	20.88	736.6	27.14	0.9447	74
100	1	244.1	77904.22	279.11	0.8527	81
100	2	157.56	32118.49	179.22	0.8028	102
100	3	94.61	15095.81	122.87	0.8452	80
100	4	39.4	2496.63	49.97	0.8125	87

Table B.1: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	84.08	10865.35	104.24	0.9795	82
2	2	29.1	2125.12	46.1	0.9869	66
2	3	25.71	1321.83	36.36	0.9864	83
2	4	14.96	545.59	23.36	0.9589	74
5	1	57.2	4981.88	70.58	0.9906	79
5	2	28.48	1853.35	43.05	0.9886	67
5	3	31.86	1731.58	41.61	0.9822	72
5	4	12.01	447.49	21.15	0.9663	90
10	1	46.39	4636.73	68.09	0.9912	85
10	2	32.49	2279.06	47.74	0.986	74
10	3	21.72	1266.12	35.58	0.987	66
10	4	9.1	392.72	19.82	0.9704	95
25	1	64.19	7680.73	87.64	0.9855	71
25	2	46.96	3942.57	62.79	0.9757	75
25	3	23.04	1441.08	37.96	0.9852	84
25	4	11.21	445.37	21.1	0.9665	69
50	1	65.37	7576.01	87.04	0.9857	76
50	2	48.15	4352.86	65.98	0.9732	79
50	3	20.25	1337.17	36.57	0.9863	96
50	4	14.69	543.71	23.32	0.9591	68
100	1	66.35	6898.35	83.06	0.987	89
100	2	67.77	7108.23	84.31	0.9562	88
100	3	29.4	2103.43	45.86	0.9784	83
100	4	18.48	803.57	28.35	0.9395	102

Table B.2: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	81.34	9838.26	99.19	0.9814	73
2	2	32.03	2374.72	48.73	0.9853	74
2	3	22.34	1355.6	36.82	0.986	77
2	4	14.76	655.62	25.61	0.9504	67
5	1	52.14	5367.7	73.26	0.9898	70
5	2	40.38	3084.77	55.54	0.9809	73
5	3	25.26	1704.77	41.29	0.9824	84
5	4	32.83	1618.76	40.23	0.8776	69
10	1	86.16	12590.36	112.21	0.9762	84
10	2	52.06	4401.5	66.34	0.9727	86
10	3	24.04	1672.13	40.89	0.9828	95
10	4	11.03	587.99	24.25	0.9555	96
25	1	46.51	4442.98	66.66	0.9916	80
25	2	54.46	4685.4	68.45	0.9709	69
25	3	25.42	1822.47	42.69	0.9812	66
25	4	12.86	610.55	24.71	0.9538	72
50	1	54.65	5382.87	73.37	0.9898	73
50	2	74.61	7544.35	86.86	0.9532	77
50	3	28.04	2145.05	46.31	0.9779	89
50	4	12.2	630.7	25.11	0.9523	67
100	1	172.18	92623.14	304.34	0.8248	78
100	2	66.11	6658.1	81.6	0.9587	70
100	3	43.27	4178.49	64.64	0.9569	67
100	4	14.04	791.32	28.13	0.9402	92

Table B.3: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	84.79	13202.78	114.9	0.975	66
2	2	43.84	4459.62	66.78	0.9721	70
2	3	30.53	2379.4	48.78	0.9752	94
2	4	14.1	610.75	24.71	0.9535	77
5	1	85.89	13419.64	115.84	0.9746	66
5	2	49.37	5419.88	73.62	0.9661	69
5	3	32.97	2619.63	51.18	0.9727	114
5	4	15.9	767.87	27.71	0.9415	82
10	1	69.05	8608.31	92.78	0.9837	69
10	2	48.89	5245.34	72.42	0.9672	71
10	3	32.78	2615.96	51.15	0.9727	96
10	4	13.72	685.22	26.18	0.9478	78
25	1	55.62	6217.6	78.85	0.9882	72
25	2	51.66	5452.63	73.84	0.9659	87
25	3	34.02	2819.99	53.1	0.9706	71
25	4	14.13	699.17	26.44	0.9467	66
50	1	95.04	15478.75	124.41	0.9707	70
50	2	61.84	7205.98	84.89	0.9549	68
50	3	35.52	2980.7	54.6	0.9689	86
50	4	14.45	712.99	26.7	0.9457	115
100	1	609.47	1385375.2	1177.02	-1.6243	69
100	2	64.24	8239.3	90.77	0.9485	69
100	3	51.76	4533.33	67.33	0.9527	88
100	4	19.45	1066.81	32.66	0.9187	94

Table B.4: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	87.88	13910.35	117.94	0.9734	77
2	2	57.21	6981.03	83.55	0.9562	77
2	3	36.76	3302.32	57.47	0.966	91
2	4	16.24	920.04	30.33	0.9297	80
5	1	71.49	9619.44	98.08	0.9816	68
5	2	56.49	6941.44	83.32	0.9564	66
5	3	38.46	3952.27	62.87	0.9593	72
5	4	17.25	995.59	31.55	0.924	94
10	1	74.31	10059.55	100.3	0.9808	66
10	2	58.43	7261.14	85.21	0.9544	81
10	3	37.35	3753.89	61.27	0.9613	74
10	4	15.65	956.43	30.93	0.927	96
25	1	73.56	10070.6	100.35	0.9808	80
25	2	60.61	7675.03	87.61	0.9518	76
25	3	39.84	4126.25	64.24	0.9575	68
25	4	16.67	1009.17	31.77	0.9229	74
50	1	109.46	19352.91	139.11	0.963	100
50	2	63.52	7697.02	87.73	0.9517	73
50	3	50.93	4952.3	70.37	0.949	76
50	4	29.26	1573.36	39.67	0.8798	70
100	1	592.8	1038493.4	1019.06	-0.9835	66
100	2	76.81	10100.17	100.5	0.9366	73
100	3	49.18	5369.05	73.27	0.9447	82
100	4	23.38	1452.37	38.11	0.8891	87

Table B.5: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	114.54	21188.99	145.56	0.9597	79
2	2	79.58	12905.98	113.6	0.9177	71
2	3	51.13	6702.37	81.87	0.9291	91
2	4	23.52	1720.64	41.48	0.8665	78
5	1	110.9	20772.46	144.13	0.9605	77
5	2	83.75	13409.02	115.8	0.9145	66
5	3	52.01	6431.77	80.2	0.9319	72
5	4	23.06	1783.08	42.23	0.8616	67
10	1	87.03	14051.29	118.54	0.9733	79
10	2	83.18	13538.9	116.36	0.9137	69
10	3	53.52	6862.29	82.84	0.9274	73
10	4	21.99	1691.42	41.13	0.8687	86
25	1	113.53	19640.53	140.14	0.9626	66
25	2	87.46	14245.01	119.35	0.9092	68
25	3	57.14	7260.31	85.21	0.9231	74
25	4	22.79	1785.03	42.25	0.8615	74
50	1	157.66	33553.1	183.18	0.9361	84
50	2	99.31	16077.28	126.8	0.8975	82
50	3	63.12	8212.67	90.62	0.9131	92
50	4	31.42	2189.03	46.79	0.8301	72
100	1	316.71	123172.63	350.96	0.7655	76
100	2	103.94	18497.54	136.01	0.8821	73
100	3	84.68	14048.64	118.53	0.8513	75
100	4	38.84	3146.19	56.09	0.7558	77

Table B.6: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	138.86	34788.8	186.52	0.9339	66
2	2	104.94	18896.26	137.46	0.875	86
2	3	79.55	12426.86	111.48	0.8577	73
2	4	30.34	3061.56	55.33	0.7695	68
5	1	129.29	30606.44	174.95	0.9419	90
5	2	105.95	19304.87	138.94	0.8723	68
5	3	74.25	12660.96	112.52	0.855	81
5	4	32.16	3160.85	56.22	0.762	82
10	1	129.43	29125.24	170.66	0.9447	68
10	2	116.68	25315.05	159.11	0.8325	68
10	3	75.89	12570.44	112.12	0.856	102
10	4	32.94	3280.62	57.28	0.753	96
25	1	152.66	33833.48	183.94	0.9357	66
25	2	136.51	29414.25	171.51	0.8054	68
25	3	82.66	14623.81	120.93	0.8325	74
25	4	39.08	3673.49	60.61	0.7234	98
50	1	297.16	119482.11	345.66	0.7731	75
50	2	142.21	32758.55	180.99	0.7833	81
50	3	88.35	14814.64	121.72	0.8303	66
50	4	45.53	4231.54	65.05	0.6814	71
100	1	714.17	992493.6	996.24	-0.8852	93
100	2	328.73	146196.3	382.36	0.0328	66
100	3	159.71	42784.2	206.84	0.51	91
100	4	76.19	8896.59	94.32	0.3301	66

Table B.7: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	192.35	60397.34	245.76	0.8813	66
2	2	139.29	31954.42	178.76	0.7728	72
2	3	105.32	26286.03	162.13	0.6822	66
2	4	48.13	6999.32	83.66	0.507	69
5	1	199.76	62911.78	250.82	0.8764	77
5	2	184.71	56649.94	238.01	0.5972	73
5	3	113.37	29835.13	172.73	0.6393	70
5	4	59.25	8621.09	92.85	0.3927	69
10	1	200.15	60572.29	246.11	0.881	72
10	2	323.6	153939.56	392.35	-0.0945	89
10	3	106.53	28690.6	169.38	0.6531	67
10	4	58.41	8667.09	93.1	0.3895	78
25	1	258.1	98811.79	314.34	0.8058	67
25	2	318.26	155104.94	393.83	-0.1028	77
25	3	119.07	31455.09	177.36	0.6197	84
25	4	62.58	8014.93	89.53	0.4354	88
50	1	204.15	84380.69	290.48	0.8342	75
50	2	347.53	177061.95	420.79	-0.2589	82
50	3	194.36	49400.46	222.26	0.4027	86
50	4	67.35	7989.15	89.38	0.4372	82
100	1	751.86	963423.1	981.54	-0.8934	65
100	2	506.06	344774.9	587.18	-1.4513	69
100	3	189.03	81809.64	286.02	0.0109	114
100	4	120.49	21979.2	148.25	-0.5483	71

Table B.8: Result metrics for the $Base_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.1.2 The Time Differenced Baseline Model $Base_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.81	17.26	-0.0004	202
2	2	11.83	363.38	19.06	-0.0018	164
2	3	8.45	215.21	14.67	-0.0018	100
2	4	4.6	91.92	9.59	-0.0043	102
5	1	10.48	297.75	17.26	-0.0002	318
5	2	11.83	363.36	19.06	-0.0017	195
5	3	8.45	214.97	14.66	-0.0007	192
5	4	4.58	91.74	9.58	-0.0023	66
10	1	10.48	297.8	17.26	-0.0003	236
10	2	11.83	363.53	19.07	-0.0022	152
10	3	8.45	215.05	14.66	-0.0011	93
10	4	4.56	91.59	9.57	-0.0007	98
25	1	10.48	297.71	17.25	-0.0	370
25	2	11.84	363.71	19.07	-0.0027	158
25	3	8.45	215.02	14.66	-0.001	191
25	4	4.58	91.77	9.58	-0.0027	73
50	1	10.51	298.25	17.27	-0.0019	76
50	2	11.83	363.19	19.06	-0.0012	247
50	3	8.45	215.07	14.67	-0.0012	171
50	4	4.58	91.78	9.58	-0.0027	140
100	1	10.49	298.05	17.26	-0.0012	110
100	2	11.98	368.98	19.21	-0.0172	119
100	3	8.45	214.92	14.66	-0.0005	74
100	4	4.55	91.55	9.57	-0.0002	66

Table B.9: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.52	908.18	30.14	-0.0034	86
2	2	19.83	1010.88	31.79	-0.0009	66
2	3	14.78	742.97	27.26	-0.0065	69
2	4	7.4	263.41	16.23	-0.0005	66
5	1	17.48	906.45	30.11	-0.0015	78
5	2	19.84	1010.92	31.79	-0.001	78
5	3	14.82	744.17	27.28	-0.0081	69
5	4	7.48	264.27	16.26	-0.0037	93
10	1	17.48	906.43	30.11	-0.0015	86
10	2	19.84	1011.36	31.8	-0.0014	67
10	3	14.84	744.8	27.29	-0.009	68
10	4	7.51	264.64	16.27	-0.0051	68
25	1	17.5	907.4	30.12	-0.0026	71
25	2	19.84	1011.14	31.8	-0.0012	75
25	3	14.8	743.6	27.27	-0.0074	77
25	4	7.53	264.88	16.28	-0.006	70
50	1	17.47	906.07	30.1	-0.0011	83
50	2	19.83	1010.84	31.79	-0.0009	85
50	3	14.85	745.2	27.3	-0.0095	74
50	4	7.58	265.59	16.3	-0.0087	78
100	1	17.49	906.97	30.12	-0.0021	72
100	2	19.84	1011.41	31.8	-0.0015	86
100	3	15.05	751.32	27.41	-0.0178	83
100	4	7.42	263.65	16.24	-0.0014	81

Table B.10: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.08	1820.51	42.67	-0.006	79
2	2	28.63	2064.21	45.43	-0.0001	74
2	3	20.08	1176.15	34.3	-0.0058	85
2	4	9.87	466.43	21.6	-0.0	76
5	1	25.05	1818.4	42.64	-0.0048	98
5	2	28.63	2064.15	45.43	-0.0001	67
5	3	20.05	1173.76	34.26	-0.0037	69
5	4	9.87	466.42	21.6	-0.0	66
10	1	25.09	1821.1	42.67	-0.0063	76
10	2	28.66	2064.83	45.44	-0.0004	110
10	3	20.04	1172.98	34.25	-0.0031	68
10	4	9.88	466.49	21.6	-0.0002	83
25	1	25.07	1820.19	42.66	-0.0058	78
25	2	28.6	2064.12	45.43	-0.0001	96
25	3	20.09	1176.85	34.31	-0.0064	67
25	4	9.87	466.43	21.6	-0.0	89
50	1	25.08	1820.83	42.67	-0.0061	95
50	2	28.62	2064.06	45.43	-0.0001	72
50	3	20.05	1174.38	34.27	-0.0042	73
50	4	9.88	466.47	21.6	-0.0001	86
100	1	25.17	1826.91	42.74	-0.0095	88
100	2	28.6	2064.06	45.43	-0.0001	69
100	3	20.04	1173.14	34.25	-0.0032	68
100	4	9.87	466.43	21.6	-0.0	71

Table B.11: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.06	3739.47	61.15	-0.0069	103
2	2	42.15	4291.0	65.51	-0.0001	68
2	3	29.38	2162.65	46.5	-0.0001	75
2	4	12.84	559.72	23.66	-0.0	71
5	1	36.08	3741.04	61.16	-0.0073	70
5	2	42.12	4290.77	65.5	-0.0	79
5	3	29.38	2162.51	46.5	-0.0	67
5	4	12.84	559.81	23.66	-0.0002	82
10	1	36.07	3740.07	61.16	-0.007	84
10	2	42.15	4290.97	65.51	-0.0	93
10	3	29.38	2162.62	46.5	-0.0001	82
10	4	12.84	559.72	23.66	-0.0	88
25	1	36.08	3740.66	61.16	-0.0072	104
25	2	42.15	4290.93	65.51	-0.0	68
25	3	29.38	2162.51	46.5	-0.0	68
25	4	12.84	559.77	23.66	-0.0001	67
50	1	36.06	3739.09	61.15	-0.0068	111
50	2	42.16	4291.1	65.51	-0.0001	93
50	3	29.37	2162.78	46.51	-0.0002	80
50	4	12.84	559.82	23.66	-0.0002	69
100	1	35.99	3733.3	61.1	-0.0052	153
100	2	42.12	4290.79	65.5	-0.0	70
100	3	29.38	2162.55	46.5	-0.0001	67
100	4	12.84	559.74	23.66	-0.0	68

Table B.12: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.24	6168.97	78.54	-0.0114	171
2	2	53.53	6324.21	79.52	-0.0002	68
2	3	34.73	3209.49	56.65	-0.0001	83
2	4	14.87	871.24	29.52	-0.0005	95
5	1	45.25	6170.4	78.55	-0.0116	154
5	2	53.54	6324.4	79.53	-0.0002	68
5	3	34.68	3209.31	56.65	-0.0	67
5	4	14.88	871.41	29.52	-0.0007	66
10	1	45.1	6151.36	78.43	-0.0085	101
10	2	53.45	6323.2	79.52	-0.0	88
10	3	34.64	3209.35	56.65	-0.0	84
10	4	14.85	870.93	29.51	-0.0002	79
25	1	45.11	6153.25	78.44	-0.0088	78
25	2	53.44	6323.18	79.52	-0.0	71
25	3	34.64	3209.35	56.65	-0.0	98
25	4	14.85	870.89	29.51	-0.0001	96
50	1	45.1	6151.0	78.43	-0.0084	68
50	2	54.9	6546.26	80.91	-0.0353	69
50	3	35.55	3363.67	58.0	-0.0481	78
50	4	16.12	928.26	30.47	-0.066	96
100	1	45.0	6138.78	78.35	-0.0064	106
100	2	53.47	6323.27	79.52	-0.0	77
100	3	34.68	3209.3	56.65	-0.0	95
100	4	14.86	871.04	29.51	-0.0003	89

Table B.13: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.37	12272.36	110.78	-0.0245	72
2	2	80.88	13360.94	115.59	-0.0021	69
2	3	52.41	6384.9	79.91	-0.0135	73
2	4	22.04	1698.91	41.22	-0.0137	77
5	1	68.7	12165.44	110.3	-0.0156	78
5	2	77.86	12590.38	112.21	0.0557	66
5	3	51.83	6373.55	79.83	-0.0117	77
5	4	21.29	1757.66	41.92	-0.0488	81
10	1	69.25	12253.44	110.7	-0.0229	68
10	2	81.78	13334.69	115.48	-0.0001	68
10	3	51.68	6300.02	79.37	-0.0001	83
10	4	21.65	1676.89	40.95	-0.0006	75
25	1	68.74	12173.98	110.34	-0.0163	66
25	2	81.9	13337.03	115.49	-0.0003	69
25	3	51.84	6299.67	79.37	-0.0	74
25	4	21.82	1679.68	40.98	-0.0023	102
50	1	68.11	11509.1	107.28	0.0392	76
50	2	93.57	15270.37	123.57	-0.1453	70
50	3	56.53	6863.85	82.85	-0.0896	93
50	4	29.57	2148.57	46.35	-0.282	79
100	1	68.18	12026.67	109.67	-0.004	66
100	2	86.54	13240.45	115.07	0.0069	67
100	3	57.89	7093.31	84.22	-0.126	69
100	4	24.03	1876.04	43.31	-0.1194	75

Table B.14: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	102.46	26871.23	163.92	-0.0548	90
2	2	110.99	20225.3	142.22	-0.009	70
2	3	83.21	13843.32	117.66	-0.0596	84
2	4	32.52	3607.76	60.06	-0.099	72
5	1	98.77	25727.67	160.4	-0.0099	81
5	2	110.04	20492.06	143.15	-0.0223	87
5	3	81.79	13840.9	117.65	-0.0595	67
5	4	31.12	3502.14	59.18	-0.0668	82
10	1	98.56	24830.14	157.58	0.0254	73
10	2	111.35	20769.46	144.12	-0.0361	90
10	3	86.0	16376.44	127.97	-0.2535	77
10	4	31.09	3483.86	59.02	-0.0612	69
25	1	102.68	26061.45	161.44	-0.023	68
25	2	111.22	20074.14	141.68	-0.0014	66
25	3	79.04	13144.55	114.65	-0.0061	70
25	4	30.34	3288.33	57.34	-0.0017	67
50	1	120.25	26808.36	163.73	-0.0523	70
50	2	158.57	39895.33	199.74	-0.9903	68
50	3	84.77	15610.98	124.94	-0.1949	92
50	4	47.39	5156.68	71.81	-0.5708	68
100	1	132.01	32582.26	180.51	-0.2789	72
100	2	123.21	23886.89	154.55	-0.1917	69
100	3	95.02	18579.78	136.31	-0.4222	75
100	4	54.45	5499.27	74.16	-0.6751	70

Table B.15: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	173.4	60388.17	245.74	-0.0791	66
2	2	140.42	31798.51	178.32	0.0819	70
2	3	113.34	29558.42	171.93	-0.0453	76
2	4	53.39	8185.43	90.47	-0.0726	80
5	1	172.83	59283.15	243.48	-0.0593	78
5	2	151.8	35055.22	187.23	-0.0121	79
5	3	113.44	30524.7	174.71	-0.0795	72
5	4	53.95	8075.82	89.87	-0.0582	75
10	1	168.16	57633.84	240.07	-0.0299	73
10	2	151.46	36657.7	191.46	-0.0584	80
10	3	110.09	29614.43	172.09	-0.0473	66
10	4	52.36	8150.23	90.28	-0.0679	85
25	1	172.65	57964.59	240.76	-0.0358	76
25	2	148.51	35292.96	187.86	-0.019	66
25	3	134.95	37006.58	192.37	-0.3087	66
25	4	67.11	9582.03	97.89	-0.2556	66
50	1	208.63	92629.76	304.35	-0.6552	70
50	2	193.67	61641.78	248.28	-0.7797	118
50	3	176.53	63172.98	251.34	-1.2341	71
50	4	108.07	17979.54	134.09	-1.3559	68
100	1	184.68	59211.48	243.33	-0.0581	66
100	2	165.65	50589.25	224.92	-0.4606	67
100	3	122.94	31764.79	178.23	-0.1233	70
100	4	104.24	21634.76	147.09	-1.8348	72

Table B.16: Result metrics for the $Base_{td}$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.2 Sentiment Models Results

B.2.1 The Non-Time Differenced Sentiment Model $Sent_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	131.24	23914.5	154.64	0.9548	67
2	2	192.92	43684.7	209.01	0.7318	71
2	3	131.87	19546.51	139.81	0.7996	71
2	4	31.99	1262.29	35.53	0.9052	75
5	1	179.31	47434.5	217.79	0.9103	68
5	2	208.74	52573.5	229.29	0.6773	68
5	3	97.5	13154.64	114.69	0.8651	104
5	4	19.46	610.92	24.72	0.9541	76
10	1	189.2	53032.64	230.29	0.8997	66
10	2	168.63	34956.94	186.97	0.7854	70
10	3	150.04	29806.98	172.65	0.6944	92
10	4	61.27	5058.14	71.12	0.6201	67
25	1	168.89	42795.82	206.87	0.9191	113
25	2	99.53	17320.28	131.61	0.8937	86
25	3	89.04	19255.4	138.76	0.8026	68
25	4	49.47	3191.77	56.5	0.7603	69
50	1	166.32	44339.42	210.57	0.9162	124
50	2	153.14	28538.36	168.93	0.8248	81
50	3	150.79	42698.94	206.64	0.5622	68
50	4	59.57	4874.48	69.82	0.6339	68
100	1	146.53	27734.24	166.54	0.9476	106
100	2	215.03	64671.78	254.31	0.603	72
100	3	97.75	20084.79	141.72	0.7941	82
100	4	102.37	12586.55	112.19	0.0547	81

Table B.17: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	73.72	8609.45	92.79	0.9837	76
2	2	42.73	2827.43	53.17	0.9826	82
2	3	24.3	1200.52	34.65	0.9877	81
2	4	8.92	319.65	17.88	0.9759	68
5	1	62.68	9551.31	97.73	0.9819	75
5	2	37.55	2802.8	52.94	0.9827	93
5	3	30.39	2267.64	47.62	0.9767	84
5	4	21.97	846.93	29.1	0.9362	71
10	1	92.58	17717.42	133.11	0.9665	97
10	2	42.76	3726.2	61.04	0.977	83
10	3	41.38	2811.23	53.02	0.9711	90
10	4	13.44	599.4	24.48	0.9549	79
25	1	82.73	9862.65	99.31	0.9813	71
25	2	62.88	6170.58	78.55	0.962	78
25	3	45.81	4115.14	64.15	0.9578	68
25	4	45.43	2708.05	52.04	0.7961	79
50	1	102.39	14269.83	119.46	0.973	76
50	2	64.04	6647.26	81.53	0.959	68
50	3	46.89	4015.4	63.37	0.9588	78
50	4	37.65	2320.4	48.17	0.8253	77
100	1	114.24	19182.12	138.5	0.9637	96
100	2	87.19	10952.35	104.65	0.9325	68
100	3	64.1	6560.76	81.0	0.9326	79
100	4	33.87	1995.0	44.67	0.8498	67

Table B.18: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	40.03	3417.32	58.46	0.9935	67
2	2	46.33	3614.34	60.12	0.9776	81
2	3	26.81	1641.09	40.51	0.9831	87
2	4	32.91	1546.94	39.33	0.883	70
5	1	56.01	5309.89	72.87	0.99	107
5	2	38.91	3168.67	56.29	0.9803	75
5	3	27.02	1858.82	43.11	0.9808	93
5	4	48.72	3079.45	55.49	0.7672	72
10	1	67.34	7471.97	86.44	0.9859	96
10	2	56.56	5270.35	72.6	0.9673	71
10	3	42.15	3685.41	60.71	0.962	66
10	4	42.39	2540.53	50.4	0.8079	74
25	1	66.18	7990.43	89.39	0.9849	67
25	2	94.07	12083.83	109.93	0.925	73
25	3	40.56	3346.8	57.85	0.9655	80
25	4	23.46	1264.74	35.56	0.9044	74
50	1	121.89	20983.47	144.86	0.9603	74
50	2	124.59	22627.5	150.42	0.8595	75
50	3	59.07	5839.9	76.42	0.9398	72
50	4	41.91	3048.12	55.21	0.7695	82
100	1	112.19	21196.67	145.59	0.9599	73
100	2	135.02	25390.84	159.35	0.8424	81
100	3	54.83	5153.17	71.79	0.9469	89
100	4	42.29	3041.04	55.15	0.7701	80

Table B.19: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	87.64	14013.02	118.38	0.9735	97
2	2	46.08	4635.02	68.08	0.971	70
2	3	33.86	2689.56	51.86	0.972	82
2	4	15.72	715.06	26.74	0.9455	97
5	1	103.92	19560.36	139.86	0.9629	80
5	2	51.23	5481.35	74.04	0.9657	76
5	3	38.66	3182.32	56.41	0.9668	71
5	4	16.94	809.56	28.45	0.9383	70
10	1	100.75	18949.43	137.66	0.9641	85
10	2	60.55	6820.74	82.59	0.9573	84
10	3	37.72	3075.96	55.46	0.9679	88
10	4	20.93	1005.58	31.71	0.9234	77
25	1	91.11	13597.58	116.61	0.9742	129
25	2	64.03	7591.94	87.13	0.9525	75
25	3	48.8	4582.41	67.69	0.9522	76
25	4	26.87	1520.1	38.99	0.8842	86
50	1	146.44	34689.27	186.25	0.9343	80
50	2	90.42	13161.52	114.72	0.9177	73
50	3	54.68	5940.82	77.08	0.9381	75
50	4	34.74	2423.67	49.23	0.8154	66
100	1	139.48	28460.14	168.7	0.9461	76
100	2	100.04	16623.64	128.93	0.896	111
100	3	62.63	7018.17	83.77	0.9268	78
100	4	40.19	2882.17	53.69	0.7805	68

Table B.20: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	100.09	17233.48	131.28	0.9671	73
2	2	55.87	6854.91	82.79	0.957	68
2	3	39.29	3509.92	59.24	0.9638	66
2	4	19.11	1124.92	33.54	0.9141	72
5	1	81.59	12125.87	110.12	0.9768	75
5	2	60.36	7586.82	87.1	0.9524	69
5	3	41.2	3817.46	61.79	0.9606	79
5	4	20.71	1287.62	35.88	0.9017	77
10	1	77.77	11051.25	105.12	0.9789	73
10	2	64.62	8326.68	91.25	0.9477	101
10	3	45.15	4354.7	65.99	0.9551	66
10	4	24.84	1505.16	38.8	0.885	85
25	1	81.28	11953.67	109.33	0.9772	85
25	2	75.3	10351.21	101.74	0.935	92
25	3	51.14	5118.05	71.54	0.9472	75
25	4	28.55	1884.1	43.41	0.8561	98
50	1	137.43	27328.05	165.31	0.9478	78
50	2	93.31	15085.73	122.82	0.9053	93
50	3	65.85	9035.86	95.06	0.9069	78
50	4	33.97	2403.5	49.03	0.8164	94
100	1	243.03	76435.2	276.47	0.854	76
100	2	107.77	19871.32	140.97	0.8753	81
100	3	80.46	11487.86	107.18	0.8816	79
100	4	42.84	3412.93	58.42	0.7393	67

Table B.21: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	124.84	25080.76	158.37	0.9523	97
2	2	82.04	14088.84	118.7	0.9102	85
2	3	57.0	6901.16	83.07	0.9269	74
2	4	25.29	1970.96	44.4	0.847	68
5	1	91.07	14913.02	122.12	0.9716	73
5	2	82.01	13664.43	116.89	0.9129	66
5	3	56.74	6949.58	83.36	0.9264	66
5	4	26.19	1998.01	44.7	0.8449	68
10	1	106.7	18633.54	136.5	0.9645	121
10	2	87.06	14327.0	119.7	0.9087	76
10	3	68.42	9032.94	95.04	0.9044	96
10	4	33.53	2444.5	49.44	0.8103	96
25	1	136.91	29426.74	171.54	0.944	78
25	2	116.91	23275.63	152.56	0.8517	67
25	3	81.16	12358.12	111.17	0.8692	66
25	4	41.23	3217.2	56.72	0.7503	70
50	1	228.16	68498.3	261.72	0.8696	80
50	2	141.53	34637.41	186.11	0.7793	81
50	3	119.82	22641.71	150.47	0.7603	68
50	4	54.23	5082.79	71.29	0.6055	70
100	1	294.44	131985.1	363.3	0.7488	71
100	2	178.73	45856.34	214.14	0.7078	84
100	3	123.08	35791.24	189.19	0.6211	73
100	4	58.32	5923.99	76.97	0.5402	76

Table B.22: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	141.89	35469.86	188.33	0.9326	66
2	2	108.27	19677.67	140.28	0.8698	78
2	3	82.09	13736.27	117.2	0.8427	73
2	4	40.88	3877.5	62.27	0.708	73
5	1	126.47	29701.27	172.34	0.9436	86
5	2	124.15	24975.67	158.04	0.8348	71
5	3	87.33	15118.33	122.96	0.8269	80
5	4	44.98	4625.46	68.01	0.6517	81
10	1	148.09	36301.45	190.53	0.931	71
10	2	158.7	38227.75	195.52	0.7471	90
10	3	112.99	23314.68	152.69	0.733	79
10	4	60.9	7365.4	85.82	0.4454	72
25	1	178.82	50908.58	225.63	0.9033	87
25	2	210.2	69949.69	264.48	0.5372	67
25	3	143.01	38724.07	196.78	0.5565	74
25	4	69.34	8293.91	91.07	0.3755	85
50	1	406.21	203511.0	451.12	0.6134	73
50	2	145.06	34130.68	184.74	0.7742	73
50	3	160.07	43276.42	208.03	0.5044	82
50	4	96.54	16201.33	127.28	-0.22	72
100	1	270.67	103069.82	321.04	0.8042	68
100	2	149.68	36342.58	190.64	0.7596	69
100	3	225.64	89444.52	299.07	-0.0243	69
100	4	111.85	21221.02	145.67	-0.598	87

Table B.23: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	201.22	62120.88	249.24	0.8779	66
2	2	148.87	34895.28	186.8	0.7519	91
2	3	108.75	27638.8	166.25	0.6658	68
2	4	56.64	7166.42	84.65	0.4952	67
5	1	200.65	63840.7	252.67	0.8745	66
5	2	251.44	94682.23	307.7	0.3268	66
5	3	133.47	36960.24	192.25	0.5531	81
5	4	126.53	25314.93	159.11	-0.7832	78
10	1	236.33	86333.03	293.82	0.8303	69
10	2	291.65	138998.4	372.82	0.0117	83
10	3	180.78	53517.04	231.34	0.353	78
10	4	177.08	42695.92	206.63	-2.0076	77
25	1	473.41	335150.47	578.92	0.3413	80
25	2	393.43	263088.84	512.92	-0.8705	104
25	3	169.26	52043.56	228.13	0.3708	67
25	4	186.85	66210.33	257.31	-3.664	68
50	1	247.73	102242.85	319.75	0.7991	66
50	2	278.06	127373.02	356.89	0.0944	70
50	3	244.68	102714.65	320.49	-0.2418	84
50	4	170.04	38969.19	197.41	-1.7451	68
100	1	752.32	964449.94	982.06	-0.8954	65
100	2	352.35	187079.4	432.53	-0.3301	78
100	3	200.1	76329.74	276.28	0.0772	69
100	4	123.24	30143.78	173.62	-1.1234	67

Table B.24: Result metrics for the $Sent_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.2.2 The Time Differenced Sentiment Model $Sent_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.89	17.26	-0.0006	152
2	2	11.83	363.36	19.06	-0.0017	138
2	3	8.45	214.98	14.66	-0.0008	173
2	4	4.56	91.61	9.57	-0.0009	106
5	1	10.48	297.82	17.26	-0.0004	290
5	2	11.83	363.36	19.06	-0.0017	164
5	3	8.45	215.16	14.67	-0.0016	133
5	4	4.7	92.9	9.64	-0.0149	79
10	1	10.48	297.85	17.26	-0.0005	262
10	2	11.84	363.57	19.07	-0.0023	170
10	3	8.45	214.98	14.66	-0.0008	165
10	4	4.55	91.56	9.57	-0.0004	67
25	1	10.48	297.79	17.26	-0.0003	205
25	2	11.83	363.23	19.06	-0.0013	227
25	3	8.45	214.99	14.66	-0.0008	86
25	4	4.58	91.76	9.58	-0.0025	84
50	1	10.48	297.8	17.26	-0.0003	151
50	2	11.83	363.23	19.06	-0.0013	99
50	3	8.45	215.01	14.66	-0.0009	213
50	4	4.56	91.62	9.57	-0.001	78
100	1	10.48	297.77	17.26	-0.0002	149
100	2	11.83	363.11	19.06	-0.001	183
100	3	8.45	214.98	14.66	-0.0008	102
100	4	4.59	91.84	9.58	-0.0034	73

Table B.25: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.48	906.27	30.1	-0.0013	67
2	2	19.83	1010.89	31.79	-0.001	81
2	3	14.77	742.62	27.25	-0.006	78
2	4	7.4	263.42	16.23	-0.0005	72
5	1	17.47	906.0	30.1	-0.001	67
5	2	19.84	1010.97	31.8	-0.001	71
5	3	14.86	745.6	27.31	-0.0101	78
5	4	7.48	264.26	16.26	-0.0037	67
10	1	17.48	906.46	30.11	-0.0015	79
10	2	19.84	1011.28	31.8	-0.0013	77
10	3	14.82	744.26	27.28	-0.0083	99
10	4	7.49	264.42	16.26	-0.0043	94
25	1	17.51	907.84	30.13	-0.0031	80
25	2	19.84	1010.95	31.8	-0.001	78
25	3	14.75	742.19	27.24	-0.0055	72
25	4	7.51	264.68	16.27	-0.0053	70
50	1	17.47	906.06	30.1	-0.0011	96
50	2	19.83	1010.74	31.79	-0.0008	74
50	3	14.85	745.06	27.3	-0.0094	79
50	4	7.61	265.99	16.31	-0.0102	72
100	1	17.47	906.09	30.1	-0.0011	89
100	2	19.84	1010.37	31.79	-0.0004	69
100	3	14.82	744.37	27.28	-0.0084	88
100	4	7.5	264.54	16.26	-0.0048	69

Table B.26: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.12	1823.51	42.7	-0.0076	78
2	2	28.62	2063.99	45.43	-0.0	75
2	3	20.11	1178.11	34.32	-0.0074	68
2	4	9.87	466.44	21.6	-0.0	69
5	1	25.1	1822.31	42.69	-0.007	68
5	2	28.61	2063.96	45.43	-0.0	89
5	3	20.06	1174.94	34.28	-0.0047	74
5	4	9.87	466.44	21.6	-0.0	76
10	1	25.06	1819.25	42.65	-0.0053	80
10	2	28.67	2065.0	45.44	-0.0005	78
10	3	20.02	1170.49	34.21	-0.0009	73
10	4	9.87	466.46	21.6	-0.0001	85
25	1	25.06	1819.62	42.66	-0.0055	83
25	2	28.63	2064.18	45.43	-0.0001	71
25	3	20.05	1173.81	34.26	-0.0038	69
25	4	9.87	466.43	21.6	-0.0	124
50	1	25.07	1820.31	42.67	-0.0059	88
50	2	28.6	2064.01	45.43	-0.0	72
50	3	20.06	1174.97	34.28	-0.0048	69
50	4	9.88	466.49	21.6	-0.0001	79
100	1	25.05	1818.23	42.64	-0.0047	96
100	2	28.63	2064.24	45.43	-0.0001	72
100	3	20.05	1174.48	34.27	-0.0043	86
100	4	9.87	466.43	21.6	-0.0	67

Table B.27: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.07	3739.57	61.15	-0.0069	123
2	2	42.16	4291.12	65.51	-0.0001	88
2	3	29.38	2162.5	46.5	-0.0	78
2	4	12.84	559.72	23.66	-0.0	66
5	1	36.08	3741.15	61.16	-0.0073	67
5	2	42.15	4290.96	65.51	-0.0	106
5	3	29.38	2162.53	46.5	-0.0001	87
5	4	12.84	559.79	23.66	-0.0001	69
10	1	36.07	3740.17	61.16	-0.0071	107
10	2	42.15	4291.02	65.51	-0.0001	84
10	3	29.38	2162.54	46.5	-0.0001	80
10	4	12.84	559.72	23.66	-0.0	77
25	1	36.08	3740.72	61.16	-0.0072	103
25	2	42.15	4290.93	65.51	-0.0	66
25	3	29.38	2162.5	46.5	-0.0	120
25	4	12.84	559.74	23.66	-0.0	67
50	1	36.06	3738.79	61.15	-0.0067	120
50	2	42.15	4290.95	65.51	-0.0	114
50	3	29.38	2162.67	46.5	-0.0001	92
50	4	12.84	559.81	23.66	-0.0002	73
100	1	35.99	3733.59	61.1	-0.0053	126
100	2	42.11	4290.76	65.5	0.0	75
100	3	29.38	2162.58	46.5	-0.0001	86
100	4	12.84	559.75	23.66	-0.0001	69

Table B.28: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.24	6168.64	78.54	-0.0113	118
2	2	53.55	6324.53	79.53	-0.0002	80
2	3	34.72	3209.44	56.65	-0.0	99
2	4	14.88	871.35	29.52	-0.0007	93
5	1	45.25	6170.22	78.55	-0.0116	156
5	2	53.53	6324.24	79.53	-0.0002	105
5	3	34.7	3209.35	56.65	-0.0	92
5	4	14.87	871.34	29.52	-0.0006	67
10	1	45.1	6151.26	78.43	-0.0085	71
10	2	53.45	6323.19	79.52	-0.0	66
10	3	34.64	3209.36	56.65	-0.0	108
10	4	14.85	870.94	29.51	-0.0002	78
25	1	45.11	6153.11	78.44	-0.0088	79
25	2	53.44	6323.2	79.52	-0.0	71
25	3	34.64	3209.34	56.65	-0.0	75
25	4	14.85	870.92	29.51	-0.0002	73
50	1	45.07	6147.28	78.4	-0.0078	67
50	2	59.83	7170.91	84.68	-0.1341	101
50	3	36.34	3565.04	59.71	-0.1108	117
50	4	17.95	1215.61	34.87	-0.396	105
100	1	45.0	6138.74	78.35	-0.0064	97
100	2	53.48	6323.4	79.52	-0.0	66
100	3	34.69	3209.31	56.65	-0.0	75
100	4	14.85	870.99	29.51	-0.0002	69

Table B.29: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.46	12005.83	109.57	-0.0022	67
2	2	82.05	13862.14	117.74	-0.0397	101
2	3	52.61	6344.17	79.65	-0.0071	69
2	4	24.22	1917.21	43.79	-0.144	69
5	1	70.28	12336.94	111.07	-0.0299	66
5	2	79.85	13258.04	115.14	0.0056	98
5	3	52.73	6481.79	80.51	-0.0289	70
5	4	26.51	2078.66	45.59	-0.2403	80
10	1	69.24	12252.42	110.69	-0.0228	66
10	2	85.4	14395.73	119.98	-0.0797	85
10	3	51.97	6780.37	82.34	-0.0763	79
10	4	25.0	1960.79	44.28	-0.17	69
25	1	68.71	12167.12	110.3	-0.0157	70
25	2	81.91	13337.06	115.49	-0.0003	71
25	3	51.84	6299.66	79.37	-0.0	85
25	4	21.82	1679.68	40.98	-0.0023	90
50	1	68.7	12327.53	111.03	-0.0291	66
50	2	97.54	16770.24	129.5	-0.2578	87
50	3	84.66	12894.52	113.55	-1.0469	68
50	4	66.77	7524.96	86.75	-3.4901	71
100	1	68.18	12029.1	109.68	-0.0042	66
100	2	81.02	13342.63	115.51	-0.0007	81
100	3	51.09	6310.76	79.44	-0.0018	90
100	4	21.43	1676.45	40.94	-0.0003	67

Table B.30: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	106.0	26895.74	164.0	-0.0557	69
2	2	110.45	20553.88	143.37	-0.0254	80
2	3	86.9	15034.25	122.61	-0.1508	67
2	4	42.02	4553.5	67.48	-0.387	70
5	1	103.89	26439.38	162.6	-0.0378	67
5	2	117.57	24167.34	155.46	-0.2056	72
5	3	90.09	16363.66	127.92	-0.2526	69
5	4	50.16	6024.04	77.61	-0.835	92
10	1	102.38	26319.74	162.23	-0.0331	66
10	2	115.54	24640.22	156.97	-0.2292	96
10	3	112.63	21719.74	147.38	-0.6625	68
10	4	65.62	8610.97	92.8	-1.623	74
25	1	102.68	26061.47	161.44	-0.023	67
25	2	111.22	20074.24	141.68	-0.0015	74
25	3	79.04	13144.73	114.65	-0.0062	68
25	4	30.34	3288.32	57.34	-0.0017	94
50	1	110.04	27809.46	166.76	-0.0916	76
50	2	159.31	43019.72	207.41	-1.1461	74
50	3	145.31	35705.4	188.96	-1.7331	101
50	4	85.51	12368.98	111.22	-2.7677	68
100	1	109.59	27944.02	167.16	-0.0969	66
100	2	112.73	20250.8	142.31	-0.0103	80
100	3	145.35	35988.48	189.71	-1.7547	92
100	4	74.22	8464.85	92.0	-1.5785	68

Table B.31: Result metrics for the $Sent_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	168.51	62008.98	249.02	-0.108	89
2	2	147.41	33101.53	181.94	0.0443	68
2	3	130.37	39924.5	199.81	-0.4119	81
2	4	84.58	13008.87	114.06	-0.7046	75
5	1	174.01	58152.38	241.15	-0.0391	66
5	2	170.95	45347.65	212.95	-0.3093	76
5	3	130.25	38710.56	196.75	-0.369	66
5	4	103.55	16809.79	129.65	-1.2026	73
10	1	183.78	63903.96	252.79	-0.1419	74
10	2	172.94	48948.4	221.24	-0.4132	73
10	3	110.36	31386.31	177.16	-0.11	68
10	4	92.62	17202.17	131.16	-1.254	77
25	1	172.52	57619.96	240.04	-0.0296	66
25	2	249.69	107843.02	328.39	-2.1136	85
25	3	222.03	84029.88	289.88	-1.9717	69
25	4	117.17	21086.86	145.21	-1.7631	67
50	1	178.2	63465.86	251.92	-0.1341	78
50	2	188.22	58901.04	242.7	-0.7006	78
50	3	205.15	74876.25	273.64	-1.648	84
50	4	103.38	19459.6	139.5	-1.5498	66
100	1	185.19	59336.98	243.59	-0.0603	74
100	2	149.1	36779.89	191.78	-0.0619	74
100	3	124.6	33191.78	182.19	-0.1738	94
100	4	116.64	22710.18	150.7	-1.9758	77

Table B.32: Result metrics for the $Sent_{td}$ trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.2.3 The Non-Time Differenced Compound Sentiment Model $Comp_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	92.29	11676.92	108.06	0.9779	70
2	2	169.17	33401.53	182.76	0.795	70
2	3	87.57	9280.31	96.33	0.9049	84
2	4	36.43	1519.12	38.98	0.8859	80
5	1	165.53	38199.78	195.45	0.9278	70
5	2	123.51	18507.43	136.04	0.8864	71
5	3	112.9	15466.57	124.36	0.8414	100
5	4	51.56	3113.87	55.8	0.7661	80
10	1	207.59	61355.0	247.7	0.884	74
10	2	103.18	13311.46	115.38	0.9183	95
10	3	111.39	16279.11	127.59	0.8331	104
10	4	66.95	5530.88	74.37	0.5846	81
25	1	230.51	79663.7	282.25	0.8494	66
25	2	84.26	12325.87	111.02	0.9243	72
25	3	112.75	22114.64	148.71	0.7733	80
25	4	69.53	5199.34	72.11	0.6095	86
50	1	232.56	85936.2	293.15	0.8375	69
50	2	137.97	23512.26	153.34	0.8557	76
50	3	97.75	16562.92	128.7	0.8302	70
50	4	58.12	4280.1	65.42	0.6786	68
100	1	260.67	98828.16	314.37	0.8131	105
100	2	108.37	18868.88	137.36	0.8842	73
100	3	116.37	22373.55	149.58	0.7706	104
100	4	84.81	8261.75	90.89	0.3795	79

Table B.33: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	67.07	8168.29	90.38	0.9846	76
2	2	29.66	1692.65	41.14	0.9896	81
2	3	25.18	1306.03	36.14	0.9866	75
2	4	12.52	426.99	20.66	0.9678	93
5	1	58.46	5004.22	70.74	0.9905	80
5	2	30.65	2027.59	45.03	0.9875	66
5	3	19.5	1138.42	33.74	0.9883	85
5	4	12.17	461.35	21.48	0.9653	80
10	1	74.43	10484.88	102.4	0.9802	89
10	2	31.78	2127.99	46.13	0.9869	83
10	3	29.62	1773.43	42.11	0.9818	83
10	4	10.04	414.93	20.37	0.9688	70
25	1	53.9	5538.51	74.42	0.9895	66
25	2	52.96	4722.69	68.72	0.9709	74
25	3	34.86	2425.18	49.25	0.9751	71
25	4	17.12	748.06	27.35	0.9437	88
50	1	71.57	8519.39	92.3	0.9839	71
50	2	51.05	4550.58	67.46	0.9719	68
50	3	35.47	2648.98	51.47	0.9728	67
50	4	20.87	978.91	31.29	0.9263	80
100	1	168.43	42196.64	205.42	0.9202	82
100	2	69.93	7878.82	88.76	0.9514	93
100	3	80.33	9880.58	99.4	0.8986	82
100	4	17.62	802.62	28.33	0.9396	66

Table B.34: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	49.59	4818.01	69.41	0.9909	79
2	2	55.07	4473.93	66.89	0.9722	79
2	3	24.81	1524.78	39.05	0.9843	69
2	4	35.75	1698.99	41.22	0.8715	71
5	1	67.85	7317.21	85.54	0.9862	71
5	2	54.8	4594.26	67.78	0.9715	70
5	3	23.82	1602.19	40.03	0.9835	87
5	4	14.08	659.13	25.67	0.9502	77
10	1	60.52	6624.46	81.39	0.9875	85
10	2	53.11	4569.9	67.6	0.9716	84
10	3	27.22	1841.2	42.91	0.981	83
10	4	65.7	4897.51	69.98	0.6297	70
25	1	88.35	12481.06	111.72	0.9764	68
25	2	46.99	4502.41	67.1	0.9721	66
25	3	30.55	2318.86	48.15	0.9761	69
25	4	18.3	861.63	29.35	0.9349	73
50	1	79.46	9732.7	98.65	0.9816	76
50	2	79.64	9094.14	95.36	0.9435	82
50	3	70.75	7380.46	85.91	0.9239	69
50	4	39.46	2199.94	46.9	0.8337	69
100	1	124.85	23595.73	153.61	0.9554	67
100	2	127.85	22624.57	150.41	0.8596	82
100	3	80.85	9107.76	95.43	0.9061	68
100	4	29.2	1531.84	39.14	0.8842	67

Table B.35: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	81.22	12235.15	110.61	0.9768	75
2	2	46.2	4626.36	68.02	0.9711	94
2	3	32.77	2558.78	50.58	0.9733	80
2	4	14.39	645.36	25.4	0.9508	75
5	1	93.53	15989.12	126.45	0.9697	72
5	2	50.19	5215.67	72.22	0.9674	71
5	3	33.13	2723.35	52.19	0.9716	67
5	4	15.76	750.33	27.39	0.9429	101
10	1	94.66	16064.07	126.74	0.9696	68
10	2	55.21	6024.11	77.62	0.9623	79
10	3	35.18	2793.02	52.85	0.9709	73
10	4	19.98	946.51	30.77	0.9279	74
25	1	89.38	13034.92	114.17	0.9753	68
25	2	62.69	7162.26	84.63	0.9552	66
25	3	42.34	4009.63	63.32	0.9582	74
25	4	24.3	1261.03	35.51	0.904	67
50	1	128.92	26835.84	163.82	0.9492	76
50	2	67.58	8479.32	92.08	0.947	77
50	3	43.85	4556.63	67.5	0.9525	80
50	4	33.22	2061.32	45.4	0.843	79
100	1	119.09	22731.44	150.77	0.9569	77
100	2	77.62	11573.04	107.58	0.9276	67
100	3	47.88	4736.84	68.82	0.9506	67
100	4	33.59	2102.55	45.85	0.8399	78

Table B.36: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	72.78	9524.7	97.59	0.9818	69
2	2	54.95	6727.96	82.02	0.9578	72
2	3	41.22	3806.28	61.7	0.9608	82
2	4	17.21	1027.17	32.05	0.9216	83
5	1	80.92	11823.11	108.73	0.9774	69
5	2	59.19	7349.28	85.73	0.9539	75
5	3	41.44	4225.8	65.01	0.9564	78
5	4	21.34	1239.74	35.21	0.9053	85
10	1	85.42	12648.58	112.47	0.9758	73
10	2	61.8	7774.31	88.17	0.9512	78
10	3	43.29	4313.96	65.68	0.9555	74
10	4	20.9	1213.52	34.84	0.9073	70
25	1	89.16	12628.0	112.37	0.9759	78
25	2	67.35	8936.92	94.54	0.9439	72
25	3	43.33	4511.46	67.17	0.9535	68
25	4	24.88	1498.44	38.71	0.8856	71
50	1	125.34	23249.06	152.48	0.9556	72
50	2	76.43	11763.8	108.46	0.9262	80
50	3	51.78	6622.94	81.38	0.9317	79
50	4	28.93	1858.45	43.11	0.8581	86
100	1	165.33	37368.63	193.31	0.9286	81
100	2	85.14	13606.86	116.65	0.9146	76
100	3	55.1	7349.77	85.73	0.9242	73
100	4	25.29	1635.61	40.44	0.8751	70

Table B.37: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	101.47	17111.57	130.81	0.9674	66
2	2	81.85	13183.23	114.82	0.916	72
2	3	54.19	6956.95	83.41	0.9264	69
2	4	23.39	1827.14	42.75	0.8582	74
5	1	103.24	17824.5	133.51	0.9661	70
5	2	84.99	14034.6	118.47	0.9106	83
5	3	54.97	6694.61	81.82	0.9291	69
5	4	23.11	1788.61	42.29	0.8612	66
10	1	98.46	15687.09	125.25	0.9701	97
10	2	85.94	14345.58	119.77	0.9086	92
10	3	58.75	7603.85	87.2	0.9195	84
10	4	26.8	1960.33	44.28	0.8479	67
25	1	121.66	22337.63	149.46	0.9575	74
25	2	97.03	16788.25	129.57	0.893	71
25	3	63.18	8905.32	94.37	0.9057	74
25	4	28.57	2236.18	47.29	0.8264	69
50	1	146.51	31480.2	177.43	0.9401	86
50	2	102.93	19181.92	138.5	0.8778	70
50	3	70.61	10369.77	101.83	0.8902	81
50	4	34.71	2761.03	52.55	0.7857	66
100	1	200.79	54033.84	232.45	0.8971	82
100	2	138.53	34577.79	185.95	0.7796	72
100	3	83.22	17988.38	134.12	0.8096	71
100	4	38.79	3250.41	57.01	0.7477	79

Table B.38: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	133.42	31496.37	177.47	0.9402	74
2	2	108.95	19921.54	141.14	0.8682	70
2	3	76.15	12360.49	111.18	0.8584	92
2	4	34.59	3389.47	58.22	0.7448	92
5	1	127.02	30484.01	174.6	0.9421	107
5	2	109.82	20319.01	142.54	0.8656	76
5	3	75.63	12858.89	113.4	0.8527	71
5	4	36.43	3722.0	61.01	0.7197	73
10	1	144.05	32742.89	180.95	0.9378	68
10	2	120.94	25197.66	158.74	0.8333	80
10	3	93.85	16925.03	130.1	0.8062	69
10	4	39.61	4480.24	66.93	0.6626	73
25	1	232.71	70854.35	266.18	0.8654	105
25	2	140.35	29944.64	173.05	0.8019	74
25	3	99.51	17874.66	133.7	0.7953	66
25	4	50.26	5393.11	73.44	0.5939	101
50	1	297.21	111579.54	334.04	0.7881	78
50	2	132.3	26181.4	161.81	0.8268	71
50	3	117.16	29047.88	170.43	0.6673	78
50	4	60.36	7277.97	85.31	0.452	94
100	1	245.08	88472.99	297.44	0.832	110
100	2	186.88	54436.8	233.32	0.6399	66
100	3	160.03	45000.09	212.13	0.4847	74
100	4	194.75	51062.26	225.97	-2.8451	72

Table B.39: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	182.06	57513.91	239.82	0.887	66
2	2	126.5	29774.07	172.55	0.7883	89
2	3	111.9	28544.57	168.95	0.6549	74
2	4	49.36	7898.68	88.87	0.4436	74
5	1	181.44	60234.14	245.43	0.8816	66
5	2	275.8	113131.48	336.35	0.1957	73
5	3	121.04	30509.07	174.67	0.6311	66
5	4	88.74	14742.83	121.42	-0.0385	67
10	1	210.66	66568.12	258.01	0.8692	67
10	2	505.09	345149.25	587.49	-1.454	66
10	3	138.92	38815.73	197.02	0.5307	70
10	4	105.04	19168.58	138.45	-0.3503	81
25	1	406.25	215453.36	464.17	0.5766	115
25	2	576.39	454365.38	674.07	-2.2305	92
25	3	179.71	55136.64	234.81	0.3334	73
25	4	146.66	39841.53	199.6	-1.8065	70
50	1	308.8	162910.0	403.62	0.6798	66
50	2	246.86	99017.9	314.67	0.296	70
50	3	186.03	62060.82	249.12	0.2497	66
50	4	100.08	18581.72	136.31	-0.3089	69
100	1	752.17	964178.5	981.93	-0.8949	65
100	2	454.8	267787.38	517.48	-0.9039	66
100	3	179.04	51488.65	226.91	0.3775	86
100	4	99.64	17421.49	131.99	-0.2272	67

Table B.40: Result metrics for the $Comp_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.2.4 The Time Differenced Compound Sentiment Model $Comp_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.81	17.26	-0.0004	331
2	2	11.83	363.38	19.06	-0.0018	271
2	3	8.45	215.11	14.67	-0.0014	232
2	4	4.58	91.8	9.58	-0.003	114
5	1	10.48	297.84	17.26	-0.0005	271
5	2	11.83	363.14	19.06	-0.0011	280
5	3	8.45	215.01	14.66	-0.0009	142
5	4	4.56	91.65	9.57	-0.0013	67
10	1	10.48	297.76	17.26	-0.0002	130
10	2	11.83	363.09	19.05	-0.001	240
10	3	8.45	215.0	14.66	-0.0008	88
10	4	4.55	91.55	9.57	-0.0002	70
25	1	10.48	297.84	17.26	-0.0005	323
25	2	11.84	363.78	19.07	-0.0029	156
25	3	8.46	215.24	14.67	-0.002	69
25	4	4.66	92.46	9.62	-0.0102	132
50	1	10.48	297.79	17.26	-0.0003	233
50	2	11.83	363.22	19.06	-0.0013	132
50	3	8.46	215.23	14.67	-0.0019	127
50	4	4.6	91.94	9.59	-0.0045	112
100	1	10.48	297.8	17.26	-0.0004	375
100	2	11.84	363.69	19.07	-0.0026	242
100	3	8.46	215.23	14.67	-0.0019	95
100	4	4.65	92.35	9.61	-0.009	214

Table B.41: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.52	908.03	30.13	-0.0033	89
2	2	19.84	1010.97	31.8	-0.001	81
2	3	14.76	742.44	27.25	-0.0058	73
2	4	7.36	263.34	16.23	-0.0002	70
5	1	17.51	907.72	30.13	-0.0029	72
5	2	19.83	1010.65	31.79	-0.0007	75
5	3	14.79	743.43	27.27	-0.0071	79
5	4	7.37	263.3	16.23	-0.0	81
10	1	17.51	907.57	30.13	-0.0028	71
10	2	19.83	1010.72	31.79	-0.0008	90
10	3	14.83	744.7	27.29	-0.0089	74
10	4	7.38	263.3	16.23	-0.0	69
25	1	17.51	907.83	30.13	-0.0031	67
25	2	19.83	1010.8	31.79	-0.0009	86
25	3	14.86	745.43	27.3	-0.0098	76
25	4	7.38	263.3	16.23	-0.0	96
50	1	17.51	907.53	30.13	-0.0027	89
50	2	19.84	1010.36	31.79	-0.0004	67
50	3	14.85	745.22	27.3	-0.0096	68
50	4	7.4	263.43	16.23	-0.0005	69
100	1	17.52	907.98	30.13	-0.0032	70
100	2	19.84	1010.31	31.79	-0.0004	66
100	3	14.79	743.17	27.26	-0.0068	121
100	4	7.37	263.3	16.23	-0.0	94

Table B.42: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.05	1818.24	42.64	-0.0047	74
2	2	28.65	2064.58	45.44	-0.0003	88
2	3	20.04	1173.29	34.25	-0.0033	82
2	4	9.87	466.43	21.6	-0.0	66
5	1	25.01	1815.3	42.61	-0.0031	71
5	2	28.66	2064.77	45.44	-0.0004	68
5	3	20.06	1174.6	34.27	-0.0044	80
5	4	9.87	466.43	21.6	-0.0	74
10	1	25.1	1821.88	42.68	-0.0067	75
10	2	28.69	2065.86	45.45	-0.0009	86
10	3	20.06	1175.36	34.28	-0.0051	86
10	4	9.87	466.48	21.6	-0.0001	78
25	1	25.08	1820.91	42.67	-0.0062	95
25	2	28.63	2064.21	45.43	-0.0001	66
25	3	20.06	1175.15	34.28	-0.0049	90
25	4	9.88	466.49	21.6	-0.0002	77
50	1	25.09	1821.23	42.68	-0.0064	88
50	2	28.66	2064.79	45.44	-0.0004	68
50	3	20.03	1172.23	34.24	-0.0024	66
50	4	9.87	466.43	21.6	-0.0	68
100	1	25.13	1824.08	42.71	-0.0079	70
100	2	28.67	2065.13	45.44	-0.0006	68
100	3	20.04	1172.93	34.25	-0.003	68
100	4	9.87	466.44	21.6	-0.0	75

Table B.43: Result metrics for the the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.06	3739.47	61.15	-0.0069	99
2	2	42.16	4291.15	65.51	-0.0001	69
2	3	29.38	2162.44	46.5	-0.0	70
2	4	12.84	559.72	23.66	-0.0	86
5	1	36.09	3741.18	61.17	-0.0073	69
5	2	42.18	4291.32	65.51	-0.0001	89
5	3	29.45	2176.77	46.66	-0.0066	66
5	4	12.85	560.07	23.67	-0.0006	67
10	1	36.08	3740.48	61.16	-0.0071	121
10	2	42.16	4291.04	65.51	-0.0001	80
10	3	29.38	2162.5	46.5	-0.0	82
10	4	12.84	559.72	23.66	-0.0	73
25	1	36.08	3740.72	61.16	-0.0072	118
25	2	42.14	4290.91	65.51	-0.0	94
25	3	29.38	2162.47	46.5	-0.0	80
25	4	12.84	559.73	23.66	-0.0	67
50	1	36.05	3738.43	61.14	-0.0066	77
50	2	42.17	4291.2	65.51	-0.0001	97
50	3	29.38	2162.62	46.5	-0.0001	103
50	4	12.84	559.94	23.66	-0.0004	75
100	1	35.99	3733.48	61.1	-0.0053	111
100	2	42.12	4290.78	65.5	-0.0	81
100	3	29.37	2164.63	46.53	-0.001	68
100	4	12.84	559.72	23.66	-0.0	71

Table B.44: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.24	6169.24	78.54	-0.0114	95
2	2	53.54	6324.5	79.53	-0.0002	69
2	3	34.73	3209.48	56.65	-0.0001	89
2	4	14.88	871.43	29.52	-0.0007	87
5	1	45.25	6170.67	78.55	-0.0117	139
5	2	53.53	6324.22	79.52	-0.0002	74
5	3	34.7	3209.35	56.65	-0.0	68
5	4	14.88	871.39	29.52	-0.0007	66
10	1	45.1	6151.43	78.43	-0.0085	100
10	2	53.45	6323.19	79.52	-0.0	70
10	3	34.64	3209.36	56.65	-0.0	94
10	4	14.85	870.95	29.51	-0.0002	79
25	1	45.12	6153.7	78.45	-0.0089	132
25	2	53.43	6323.21	79.52	-0.0	85
25	3	34.65	3209.33	56.65	-0.0	111
25	4	14.85	870.93	29.51	-0.0002	76
50	1	45.11	6152.98	78.44	-0.0088	127
50	2	53.51	6323.86	79.52	-0.0001	86
50	3	34.7	3209.34	56.65	-0.0	73
50	4	14.86	871.06	29.51	-0.0003	97
100	1	45.0	6138.71	78.35	-0.0064	79
100	2	53.48	6323.42	79.52	-0.0	67
100	3	34.69	3209.31	56.65	-0.0	83
100	4	14.85	870.98	29.51	-0.0002	66

Table B.45: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.85	12397.19	111.34	-0.0349	71
2	2	82.12	13649.85	116.83	-0.0238	78
2	3	53.05	6471.64	80.45	-0.0273	78
2	4	23.23	1767.75	42.04	-0.0548	66
5	1	68.8	12185.98	110.39	-0.0173	79
5	2	79.04	12849.32	113.35	0.0363	74
5	3	52.68	6476.31	80.48	-0.028	131
5	4	23.31	1848.73	43.0	-0.1031	84
10	1	69.25	12252.97	110.69	-0.0229	69
10	2	81.78	13334.69	115.48	-0.0001	68
10	3	51.68	6300.02	79.37	-0.0001	72
10	4	21.65	1676.91	40.95	-0.0006	69
25	1	68.69	12162.79	110.29	-0.0154	77
25	2	81.9	13337.01	115.49	-0.0003	75
25	3	51.84	6299.66	79.37	-0.0	72
25	4	21.82	1679.68	40.98	-0.0023	77
50	1	68.47	12119.13	110.09	-0.0117	66
50	2	81.69	13333.57	115.47	-0.0001	66
50	3	51.92	6299.84	79.37	-0.0	66
50	4	21.7	1677.54	40.96	-0.001	87
100	1	68.18	12030.49	109.68	-0.0043	66
100	2	81.02	13342.68	115.51	-0.0007	169
100	3	51.08	6310.9	79.44	-0.0018	86
100	4	21.43	1676.45	40.94	-0.0003	87

Table B.46: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	104.75	27113.3	164.66	-0.0643	117
2	2	107.86	19374.55	139.19	0.0335	79
2	3	84.99	14153.12	118.97	-0.0834	81
2	4	36.79	3740.57	61.16	-0.1394	66
5	1	104.3	28091.3	167.6	-0.1026	92
5	2	107.87	19736.86	140.49	0.0154	75
5	3	83.68	13998.2	118.31	-0.0715	72
5	4	34.51	3514.22	59.28	-0.0705	90
10	1	102.71	26218.91	161.92	-0.0292	69
10	2	112.18	21581.66	146.91	-0.0767	66
10	3	87.41	14721.55	121.33	-0.1269	66
10	4	37.38	4121.89	64.2	-0.2556	169
25	1	102.68	26061.78	161.44	-0.023	80
25	2	111.2	20073.06	141.68	-0.0014	70
25	3	85.24	14668.75	121.11	-0.1228	66
25	4	40.95	4033.44	63.51	-0.2286	68
50	1	102.26	25818.4	160.68	-0.0134	69
50	2	109.41	20112.09	141.82	-0.0033	66
50	3	79.21	13127.12	114.57	-0.0048	79
50	4	29.7	3284.41	57.31	-0.0005	70
100	1	108.31	27632.92	166.23	-0.0847	77
100	2	112.19	20364.18	142.7	-0.0159	83
100	3	109.54	23653.22	153.8	-0.8105	75
100	4	80.08	10834.03	104.09	-2.3001	75

Table B.47: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	171.6	60175.91	245.31	-0.0753	68
2	2	134.53	30922.24	175.85	0.1072	70
2	3	134.35	33700.27	183.58	-0.1918	67
2	4	73.17	10657.6	103.24	-0.3965	66
5	1	178.58	61971.74	248.94	-0.1074	79
5	2	136.26	32660.06	180.72	0.057	67
5	3	124.65	32471.91	180.2	-0.1484	66
5	4	73.41	11878.57	108.99	-0.5565	71
10	1	172.96	59270.26	243.45	-0.0591	68
10	2	142.89	33311.52	182.51	0.0382	66
10	3	117.94	31168.2	176.55	-0.1023	66
10	4	88.06	15346.81	123.88	-1.0109	71
25	1	175.66	67307.29	259.44	-0.2027	165
25	2	168.01	50536.2	224.8	-0.4591	84
25	3	150.75	54313.88	233.05	-0.9208	66
25	4	138.68	31120.48	176.41	-3.0778	76
50	1	183.6	66452.22	257.78	-0.1874	76
50	2	194.09	63128.86	251.25	-0.8227	96
50	3	190.78	68950.12	262.58	-1.4384	83
50	4	82.94	12172.82	110.33	-0.595	70
100	1	184.61	59250.99	243.42	-0.0588	77
100	2	149.32	35346.16	188.01	-0.0205	66
100	3	113.59	31333.08	177.01	-0.1081	67
100	4	98.13	17528.85	132.4	-1.2968	68

Table B.48: Result metrics for the $Comp_{td}$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3 Volume Models Results

B.3.1 The Non-Time Differenced Original Tweet Volume Model $VolumeOrg_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	75.91	8772.55	93.66	0.9834	86
2	2	182.98	38319.1	195.75	0.7648	73
2	3	119.08	16269.71	127.55	0.8332	67
2	4	35.27	1694.68	41.17	0.8727	79
5	1	125.89	25404.27	159.39	0.952	73
5	2	141.18	23245.96	152.47	0.8573	74
5	3	83.51	9132.99	95.57	0.9064	79
5	4	26.43	1018.25	31.91	0.9235	99
10	1	146.06	29418.07	171.52	0.9444	74
10	2	81.23	8963.18	94.67	0.945	87
10	3	77.48	9120.07	95.5	0.9065	71
10	4	36.16	1593.83	39.92	0.8803	67
25	1	147.65	30811.05	175.53	0.9417	69
25	2	92.82	11107.29	105.39	0.9318	79
25	3	60.03	5563.24	74.59	0.943	71
25	4	36.03	1893.63	43.52	0.8578	68
50	1	121.46	26456.98	162.66	0.95	66
50	2	82.01	13007.87	114.05	0.9202	74
50	3	68.32	10278.28	101.38	0.8946	66
50	4	47.26	3641.82	60.35	0.7265	71
100	1	157.54	38873.92	197.16	0.9265	66
100	2	131.84	27009.2	164.34	0.8342	67
100	3	68.53	11707.63	108.2	0.88	77
100	4	50.88	3629.32	60.24	0.7274	87

Table B.49: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	57.45	5853.16	76.51	0.9889	74
2	2	28.59	1680.29	40.99	0.9896	76
2	3	27.32	1679.58	40.98	0.9828	82
2	4	13.25	449.94	21.21	0.9661	77
5	1	49.82	4174.47	64.61	0.9921	90
5	2	34.32	2352.93	48.51	0.9855	71
5	3	27.99	1575.2	39.69	0.9838	90
5	4	12.63	533.37	23.09	0.9598	80
10	1	59.63	6015.96	77.56	0.9886	78
10	2	33.75	2312.36	48.09	0.9857	83
10	3	27.44	1693.51	41.15	0.9826	71
10	4	12.24	706.67	26.58	0.9468	101
25	1	71.38	8699.71	93.27	0.9835	68
25	2	43.4	3358.84	57.96	0.9793	68
25	3	29.19	1955.92	44.23	0.9799	70
25	4	13.42	879.09	29.65	0.9338	84
50	1	73.04	10736.95	103.62	0.9797	78
50	2	38.13	3034.66	55.09	0.9813	71
50	3	30.29	2279.93	47.75	0.9766	91
50	4	15.12	866.05	29.43	0.9348	71
100	1	72.44	8843.32	94.04	0.9833	69
100	2	83.06	10388.72	101.93	0.936	67
100	3	45.32	3493.51	59.11	0.9641	68
100	4	43.81	4729.3	68.77	0.6439	87

Table B.50: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	62.6	7401.92	86.03	0.986	70
2	2	68.09	6141.46	78.37	0.9619	66
2	3	24.03	1471.64	38.36	0.9848	68
2	4	15.45	668.07	25.85	0.9495	75
5	1	47.77	4266.14	65.32	0.9919	70
5	2	40.45	3253.85	57.04	0.9798	76
5	3	23.64	1603.29	40.04	0.9835	76
5	4	21.74	1276.16	35.72	0.9035	66
10	1	66.39	7219.99	84.97	0.9863	107
10	2	50.11	4433.17	66.58	0.9725	67
10	3	24.72	1755.83	41.9	0.9819	66
10	4	17.72	1796.86	42.39	0.8641	85
25	1	51.09	4987.27	70.62	0.9906	87
25	2	45.07	3782.26	61.5	0.9765	101
25	3	28.67	1902.72	43.62	0.9804	79
25	4	25.0	3443.49	58.68	0.7397	91
50	1	62.1	7237.85	85.08	0.9863	76
50	2	46.79	4369.98	66.11	0.9729	75
50	3	28.38	2069.36	45.49	0.9787	86
50	4	41.64	7894.93	88.85	0.4031	73
100	1	247.73	203394.69	450.99	0.6153	66
100	2	82.22	9570.24	97.83	0.9406	69
100	3	38.91	3560.89	59.67	0.9633	92
100	4	78.07	41624.98	204.02	-2.1471	78

Table B.51: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	83.34	13269.15	115.19	0.9749	68
2	2	54.39	5755.93	75.87	0.964	74
2	3	32.62	2627.37	51.26	0.9726	71
2	4	16.9	2298.22	47.94	0.825	103
5	1	79.76	11544.8	107.45	0.9781	81
5	2	48.81	5122.96	71.57	0.968	66
5	3	32.14	2655.13	51.53	0.9723	76
5	4	22.97	3564.47	59.7	0.7285	77
10	1	87.5	13752.02	117.27	0.9739	82
10	2	49.32	5236.07	72.36	0.9673	90
10	3	32.47	2667.63	51.65	0.9722	67
10	4	28.04	7078.93	84.14	0.4608	111
25	1	57.54	6459.67	80.37	0.9878	79
25	2	48.27	5249.97	72.46	0.9672	66
25	3	38.11	3294.45	57.4	0.9657	68
25	4	54.21	31938.97	178.71	-1.4326	81
50	1	118.75	22661.82	150.54	0.9571	67
50	2	55.09	5989.26	77.39	0.9625	70
50	3	39.6	3594.69	59.96	0.9625	67
50	4	76.31	25844.1	160.76	-0.9684	104
100	1	192.29	88090.89	296.8	0.8331	71
100	2	70.74	8621.2	92.85	0.9461	84
100	3	49.59	4797.72	69.27	0.95	71
100	4	133.99	55198.85	234.94	-3.2041	78

Table B.52: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	88.35	13804.69	117.49	0.9736	80
2	2	57.95	7408.99	86.08	0.9535	88
2	3	40.46	4064.97	63.76	0.9581	68
2	4	23.78	3177.3	56.37	0.7573	69
5	1	79.5	11499.24	107.23	0.978	77
5	2	58.66	7114.34	84.35	0.9554	91
5	3	38.23	3638.98	60.32	0.9625	83
5	4	22.91	1911.16	43.72	0.854	85
10	1	77.38	10647.78	103.19	0.9797	66
10	2	57.95	7182.32	84.75	0.9549	96
10	3	40.47	3779.54	61.48	0.961	86
10	4	42.4	16151.57	127.09	-0.2336	110
25	1	68.42	9334.52	96.62	0.9822	70
25	2	60.9	7569.84	87.0	0.9525	80
25	3	39.12	4107.88	64.09	0.9577	73
25	4	84.99	45443.46	213.17	-2.4707	75
50	1	125.93	22084.32	148.61	0.9578	66
50	2	67.24	8259.67	90.88	0.9482	77
50	3	46.18	4832.42	69.52	0.9502	83
50	4	194.8	169849.69	412.13	-11.9722	73
100	1	272.49	98990.05	314.63	0.8109	68
100	2	88.92	14196.38	119.15	0.9109	67
100	3	52.16	5937.29	77.05	0.9388	84
100	4	248.61	193975.31	440.43	-13.8147	98

Table B.53: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	113.05	21199.28	145.6	0.9596	66
2	2	86.08	13341.7	115.51	0.915	77
2	3	56.13	6915.12	83.16	0.9268	73
2	4	38.84	14464.98	120.27	-0.1227	79
5	1	95.31	16180.7	127.2	0.9692	85
5	2	85.15	13399.44	115.76	0.9146	92
5	3	60.02	7022.88	83.8	0.9257	68
5	4	80.03	81232.48	285.01	-5.3047	69
10	1	80.81	14232.51	119.3	0.9729	82
10	2	87.11	13532.8	116.33	0.9138	67
10	3	56.17	7174.59	84.7	0.9241	68
10	4	117.48	95856.81	309.61	-6.4398	71
25	1	119.87	21371.01	146.19	0.9593	82
25	2	86.69	13418.66	115.84	0.9145	67
25	3	61.64	8083.78	89.91	0.9144	69
25	4	244.85	294180.12	542.38	-21.8323	83
50	1	135.67	25810.3	160.66	0.9509	66
50	2	105.7	17916.2	133.85	0.8858	66
50	3	69.64	9623.04	98.1	0.8981	91
50	4	590.85	1040531.2	1020.06	-79.7592	85
100	1	287.18	124917.15	353.44	0.7622	78
100	2	105.97	18699.8	136.75	0.8808	102
100	3	83.86	13478.49	116.1	0.8573	71
100	4	443.22	455015.1	674.55	-34.3153	80

Table B.54: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	146.88	37238.86	192.97	0.9293	67
2	2	106.91	18243.8	135.07	0.8793	78
2	3	84.74	13271.77	115.2	0.848	80
2	4	82.71	83214.64	288.47	-5.2662	73
5	1	123.67	30175.53	173.71	0.9427	75
5	2	114.04	19174.25	138.47	0.8732	83
5	3	84.83	13068.64	114.32	0.8503	70
5	4	153.61	170717.02	413.18	-11.8552	66
10	1	128.15	28057.59	167.5	0.9467	66
10	2	118.94	21044.13	145.07	0.8608	111
10	3	92.08	14908.66	122.1	0.8293	77
10	4	294.19	480030.88	692.84	-35.147	78
25	1	152.02	35537.91	188.52	0.9325	74
25	2	120.74	21359.76	146.15	0.8587	74
25	3	95.97	16909.32	130.04	0.8064	71
25	4	558.82	791502.4	889.66	-58.6013	75
50	1	252.41	93455.65	305.71	0.8225	68
50	2	143.52	39919.6	199.8	0.7359	82
50	3	97.35	17461.84	132.14	0.8	80
50	4	804.7	1385771.4	1177.19	-103.3506	78
100	1	300.6	127951.53	357.7	0.757	66
100	2	355.91	201395.45	448.77	-0.3323	67
100	3	133.58	28516.26	168.87	0.6734	67
100	4	981.97	3877172.8	1969.05	-290.9567	81

Table B.55: Result metrics for the $VolumeOrg_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	182.9	60623.63	246.22	0.8809	66
2	2	134.04	31148.41	176.49	0.7785	74
2	3	108.7	24889.16	157.76	0.6991	70
2	4	152.57	247198.38	497.19	-16.4132	83
5	1	175.43	56145.18	236.95	0.8897	66
5	2	231.84	72311.31	268.91	0.4859	66
5	3	111.8	25411.61	159.41	0.6928	71
5	4	230.05	231710.81	481.36	-15.3222	73
10	1	178.94	56265.08	237.2	0.8894	66
10	2	280.71	115634.44	340.05	0.1779	76
10	3	128.24	29645.0	172.18	0.6416	74
10	4	681.83	1465308.6	1210.5	-102.2194	69
25	1	259.72	92573.99	304.26	0.8181	66
25	2	383.89	171923.27	414.64	-0.2223	66
25	3	109.15	31660.03	177.93	0.6172	66
25	4	903.31	1703651.4	1305.24	-119.0087	70
50	1	225.45	98325.3	313.57	0.8068	75
50	2	553.64	452244.1	672.49	-2.2154	68
50	3	197.6	66330.7	257.55	0.1981	80
50	4	505.07	544805.44	738.11	-37.3772	66
100	1	752.35	964484.75	982.08	-0.8955	65
100	2	561.06	400690.2	633.0	-1.8488	89
100	3	214.71	65606.86	256.14	0.2068	78
100	4	301.83	154284.06	392.79	-9.8681	77

Table B.56: Result metrics for the $VolumeOrg_c$ trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3.2 The Time Differenced Original Tweet Volume Model $VolumeOrg_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.8	17.26	-0.0003	196
2	2	11.83	363.37	19.06	-0.0017	188
2	3	8.46	215.25	14.67	-0.002	112
2	4	4.61	92.06	9.6	-0.0059	121
5	1	10.48	297.8	17.26	-0.0003	305
5	2	11.83	363.41	19.06	-0.0019	187
5	3	8.45	215.22	14.67	-0.0019	203
5	4	4.74	93.28	9.66	-0.0192	81
10	1	10.48	297.82	17.26	-0.0004	376
10	2	11.84	363.66	19.07	-0.0025	216
10	3	8.45	215.14	14.67	-0.0015	107
10	4	4.65	92.37	9.61	-0.0092	127
25	1	10.48	297.76	17.26	-0.0002	249
25	2	11.83	363.1	19.06	-0.001	246
25	3	8.45	215.16	14.67	-0.0016	71
25	4	4.66	92.44	9.61	-0.01	70
50	1	10.48	297.79	17.26	-0.0003	186
50	2	11.83	363.16	19.06	-0.0012	130
50	3	8.45	214.99	14.66	-0.0008	274
50	4	4.56	91.64	9.57	-0.0012	82
100	1	10.48	297.83	17.26	-0.0004	299
100	2	11.83	363.12	19.06	-0.001	240
100	3	8.45	214.96	14.66	-0.0007	153
100	4	4.57	91.7	9.58	-0.0018	70

Table B.57: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.55	909.41	30.16	-0.0048	70
2	2	19.84	1010.93	31.8	-0.001	120
2	3	14.79	743.21	27.26	-0.0068	71
2	4	7.36	263.36	16.23	-0.0003	100
5	1	17.47	906.06	30.1	-0.0011	68
5	2	19.84	1011.08	31.8	-0.0012	71
5	3	14.98	749.23	27.37	-0.015	117
5	4	7.39	263.33	16.23	-0.0001	77
10	1	17.49	907.0	30.12	-0.0021	76
10	2	19.83	1010.8	31.79	-0.0009	83
10	3	14.84	744.77	27.29	-0.009	74
10	4	7.37	263.29	16.23	-0.0	77
25	1	17.51	907.79	30.13	-0.003	99
25	2	19.83	1010.66	31.79	-0.0007	76
25	3	14.85	745.23	27.3	-0.0096	152
25	4	7.38	263.29	16.23	-0.0	81
50	1	17.48	906.29	30.1	-0.0014	86
50	2	19.84	1010.39	31.79	-0.0005	66
50	3	14.84	745.0	27.29	-0.0093	66
50	4	7.38	263.29	16.23	-0.0	73
100	1	17.53	908.63	30.14	-0.0039	68
100	2	19.84	1010.42	31.79	-0.0005	82
100	3	14.79	743.33	27.26	-0.007	75
100	4	7.36	263.35	16.23	-0.0002	71

Table B.58: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.01	1815.27	42.61	-0.0031	71
2	2	28.65	2064.56	45.44	-0.0003	80
2	3	20.07	1175.57	34.29	-0.0053	89
2	4	9.87	466.45	21.6	-0.0001	71
5	1	25.08	1820.44	42.67	-0.0059	77
5	2	28.59	2064.83	45.44	-0.0004	66
5	3	20.05	1174.03	34.26	-0.0039	103
5	4	9.88	466.58	21.6	-0.0003	94
10	1	25.06	1819.05	42.65	-0.0052	78
10	2	28.69	2065.84	45.45	-0.0009	74
10	3	20.06	1174.64	34.27	-0.0045	66
10	4	9.87	466.44	21.6	-0.0	67
25	1	25.08	1821.04	42.67	-0.0063	82
25	2	28.62	2064.03	45.43	-0.0	75
25	3	20.06	1174.79	34.28	-0.0046	83
25	4	9.88	466.52	21.6	-0.0002	69
50	1	25.1	1821.86	42.68	-0.0067	71
50	2	28.66	2064.95	45.44	-0.0005	66
50	3	20.03	1172.16	34.24	-0.0024	73
50	4	9.87	466.43	21.6	-0.0	98
100	1	25.09	1821.17	42.68	-0.0063	66
100	2	28.67	2064.99	45.44	-0.0005	70
100	3	20.03	1172.28	34.24	-0.0025	77
100	4	9.87	466.42	21.6	-0.0	74

Table B.59: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.07	3739.66	61.15	-0.0069	127
2	2	42.16	4291.12	65.51	-0.0001	77
2	3	29.37	2165.15	46.53	-0.0013	81
2	4	12.84	559.72	23.66	-0.0	78
5	1	36.07	3739.72	61.15	-0.0069	143
5	2	42.15	4290.97	65.51	-0.0	76
5	3	29.38	2162.64	46.5	-0.0001	81
5	4	12.86	560.43	23.67	-0.0013	95
10	1	36.08	3740.6	61.16	-0.0072	124
10	2	42.15	4291.02	65.51	-0.0001	95
10	3	29.38	2162.47	46.5	-0.0	74
10	4	12.84	559.72	23.66	-0.0	84
25	1	36.08	3740.73	61.16	-0.0072	119
25	2	42.14	4290.92	65.51	-0.0	85
25	3	29.46	2178.31	46.67	-0.0073	69
25	4	12.84	559.73	23.66	-0.0	80
50	1	36.05	3738.47	61.14	-0.0066	86
50	2	42.12	4290.79	65.5	-0.0	86
50	3	29.38	2162.61	46.5	-0.0001	107
50	4	12.85	560.11	23.67	-0.0007	75
100	1	35.99	3733.68	61.1	-0.0053	116
100	2	42.12	4290.77	65.5	-0.0	78
100	3	29.38	2162.5	46.5	-0.0	84
100	4	12.84	559.72	23.66	-0.0	67

Table B.60: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.25	6169.4	78.55	-0.0115	109
2	2	53.54	6324.49	79.53	-0.0002	78
2	3	34.73	3209.49	56.65	-0.0001	97
2	4	14.88	871.45	29.52	-0.0008	72
5	1	45.24	6168.92	78.54	-0.0114	80
5	2	53.53	6324.28	79.53	-0.0002	69
5	3	34.68	3209.31	56.65	-0.0	105
5	4	14.87	871.19	29.52	-0.0005	93
10	1	45.1	6151.49	78.43	-0.0085	91
10	2	53.45	6323.2	79.52	-0.0	68
10	3	34.64	3209.35	56.65	-0.0	78
10	4	14.85	870.95	29.51	-0.0002	76
25	1	45.12	6153.44	78.44	-0.0088	66
25	2	53.44	6323.19	79.52	-0.0	121
25	3	34.65	3209.34	56.65	-0.0	67
25	4	14.85	870.92	29.51	-0.0002	66
50	1	45.11	6152.85	78.44	-0.0087	88
50	2	53.51	6323.82	79.52	-0.0001	66
50	3	34.7	3209.34	56.65	-0.0	78
50	4	14.86	871.07	29.51	-0.0003	66
100	1	45.0	6138.77	78.35	-0.0064	122
100	2	53.48	6323.38	79.52	-0.0	68
100	3	34.69	3209.31	56.65	-0.0	84
100	4	14.86	871.01	29.51	-0.0003	82

Table B.61: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.34	12267.02	110.76	-0.0241	66
2	2	81.04	13361.77	115.59	-0.0022	78
2	3	52.3	6361.56	79.76	-0.0098	69
2	4	25.59	2593.76	50.93	-0.5477	67
5	1	68.6	12145.46	110.21	-0.0139	68
5	2	79.22	12581.7	112.17	0.0563	69
5	3	52.41	6412.18	80.08	-0.0179	81
5	4	67.16	29082.74	170.54	-16.3536	71
10	1	69.25	12253.03	110.69	-0.0229	74
10	2	81.77	13334.68	115.48	-0.0001	79
10	3	51.68	6300.02	79.37	-0.0001	69
10	4	21.65	1676.9	40.95	-0.0006	81
25	1	68.69	12164.85	110.29	-0.0155	66
25	2	81.9	13336.97	115.49	-0.0003	77
25	3	51.84	6299.66	79.37	-0.0	69
25	4	21.82	1679.67	40.98	-0.0023	75
50	1	68.49	12122.59	110.1	-0.012	67
50	2	81.69	13333.56	115.47	-0.0001	85
50	3	51.92	6299.84	79.37	-0.0	68
50	4	21.7	1677.55	40.96	-0.001	83
100	1	68.18	12031.2	109.69	-0.0044	67
100	2	81.02	13342.63	115.51	-0.0007	78
100	3	51.08	6310.77	79.44	-0.0018	67
100	4	21.43	1676.44	40.94	-0.0003	68

Table B.62: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	101.87	26594.9	163.08	-0.0439	81
2	2	110.83	20139.71	141.91	-0.0047	76
2	3	81.76	13661.9	116.88	-0.0457	66
2	4	36.93	4152.37	64.44	-0.2648	69
5	1	104.63	27614.48	166.18	-0.0839	83
5	2	110.81	20298.23	142.47	-0.0126	79
5	3	81.97	13636.72	116.78	-0.0438	68
5	4	193.25	264172.4	513.98	-79.4692	78
10	1	104.47	25269.48	158.96	0.0081	66
10	2	114.98	21476.5	146.55	-0.0714	81
10	3	82.02	13597.74	116.61	-0.0408	70
10	4	269.84	393074.38	626.96	-118.7339	74
25	1	102.68	26061.4	161.44	-0.023	66
25	2	111.21	20073.88	141.68	-0.0014	66
25	3	79.04	13144.93	114.65	-0.0062	77
25	4	30.34	3288.33	57.34	-0.0017	92
50	1	102.26	25819.26	160.68	-0.0135	69
50	2	109.41	20112.23	141.82	-0.0033	66
50	3	79.21	13127.14	114.57	-0.0048	68
50	4	29.7	3284.41	57.31	-0.0005	66
100	1	108.37	27647.19	166.27	-0.0852	66
100	2	142.83	33543.38	183.15	-0.6734	100
100	3	101.53	20664.85	143.75	-0.5818	79
100	4	303.94	157320.17	396.64	-46.9211	93

Table B.63: Result metrics for the $VolumeOrg_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	174.43	60588.01	246.15	-0.0827	132
2	2	141.67	31952.42	178.75	0.0775	72
2	3	112.92	29184.88	170.84	-0.0321	67
2	4	147.98	95355.46	308.8	-11.4946	102
5	1	173.32	59170.0	243.25	-0.0573	89
5	2	146.35	33977.05	184.33	0.019	66
5	3	115.34	29117.82	170.64	-0.0297	68
5	4	337.13	388952.62	623.66	-49.9652	68
10	1	169.36	59788.7	244.52	-0.0684	82
10	2	152.24	36510.67	191.08	-0.0541	76
10	3	114.86	30859.28	175.67	-0.0913	75
10	4	158.37	67062.51	258.96	-7.7873	77
25	1	172.64	57939.86	240.71	-0.0353	75
25	2	226.91	66837.06	258.53	-0.9297	80
25	3	119.1	31912.1	178.64	-0.1286	68
25	4	849.59	2011851.2	1418.4	-262.6169	67
50	1	234.21	108320.98	329.12	-0.9356	111
50	2	223.26	92033.89	303.37	-1.6572	96
50	3	137.66	42750.34	206.76	-0.5118	67
50	4	646.51	1115964.5	1056.39	-145.2271	70
100	1	185.32	59386.12	243.69	-0.0612	77
100	2	149.95	38297.26	195.7	-0.1057	131
100	3	125.38	33350.7	182.62	-0.1794	66
100	4	173.29	38224.28	195.51	-4.0086	81

Table B.64: Result metrics for the $VolumeOrg_{td}$ trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3.3 The Non-Time Differenced Filtered Tweet Volume Model $VolumeFilt_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	85.89	10961.9	104.7	0.9793	67
2	2	112.98	15245.49	123.47	0.9064	80
2	3	87.64	9378.92	96.84	0.9038	75
2	4	33.97	1404.49	37.48	0.8945	84
5	1	125.92	21258.2	145.8	0.9598	66
5	2	142.97	23738.46	154.07	0.8543	91
5	3	101.44	13864.34	117.75	0.8579	68
5	4	31.6	1232.77	35.11	0.9074	83
10	1	155.84	32280.18	179.67	0.939	82
10	2	112.17	15890.11	126.06	0.9025	78
10	3	93.82	11388.98	106.72	0.8832	70
10	4	49.26	2995.52	54.73	0.775	76
25	1	151.98	33062.6	181.83	0.9375	72
25	2	97.15	12632.54	112.39	0.9225	92
25	3	85.95	16039.52	126.65	0.8356	77
25	4	43.03	2287.79	47.83	0.8282	75
50	1	171.23	41819.08	204.5	0.9209	70
50	2	104.18	18987.58	137.8	0.8834	88
50	3	76.13	14119.18	118.82	0.8552	75
50	4	40.57	2405.84	49.05	0.8193	74
100	1	125.78	27991.9	167.31	0.9471	72
100	2	154.07	39786.42	199.47	0.7558	78
100	3	89.53	13828.7	117.6	0.8582	87
100	4	53.39	3793.5	61.59	0.7151	79

Table B.65: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	83.66	10631.72	103.11	0.9799	77
2	2	28.26	1677.87	40.96	0.9897	74
2	3	34.29	1759.94	41.95	0.9819	67
2	4	13.43	496.67	22.29	0.9626	85
5	1	59.4	5322.39	72.95	0.9899	70
5	2	34.69	2396.04	48.95	0.9852	71
5	3	23.44	1414.32	37.61	0.9855	71
5	4	23.11	1005.02	31.7	0.9243	84
10	1	67.47	8069.47	89.83	0.9847	68
10	2	39.43	2919.47	54.03	0.982	80
10	3	29.56	1918.47	43.8	0.9803	78
10	4	11.99	491.18	22.16	0.963	91
25	1	73.2	10116.62	100.58	0.9809	75
25	2	41.14	3080.42	55.5	0.981	67
25	3	35.86	3218.19	56.73	0.967	76
25	4	13.72	629.79	25.1	0.9526	100
50	1	79.65	9688.04	98.43	0.9817	85
50	2	71.6	7990.95	89.39	0.9507	92
50	3	43.1	3388.87	58.21	0.9652	82
50	4	23.29	1087.02	32.97	0.9181	89
100	1	109.71	15976.68	126.4	0.9698	79
100	2	101.04	14789.71	121.61	0.9088	71
100	3	46.36	4050.45	63.64	0.9584	81
100	4	27.92	1513.23	38.9	0.8861	75

Table B.66: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	65.18	8279.77	90.99	0.9843	66
2	2	39.49	2872.9	53.6	0.9822	73
2	3	26.17	1583.72	39.8	0.9837	68
2	4	17.73	714.68	26.73	0.946	66
5	1	47.18	4673.85	68.37	0.9912	66
5	2	51.98	4331.9	65.82	0.9731	70
5	3	26.39	1765.08	42.01	0.9818	81
5	4	11.56	577.94	24.04	0.9563	89
10	1	50.88	4912.29	70.09	0.9907	87
10	2	39.39	3266.55	57.15	0.9797	95
10	3	32.05	2252.89	47.46	0.9768	78
10	4	21.8	928.05	30.46	0.9298	91
25	1	63.31	7186.86	84.78	0.9864	69
25	2	50.54	4776.18	69.11	0.9704	89
25	3	33.96	2508.01	50.08	0.9741	82
25	4	15.24	872.54	29.54	0.934	76
50	1	78.01	10038.04	100.19	0.981	68
50	2	52.24	5249.61	72.45	0.9674	79
50	3	31.26	2469.73	49.7	0.9745	92
50	4	17.13	834.07	28.88	0.9369	78
100	1	144.89	34759.08	186.44	0.9342	79
100	2	75.99	8979.6	94.76	0.9443	70
100	3	38.49	3384.99	58.18	0.9651	95
100	4	35.24	1994.03	44.65	0.8492	76

Table B.67: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	102.92	20199.22	142.12	0.9617	66
2	2	47.73	5012.13	70.8	0.9687	92
2	3	30.4	2422.46	49.22	0.9747	82
2	4	13.51	588.63	24.26	0.9552	73
5	1	83.54	12949.93	113.8	0.9755	72
5	2	53.87	5589.51	74.76	0.965	80
5	3	32.94	2821.64	53.12	0.9706	97
5	4	14.33	745.53	27.3	0.9432	93
10	1	80.89	11971.87	109.42	0.9773	70
10	2	49.57	5392.15	73.43	0.9663	84
10	3	34.41	2751.71	52.46	0.9713	72
10	4	16.04	718.57	26.81	0.9453	77
25	1	74.13	9790.82	98.95	0.9815	67
25	2	54.92	6055.07	77.81	0.9621	78
25	3	37.06	3297.62	57.42	0.9656	78
25	4	17.56	830.03	28.81	0.9368	70
50	1	126.54	26934.64	164.12	0.949	67
50	2	62.97	7488.04	86.53	0.9532	95
50	3	43.24	4110.86	64.12	0.9571	100
50	4	20.89	1094.26	33.08	0.9167	101
100	1	143.85	31764.06	178.22	0.9398	66
100	2	96.06	14380.27	119.92	0.9101	81
100	3	63.78	7810.84	88.38	0.9186	85
100	4	34.38	2359.07	48.57	0.8203	105

Table B.68: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	85.27	12481.42	111.72	0.9762	77
2	2	56.91	6901.17	83.07	0.9567	66
2	3	36.0	3508.4	59.23	0.9638	87
2	4	16.19	907.79	30.13	0.9307	74
5	1	73.73	10282.55	101.4	0.9804	68
5	2	60.72	7323.35	85.58	0.954	71
5	3	40.11	3817.65	61.79	0.9606	81
5	4	16.26	990.3	31.47	0.9244	82
10	1	79.79	11299.24	106.3	0.9784	79
10	2	59.83	7363.36	85.81	0.9538	83
10	3	38.21	3963.52	62.96	0.9591	75
10	4	16.95	1009.59	31.77	0.9229	66
25	1	79.37	11026.85	105.01	0.9789	69
25	2	63.18	8144.1	90.24	0.9489	69
25	3	46.09	5416.91	73.6	0.9442	82
25	4	24.54	1573.66	39.67	0.8798	80
50	1	134.19	25472.31	159.6	0.9513	68
50	2	74.61	10626.14	103.08	0.9333	79
50	3	54.92	6173.83	78.57	0.9364	74
50	4	24.39	1533.38	39.16	0.8829	68
100	1	204.75	56583.01	237.87	0.8919	69
100	2	97.57	17069.92	130.65	0.8929	67
100	3	56.28	6608.18	81.29	0.9319	71
100	4	45.53	3970.66	63.01	0.6967	66

Table B.69: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	105.12	18085.74	134.48	0.9656	117
2	2	80.02	13111.9	114.51	0.9164	73
2	3	51.17	6607.19	81.28	0.9301	69
2	4	23.44	1709.61	41.35	0.8673	68
5	1	85.79	14474.12	120.31	0.9724	66
5	2	86.41	13909.26	117.94	0.9114	78
5	3	54.78	6931.95	83.26	0.9266	69
5	4	24.0	1908.18	43.68	0.8519	67
10	1	83.03	14957.97	122.3	0.9715	86
10	2	85.44	13961.68	118.16	0.911	69
10	3	59.41	6990.08	83.61	0.926	104
10	4	27.57	2064.17	45.43	0.8398	66
25	1	112.89	19441.22	139.43	0.963	67
25	2	92.71	14839.92	121.82	0.9054	80
25	3	64.71	8555.49	92.5	0.9094	82
25	4	45.47	3506.53	59.22	0.7278	73
50	1	184.96	48038.4	219.18	0.9086	67
50	2	102.66	17902.31	133.8	0.8859	68
50	3	81.65	12114.91	110.07	0.8718	80
50	4	39.96	3168.19	56.29	0.7541	74
100	1	351.27	158254.31	397.81	0.6988	68
100	2	132.58	25786.06	160.58	0.8357	72
100	3	89.74	17058.22	130.61	0.8194	92
100	4	54.9	4988.49	70.63	0.6128	111

Table B.70: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	158.63	43475.81	208.51	0.9174	73
2	2	107.22	18838.9	137.25	0.8754	69
2	3	82.43	13057.98	114.27	0.8505	71
2	4	35.19	3566.74	59.72	0.7314	72
5	1	120.21	28882.28	169.95	0.9451	66
5	2	112.96	19635.07	140.13	0.8701	66
5	3	78.87	12287.17	110.85	0.8593	72
5	4	38.01	3860.94	62.14	0.7093	77
10	1	131.16	30511.0	174.67	0.942	80
10	2	112.92	20103.6	141.79	0.867	70
10	3	86.84	13903.8	117.91	0.8408	68
10	4	56.29	4944.07	70.31	0.6277	86
25	1	156.25	36265.2	190.43	0.9311	66
25	2	134.01	28838.69	169.82	0.8092	67
25	3	99.88	18585.01	136.33	0.7872	78
25	4	59.16	5892.34	76.76	0.5563	89
50	1	420.66	409114.6	639.62	0.2229	66
50	2	155.58	36495.34	191.04	0.7586	74
50	3	115.53	24636.63	156.96	0.7179	72
50	4	70.58	8188.04	90.49	0.3834	66
100	1	239.71	84905.95	291.39	0.8387	81
100	2	228.06	77675.16	278.7	0.4861	128
100	3	127.86	31274.06	176.84	0.6419	66
100	4	104.31	16616.74	128.91	-0.2513	69

Table B.71: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	201.65	68596.49	261.91	0.8652	66
2	2	136.97	33505.5	183.05	0.7618	71
2	3	107.4	24672.18	157.07	0.7017	80
2	4	52.92	6513.8	80.71	0.5412	72
5	1	178.45	58490.22	241.85	0.8851	66
5	2	239.75	85373.41	292.19	0.393	70
5	3	110.33	25443.0	159.51	0.6924	78
5	4	48.62	5443.18	73.78	0.6166	67
10	1	169.34	57698.34	240.2	0.8866	66
10	2	410.71	230005.4	479.59	-0.6353	69
10	3	111.65	28075.92	167.56	0.6606	77
10	4	65.05	7168.83	84.67	0.495	96
25	1	220.65	77534.53	278.45	0.8476	91
25	2	332.12	148859.33	385.82	-0.0584	66
25	3	119.36	32522.28	180.34	0.6068	74
25	4	86.3	13609.14	116.66	0.0413	67
50	1	269.75	133601.67	365.52	0.7374	66
50	2	307.93	136377.62	369.29	0.0304	86
50	3	151.16	40474.79	201.18	0.5107	72
50	4	84.6	13081.18	114.37	0.0785	66
100	1	751.88	963129.44	981.39	-0.8928	65
100	2	283.51	132681.47	364.25	0.0567	90
100	3	188.59	55642.59	235.89	0.3273	98
100	4	130.41	22650.41	150.5	-0.5955	69

Table B.72: Result metrics for the $VolumeFilt_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3.4 The Time Differenced Filtered Tweet Volume Model $VolumeFilt_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.79	17.26	-0.0003	173
2	2	11.83	363.36	19.06	-0.0017	176
2	3	8.45	215.15	14.67	-0.0015	193
2	4	4.65	92.4	9.61	-0.0095	76
5	1	10.48	297.84	17.26	-0.0005	281
5	2	11.83	363.37	19.06	-0.0017	251
5	3	8.45	215.1	14.67	-0.0013	180
5	4	4.65	92.4	9.61	-0.0095	87
10	1	10.48	297.81	17.26	-0.0004	316
10	2	11.84	363.64	19.07	-0.0025	218
10	3	8.45	214.98	14.66	-0.0008	227
10	4	4.55	91.56	9.57	-0.0003	99
25	1	10.48	297.84	17.26	-0.0005	406
25	2	11.84	363.78	19.07	-0.0029	190
25	3	8.46	215.24	14.67	-0.002	87
25	4	4.61	92.01	9.59	-0.0053	226
50	1	10.48	297.77	17.26	-0.0002	232
50	2	11.83	363.23	19.06	-0.0014	146
50	3	8.45	215.13	14.67	-0.0015	241
50	4	4.56	91.62	9.57	-0.001	144
100	1	10.48	297.79	17.26	-0.0003	387
100	2	11.84	363.63	19.07	-0.0025	232
100	3	8.46	215.27	14.67	-0.0021	82
100	4	4.67	92.59	9.62	-0.0116	195

Table B.73: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.51	907.87	30.13	-0.0031	95
2	2	19.83	1010.86	31.79	-0.0009	90
2	3	14.85	745.15	27.3	-0.0095	114
2	4	7.36	263.32	16.23	-0.0001	68
5	1	17.49	906.87	30.11	-0.002	70
5	2	19.84	1010.59	31.79	-0.0007	66
5	3	14.83	744.56	27.29	-0.0087	154
5	4	7.38	263.29	16.23	-0.0	67
10	1	17.53	908.34	30.14	-0.0036	71
10	2	19.84	1010.98	31.8	-0.001	71
10	3	14.83	744.46	27.28	-0.0085	88
10	4	7.38	263.3	16.23	-0.0	78
25	1	17.56	909.52	30.16	-0.0049	71
25	2	19.84	1010.93	31.8	-0.001	66
25	3	14.79	743.37	27.26	-0.0071	69
25	4	7.38	263.29	16.23	-0.0	78
50	1	17.48	906.54	30.11	-0.0016	91
50	2	19.84	1010.34	31.79	-0.0004	72
50	3	14.85	745.14	27.3	-0.0095	67
50	4	7.37	263.31	16.23	-0.0001	67
100	1	17.53	908.43	30.14	-0.0037	80
100	2	19.84	1010.4	31.79	-0.0005	70
100	3	14.81	744.02	27.28	-0.0079	91
100	4	7.37	263.31	16.23	-0.0001	70

Table B.74: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.03	1816.89	42.63	-0.004	83
2	2	28.65	2064.59	45.44	-0.0003	96
2	3	20.05	1173.68	34.26	-0.0036	82
2	4	9.87	466.43	21.6	-0.0	70
5	1	25.01	1815.45	42.61	-0.0032	67
5	2	28.64	2064.27	45.43	-0.0002	66
5	3	20.04	1173.52	34.26	-0.0035	88
5	4	9.87	466.42	21.6	-0.0	69
10	1	25.07	1819.7	42.66	-0.0055	79
10	2	28.69	2065.81	45.45	-0.0009	108
10	3	20.06	1174.61	34.27	-0.0045	85
10	4	9.87	466.45	21.6	-0.0001	94
25	1	25.09	1821.3	42.68	-0.0064	84
25	2	28.61	2063.95	45.43	-0.0	76
25	3	20.07	1175.74	34.29	-0.0054	71
25	4	9.88	466.51	21.6	-0.0002	79
50	1	25.07	1820.02	42.66	-0.0057	71
50	2	28.67	2065.01	45.44	-0.0005	73
50	3	20.03	1172.06	34.24	-0.0023	69
50	4	9.87	466.42	21.6	-0.0	79
100	1	25.05	1818.3	42.64	-0.0047	66
100	2	28.67	2065.0	45.44	-0.0005	79
100	3	20.04	1172.85	34.25	-0.0029	66
100	4	9.87	466.43	21.6	-0.0	66

Table B.75: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.07	3739.51	61.15	-0.0069	119
2	2	42.16	4291.15	65.51	-0.0001	82
2	3	29.38	2162.44	46.5	-0.0	82
2	4	12.84	559.72	23.66	-0.0	69
5	1	36.09	3741.18	61.17	-0.0073	66
5	2	42.18	4291.32	65.51	-0.0001	94
5	3	29.38	2162.46	46.5	-0.0	73
5	4	12.84	559.78	23.66	-0.0001	85
10	1	36.08	3740.61	61.16	-0.0072	116
10	2	42.16	4291.04	65.51	-0.0001	104
10	3	29.38	2162.46	46.5	-0.0	66
10	4	12.84	559.72	23.66	-0.0	67
25	1	36.08	3740.67	61.16	-0.0072	105
25	2	42.15	4290.93	65.51	-0.0	68
25	3	29.38	2162.49	46.5	-0.0	76
25	4	12.84	559.74	23.66	-0.0	69
50	1	36.06	3738.7	61.14	-0.0067	107
50	2	42.16	4291.06	65.51	-0.0001	85
50	3	29.38	2162.5	46.5	-0.0	81
50	4	12.84	559.8	23.66	-0.0002	76
100	1	35.99	3733.69	61.1	-0.0053	142
100	2	42.12	4290.77	65.5	-0.0	67
100	3	29.38	2162.51	46.5	-0.0	69
100	4	12.84	559.72	23.66	-0.0	67

Table B.76: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.25	6169.68	78.55	-0.0115	74
2	2	53.55	6324.56	79.53	-0.0002	104
2	3	34.73	3209.47	56.65	-0.0001	95
2	4	14.88	871.45	29.52	-0.0008	67
5	1	45.26	6171.03	78.56	-0.0117	147
5	2	53.54	6324.35	79.53	-0.0002	84
5	3	34.71	3209.39	56.65	-0.0	76
5	4	14.88	871.39	29.52	-0.0007	71
10	1	45.1	6151.42	78.43	-0.0085	72
10	2	53.45	6323.19	79.52	-0.0	73
10	3	34.64	3209.35	56.65	-0.0	104
10	4	14.85	870.96	29.51	-0.0002	84
25	1	45.12	6153.45	78.44	-0.0088	69
25	2	53.43	6323.26	79.52	-0.0	105
25	3	34.64	3209.34	56.65	-0.0	107
25	4	14.85	870.94	29.51	-0.0002	124
50	1	45.12	6154.47	78.45	-0.009	68
50	2	53.51	6323.8	79.52	-0.0001	69
50	3	34.7	3209.35	56.65	-0.0	102
50	4	14.86	871.09	29.51	-0.0004	90
100	1	45.0	6138.71	78.35	-0.0064	76
100	2	53.47	6323.33	79.52	-0.0	75
100	3	34.69	3209.31	56.65	-0.0	66
100	4	14.86	871.02	29.51	-0.0003	69

Table B.77: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.33	12265.26	110.75	-0.0239	80
2	2	80.78	13354.8	115.56	-0.0017	67
2	3	51.64	6363.4	79.77	-0.0101	75
2	4	21.81	1687.34	41.08	-0.0068	87
5	1	68.68	12161.68	110.28	-0.0153	66
5	2	78.3	12679.91	112.61	0.049	67
5	3	52.25	6398.71	79.99	-0.0157	72
5	4	22.56	1839.04	42.88	-0.0974	88
10	1	69.25	12252.72	110.69	-0.0229	66
10	2	81.78	13334.7	115.48	-0.0002	66
10	3	51.68	6300.02	79.37	-0.0001	84
10	4	21.65	1676.9	40.95	-0.0006	72
25	1	68.72	12170.69	110.32	-0.016	66
25	2	81.9	13336.99	115.49	-0.0003	73
25	3	51.84	6299.66	79.37	-0.0	70
25	4	21.82	1679.69	40.98	-0.0023	102
50	1	68.44	12111.5	110.05	-0.0111	66
50	2	81.69	13333.54	115.47	-0.0001	73
50	3	51.92	6299.84	79.37	-0.0	66
50	4	21.7	1677.55	40.96	-0.001	77
100	1	68.18	12030.36	109.68	-0.0043	66
100	2	81.02	13342.7	115.51	-0.0008	136
100	3	51.08	6310.84	79.44	-0.0018	66
100	4	21.43	1676.42	40.94	-0.0003	81

Table B.78: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	100.93	26248.04	162.01	-0.0303	72
2	2	110.56	20144.51	141.93	-0.005	94
2	3	80.67	13603.89	116.64	-0.0413	66
2	4	30.42	3412.13	58.41	-0.0394	66
5	1	102.39	27280.47	165.17	-0.0708	88
5	2	111.24	20395.48	142.81	-0.0175	71
5	3	80.95	13429.47	115.89	-0.028	80
5	4	32.93	3545.43	59.54	-0.08	78
10	1	105.2	24740.44	157.29	0.0289	66
10	2	115.67	21743.49	147.46	-0.0847	78
10	3	80.72	13439.79	115.93	-0.0287	67
10	4	34.68	3867.84	62.19	-0.1782	74
25	1	102.68	26061.74	161.44	-0.023	67
25	2	111.22	20074.31	141.68	-0.0015	67
25	3	79.04	13145.03	114.65	-0.0062	71
25	4	30.34	3288.33	57.34	-0.0017	69
50	1	102.26	25817.9	160.68	-0.0134	69
50	2	109.41	20112.04	141.82	-0.0033	66
50	3	79.21	13127.17	114.57	-0.0048	94
50	4	29.7	3284.43	57.31	-0.0005	72
100	1	107.99	27549.92	165.98	-0.0814	66
100	2	134.79	29564.57	171.94	-0.4749	106
100	3	108.07	21824.12	147.73	-0.6705	70
100	4	48.79	5357.24	73.19	-0.6319	83

Table B.79: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	173.74	60093.03	245.14	-0.0738	86
2	2	142.67	32243.57	179.56	0.0691	67
2	3	111.42	29152.78	170.74	-0.031	72
2	4	52.96	7405.61	86.06	0.0296	83
5	1	173.35	59223.34	243.36	-0.0583	66
5	2	151.77	35726.49	189.01	-0.0315	66
5	3	115.94	29261.67	171.06	-0.0348	74
5	4	65.68	9497.1	97.45	-0.2444	90
10	1	174.78	59967.75	244.88	-0.0716	66
10	2	149.99	34781.1	186.5	-0.0042	74
10	3	111.58	28732.7	169.51	-0.0161	70
10	4	46.97	7059.04	84.02	0.075	81
25	1	172.4	57338.52	239.45	-0.0246	68
25	2	259.06	91415.48	302.35	-1.6393	122
25	3	113.46	29764.78	172.52	-0.0526	81
25	4	82.29	12471.13	111.67	-0.6341	66
50	1	172.82	58436.1	241.74	-0.0442	88
50	2	188.49	63033.14	251.06	-0.8199	78
50	3	130.36	42307.88	205.69	-0.4962	68
50	4	62.14	9025.99	95.01	-0.1827	70
100	1	185.4	59396.05	243.71	-0.0614	74
100	2	152.88	38817.66	197.02	-0.1207	72
100	3	146.99	37790.8	194.4	-0.3365	96
100	4	116.27	21162.99	145.48	-1.773	97

Table B.80: Result metrics for the $VolumeFilt_{td}$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3.5 The Non-Time Differenced Sentiment Volume Model $VolumeSent_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	178.41	48533.12	220.3	0.9082	67
2	2	154.76	27541.98	165.96	0.8309	75
2	3	138.23	23472.77	153.21	0.7593	69
2	4	47.52	2787.9	52.8	0.7906	81
5	1	131.04	26076.96	161.48	0.9507	81
5	2	103.52	12597.28	112.24	0.9227	117
5	3	104.47	14259.06	119.41	0.8538	83
5	4	44.42	2246.23	47.39	0.8313	108
10	1	145.01	30133.44	173.59	0.943	67
10	2	120.51	18065.96	134.41	0.8891	82
10	3	118.62	19840.29	140.86	0.7966	88
10	4	31.72	1444.79	38.01	0.8915	112
25	1	136.62	26768.42	163.61	0.9494	66
25	2	130.99	22178.2	148.92	0.8639	71
25	3	109.83	20783.86	144.17	0.7869	100
25	4	81.59	7321.78	85.57	0.4501	66
50	1	147.25	30261.76	173.96	0.9428	110
50	2	158.46	31792.6	178.3	0.8048	70
50	3	85.52	16067.23	126.76	0.8353	93
50	4	43.19	2862.91	53.51	0.785	67
100	1	204.68	69078.22	262.83	0.8694	75
100	2	160.98	33492.24	183.01	0.7944	69
100	3	72.04	9946.03	99.73	0.898	70
100	4	68.16	6708.84	81.91	0.4962	70

Table B.81: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	100.72	15988.88	126.45	0.9698	74
2	2	27.75	1676.73	40.95	0.9897	90
2	3	22.26	1116.01	33.41	0.9885	88
2	4	10.76	383.85	19.59	0.9711	76
5	1	43.38	3310.89	57.54	0.9937	67
5	2	35.63	2616.78	51.15	0.9839	80
5	3	27.14	1595.89	39.95	0.9836	82
5	4	11.88	463.11	21.52	0.9651	98
10	1	58.07	5730.31	75.7	0.9892	67
10	2	45.35	3502.37	59.18	0.9784	76
10	3	31.58	2163.2	46.51	0.9778	84
10	4	21.47	974.96	31.22	0.9266	66
25	1	115.67	20690.82	143.84	0.9609	83
25	2	80.34	9991.43	99.96	0.9384	91
25	3	46.15	4444.31	66.67	0.9544	83
25	4	40.46	2255.95	47.5	0.8301	67
50	1	78.3	9222.67	96.03	0.9826	120
50	2	72.99	8172.03	90.4	0.9496	85
50	3	40.12	3016.64	54.92	0.969	71
50	4	21.1	956.85	30.93	0.9279	81
100	1	157.59	35330.49	187.96	0.9332	88
100	2	148.4	29725.09	172.41	0.8168	71
100	3	55.2	5050.14	71.06	0.9482	69
100	4	30.07	1582.93	39.79	0.8808	66

Table B.82: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	48.22	4865.51	69.75	0.9908	83
2	2	54.83	4650.09	68.19	0.9711	69
2	3	25.88	1652.23	40.65	0.983	83
2	4	19.25	797.09	28.23	0.9397	67
5	1	55.64	6309.92	79.44	0.9881	72
5	2	41.55	3598.15	59.98	0.9777	77
5	3	35.36	2542.43	50.42	0.9738	84
5	4	24.39	1085.26	32.94	0.9179	82
10	1	65.57	8362.02	91.44	0.9842	75
10	2	63.0	6510.24	80.69	0.9596	76
10	3	33.26	2324.01	48.21	0.976	82
10	4	20.35	999.67	31.62	0.9244	72
25	1	78.5	10340.29	101.69	0.9804	105
25	2	105.77	14632.58	120.97	0.9092	82
25	3	35.1	2744.53	52.39	0.9717	82
25	4	21.22	1076.03	32.8	0.9186	72
50	1	68.08	8228.76	90.71	0.9844	81
50	2	72.0	7897.75	88.87	0.951	70
50	3	57.58	5659.08	75.23	0.9417	68
50	4	27.76	1491.25	38.62	0.8873	69
100	1	172.03	47594.86	218.16	0.91	82
100	2	87.66	11609.55	107.75	0.9279	67
100	3	67.86	7809.26	88.37	0.9195	99
100	4	29.45	1647.62	40.59	0.8754	93

Table B.83: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	78.3	10812.09	103.98	0.9795	79
2	2	48.01	5006.47	70.76	0.9687	77
2	3	32.93	2571.08	50.71	0.9732	76
2	4	15.76	703.22	26.52	0.9464	75
5	1	83.72	12967.82	113.88	0.9754	67
5	2	53.38	5967.82	77.25	0.9627	70
5	3	39.11	3301.5	57.46	0.9656	77
5	4	19.23	966.25	31.08	0.9264	71
10	1	113.2	21035.38	145.04	0.9602	74
10	2	58.81	6388.28	79.93	0.96	66
10	3	37.84	3225.05	56.79	0.9664	67
10	4	22.5	1163.75	34.11	0.9114	75
25	1	80.1	11221.17	105.93	0.9787	68
25	2	69.12	8052.07	89.73	0.9496	73
25	3	42.21	4162.29	64.52	0.9566	71
25	4	24.06	1252.17	35.39	0.9046	69
50	1	140.75	32206.03	179.46	0.939	72
50	2	83.99	10862.43	104.22	0.9321	66
50	3	53.72	5693.57	75.46	0.9406	72
50	4	34.68	2074.73	45.55	0.842	89
100	1	173.31	43120.42	207.65	0.9183	93
100	2	95.94	14340.95	119.75	0.9103	84
100	3	67.89	8906.6	94.37	0.9072	68
100	4	40.23	2909.06	53.94	0.7784	76

Table B.84: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	84.29	12868.91	113.44	0.9754	78
2	2	60.25	7648.03	87.45	0.952	73
2	3	36.56	3593.8	59.95	0.963	73
2	4	20.85	1200.76	34.65	0.9083	87
5	1	78.0	10975.16	104.76	0.979	72
5	2	61.08	7593.46	87.14	0.9523	71
5	3	40.05	4111.64	64.12	0.9576	66
5	4	21.83	1331.88	36.49	0.8983	106
10	1	86.4	13466.39	116.04	0.9743	67
10	2	65.18	8080.68	89.89	0.9493	84
10	3	42.88	4249.71	65.19	0.9562	77
10	4	22.04	1223.73	34.98	0.9065	111
25	1	83.89	12494.52	111.78	0.9761	78
25	2	70.16	9348.22	96.69	0.9413	80
25	3	48.67	5452.07	73.84	0.9438	86
25	4	29.29	1792.16	42.33	0.8631	68
50	1	132.92	27177.87	164.86	0.9481	69
50	2	81.44	11680.49	108.08	0.9267	70
50	3	62.98	7580.7	87.07	0.9219	69
50	4	31.03	1831.58	42.8	0.8601	81
100	1	228.08	70853.86	266.18	0.8647	69
100	2	102.75	18062.56	134.4	0.8867	73
100	3	78.7	10961.76	104.7	0.887	73
100	4	57.93	5942.06	77.08	0.5462	84

Table B.85: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	106.64	18761.44	136.97	0.9643	66
2	2	84.15	12853.58	113.37	0.9181	88
2	3	56.25	6885.38	82.98	0.9271	75
2	4	24.31	1764.19	42.0	0.8631	110
5	1	92.88	15223.91	123.39	0.971	97
5	2	90.51	14246.57	119.36	0.9092	80
5	3	60.13	7234.22	85.05	0.9234	69
5	4	26.05	1851.93	43.03	0.8563	71
10	1	93.19	17314.1	131.58	0.967	103
10	2	95.58	14382.08	119.93	0.9083	87
10	3	63.26	7881.81	88.78	0.9166	66
10	4	32.01	2328.15	48.25	0.8193	102
25	1	139.81	28137.45	167.74	0.9464	76
25	2	97.11	16329.89	127.79	0.8959	78
25	3	73.58	11487.05	107.18	0.8784	85
25	4	41.9	3350.85	57.89	0.7399	86
50	1	218.52	63789.17	252.57	0.8786	67
50	2	111.55	20097.79	141.77	0.8719	93
50	3	86.71	15480.37	124.42	0.8361	71
50	4	56.21	5713.49	75.59	0.5566	85
100	1	259.89	94783.65	307.87	0.8196	66
100	2	146.22	30738.6	175.32	0.8041	79
100	3	108.41	22984.1	151.61	0.7567	82
100	4	79.11	10144.01	100.72	0.2127	86

Table B.86: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	141.79	34640.57	186.12	0.9342	81
2	2	112.02	19073.9	138.11	0.8738	99
2	3	84.04	12642.68	112.44	0.8552	78
2	4	39.56	3776.58	61.45	0.7156	70
5	1	118.12	29707.48	172.36	0.9436	70
5	2	117.01	19701.37	140.36	0.8697	81
5	3	88.06	13822.97	117.57	0.8417	71
5	4	42.03	4025.91	63.45	0.6968	66
10	1	124.75	28038.82	167.45	0.9467	83
10	2	113.19	20983.66	144.86	0.8612	71
10	3	90.73	14450.99	120.21	0.8345	66
10	4	64.55	6437.14	80.23	0.5153	76
25	1	169.08	46990.83	216.77	0.9107	69
25	2	146.75	33979.42	184.34	0.7752	66
25	3	117.04	24353.47	156.06	0.7211	78
25	4	86.27	13307.36	115.36	-0.0021	93
50	1	318.93	134581.97	366.85	0.7444	67
50	2	154.24	37918.06	194.73	0.7492	66
50	3	135.27	29770.36	172.54	0.6591	68
50	4	87.12	13480.34	116.1	-0.0151	73
100	1	236.39	88476.59	297.45	0.8319	66
100	2	236.85	85887.52	293.07	0.4318	71
100	3	134.98	38217.35	195.49	0.5623	85
100	4	130.4	24405.88	156.22	-0.8378	68

Table B.87: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	175.86	57298.88	239.37	0.8874	66
2	2	141.77	33953.38	184.26	0.7586	81
2	3	110.64	25009.6	158.14	0.6976	66
2	4	49.39	6702.82	81.87	0.5278	78
5	1	171.2	57793.76	240.4	0.8864	66
5	2	188.03	57183.67	239.13	0.5934	68
5	3	124.96	31410.19	177.23	0.6202	66
5	4	78.02	9129.69	95.55	0.3569	75
10	1	196.39	64977.36	254.91	0.8723	66
10	2	218.91	72882.34	269.97	0.4818	66
10	3	142.71	35601.18	188.68	0.5696	76
10	4	70.79	7357.8	85.78	0.4817	69
25	1	302.43	120572.71	347.24	0.763	66
25	2	484.6	279335.84	528.52	-0.986	66
25	3	136.36	39543.79	198.86	0.5219	70
25	4	94.51	14631.6	120.96	-0.0307	74
50	1	488.37	433085.75	658.09	0.1489	69
50	2	311.83	141341.38	375.95	-0.0049	101
50	3	186.36	53522.14	231.35	0.3529	96
50	4	104.57	17426.91	132.01	-0.2276	102
100	1	751.69	963019.6	981.34	-0.8926	65
100	2	342.19	178231.69	422.17	-0.2672	88
100	3	213.81	67156.62	259.15	0.1881	85
100	4	106.05	17194.36	131.13	-0.2112	71

Table B.88: Result metrics for the $VolumeSent_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.3.6 The Time Differenced Sentiment volume Model $VolumeSent_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.81	17.26	-0.0004	261
2	2	11.83	363.37	19.06	-0.0017	230
2	3	8.45	215.08	14.67	-0.0012	225
2	4	4.55	91.54	9.57	-0.0002	165
5	1	10.48	297.83	17.26	-0.0004	113
5	2	11.83	363.15	19.06	-0.0011	252
5	3	8.45	214.99	14.66	-0.0008	129
5	4	4.56	91.6	9.57	-0.0008	129
10	1	10.48	297.79	17.26	-0.0003	241
10	2	11.84	363.65	19.07	-0.0025	188
10	3	8.45	215.09	14.67	-0.0013	110
10	4	4.55	91.54	9.57	-0.0001	178
25	1	10.48	297.87	17.26	-0.0006	148
25	2	11.84	363.71	19.07	-0.0027	128
25	3	8.45	214.99	14.66	-0.0008	165
25	4	4.56	91.63	9.57	-0.0011	99
50	1	10.48	297.8	17.26	-0.0004	197
50	2	11.83	363.15	19.06	-0.0011	177
50	3	8.45	214.98	14.66	-0.0008	188
50	4	4.56	91.61	9.57	-0.0009	85
100	1	10.48	297.8	17.26	-0.0003	255
100	2	11.84	363.88	19.08	-0.0032	228
100	3	8.46	215.29	14.67	-0.0022	101
100	4	4.63	92.17	9.6	-0.007	212

Table B.89: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.55	909.21	30.15	-0.0046	68
2	2	19.84	1010.97	31.8	-0.001	72
2	3	14.77	742.75	27.25	-0.0062	89
2	4	7.36	263.35	16.23	-0.0002	75
5	1	17.56	909.83	30.16	-0.0053	81
5	2	19.84	1010.98	31.8	-0.001	73
5	3	14.8	743.66	27.27	-0.0074	79
5	4	7.38	263.3	16.23	-0.0	80
10	1	17.51	907.49	30.12	-0.0027	83
10	2	19.83	1010.74	31.79	-0.0008	69
10	3	14.82	744.17	27.28	-0.0081	84
10	4	7.38	263.29	16.23	-0.0	67
25	1	17.57	910.27	30.17	-0.0058	68
25	2	19.84	1010.92	31.79	-0.001	67
25	3	14.82	744.32	27.28	-0.0084	73
25	4	7.38	263.29	16.23	-0.0	66
50	1	17.5	907.3	30.12	-0.0025	90
50	2	19.84	1010.56	31.79	-0.0006	83
50	3	14.85	745.23	27.3	-0.0096	87
50	4	7.39	263.33	16.23	-0.0002	101
100	1	17.5	907.39	30.12	-0.0026	107
100	2	19.84	1010.43	31.79	-0.0005	73
100	3	14.79	743.32	27.26	-0.007	92
100	4	7.37	263.31	16.23	-0.0001	102

Table B.90: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.13	1824.16	42.71	-0.008	77
2	2	28.65	2064.64	45.44	-0.0003	73
2	3	20.05	1174.16	34.27	-0.0041	85
2	4	9.87	466.43	21.6	-0.0	82
5	1	25.02	1815.78	42.61	-0.0034	73
5	2	28.64	2064.4	45.44	-0.0002	95
5	3	20.08	1176.68	34.3	-0.0062	67
5	4	9.87	466.45	21.6	-0.0001	66
10	1	25.05	1818.4	42.64	-0.0048	77
10	2	28.69	2065.87	45.45	-0.0009	92
10	3	20.06	1174.63	34.27	-0.0045	67
10	4	9.87	466.48	21.6	-0.0001	88
25	1	25.04	1818.12	42.64	-0.0046	68
25	2	28.65	2064.54	45.44	-0.0003	66
25	3	20.04	1173.27	34.25	-0.0033	71
25	4	9.88	466.51	21.6	-0.0002	80
50	1	25.09	1821.1	42.67	-0.0063	71
50	2	28.67	2065.14	45.44	-0.0006	78
50	3	20.04	1173.02	34.25	-0.0031	89
50	4	9.87	466.42	21.6	-0.0	69
100	1	25.05	1818.75	42.65	-0.005	75
100	2	28.66	2064.91	45.44	-0.0005	82
100	3	20.03	1171.92	34.23	-0.0021	78
100	4	9.87	466.42	21.6	-0.0	66

Table B.91: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.08	3740.67	61.16	-0.0072	135
2	2	42.12	4290.77	65.5	-0.0	95
2	3	29.38	2162.44	46.5	-0.0	89
2	4	12.84	559.72	23.66	-0.0	98
5	1	36.09	3741.34	61.17	-0.0074	68
5	2	42.18	4291.32	65.51	-0.0001	108
5	3	29.38	2162.46	46.5	-0.0	66
5	4	12.84	559.72	23.66	-0.0	70
10	1	36.08	3740.57	61.16	-0.0072	119
10	2	42.13	4290.8	65.5	-0.0	119
10	3	29.38	2162.63	46.5	-0.0001	95
10	4	12.84	559.72	23.66	-0.0	72
25	1	36.08	3740.76	61.16	-0.0072	121
25	2	42.14	4290.89	65.5	-0.0	67
25	3	29.38	2162.47	46.5	-0.0	71
25	4	12.84	559.73	23.66	-0.0	82
50	1	36.05	3738.61	61.14	-0.0066	113
50	2	42.16	4291.1	65.51	-0.0001	113
50	3	29.38	2162.47	46.5	-0.0	92
50	4	12.84	559.75	23.66	-0.0001	77
100	1	35.99	3733.68	61.1	-0.0053	119
100	2	42.12	4290.77	65.5	-0.0	69
100	3	29.38	2162.51	46.5	-0.0	69
100	4	12.84	559.74	23.66	-0.0001	66

Table B.92: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.24	6168.26	78.54	-0.0113	90
2	2	53.54	6324.45	79.53	-0.0002	101
2	3	34.73	3209.45	56.65	-0.0	69
2	4	14.88	871.39	29.52	-0.0007	74
5	1	45.26	6170.74	78.55	-0.0117	93
5	2	53.54	6324.47	79.53	-0.0002	78
5	3	34.72	3209.4	56.65	-0.0	67
5	4	14.88	871.41	29.52	-0.0007	70
10	1	45.1	6151.45	78.43	-0.0085	69
10	2	53.46	6323.2	79.52	-0.0	68
10	3	34.64	3209.36	56.65	-0.0	90
10	4	14.85	870.96	29.51	-0.0002	71
25	1	45.11	6153.23	78.44	-0.0088	69
25	2	53.44	6323.18	79.52	-0.0	79
25	3	34.64	3209.35	56.65	-0.0	85
25	4	14.85	870.92	29.51	-0.0002	101
50	1	45.11	6153.11	78.44	-0.0088	157
50	2	53.51	6323.84	79.52	-0.0001	79
50	3	34.7	3209.34	56.65	-0.0	67
50	4	14.86	871.05	29.51	-0.0003	93
100	1	45.0	6138.72	78.35	-0.0064	82
100	2	53.47	6323.29	79.52	-0.0	75
100	3	34.69	3209.31	56.65	-0.0	67
100	4	14.86	871.01	29.51	-0.0003	72

Table B.93: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	69.14	12272.8	110.78	-0.0245	72
2	2	80.65	13087.37	114.4	0.0184	67
2	3	53.2	6479.24	80.49	-0.0285	71
2	4	24.61	1724.47	41.53	-0.029	87
5	1	71.28	12719.66	112.78	-0.0618	67
5	2	80.34	12709.96	112.74	0.0467	75
5	3	53.88	6404.13	80.03	-0.0166	70
5	4	25.54	1910.21	43.71	-0.1398	72
10	1	69.25	12253.14	110.69	-0.0229	66
10	2	81.77	13334.69	115.48	-0.0001	83
10	3	51.68	6300.02	79.37	-0.0001	71
10	4	21.65	1676.89	40.95	-0.0006	94
25	1	68.69	12164.86	110.29	-0.0155	78
25	2	81.9	13337.01	115.49	-0.0003	67
25	3	51.84	6299.66	79.37	-0.0	88
25	4	21.81	1679.66	40.98	-0.0022	95
50	1	68.44	12109.78	110.04	-0.0109	67
50	2	81.69	13333.56	115.47	-0.0001	76
50	3	51.92	6299.84	79.37	-0.0	66
50	4	21.7	1677.55	40.96	-0.001	77
100	1	68.18	12031.12	109.69	-0.0044	66
100	2	81.02	13342.69	115.51	-0.0007	123
100	3	51.08	6310.81	79.44	-0.0018	78
100	4	21.43	1676.44	40.94	-0.0003	80

Table B.94: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	104.8	26970.3	164.23	-0.0586	265
2	2	107.45	19921.82	141.14	0.0061	73
2	3	84.21	13258.37	115.15	-0.0149	74
2	4	37.47	3834.6	61.92	-0.1681	77
5	1	106.61	28705.51	169.43	-0.1268	104
5	2	107.87	19750.75	140.54	0.0147	72
5	3	81.77	13183.18	114.82	-0.0091	77
5	4	34.63	3647.58	60.4	-0.1111	77
10	1	106.9	25330.24	159.15	0.0057	83
10	2	107.11	19664.52	140.23	0.019	78
10	3	82.8	13276.35	115.22	-0.0162	75
10	4	32.98	3646.91	60.39	-0.1109	79
25	1	118.67	30786.04	175.46	-0.2084	94
25	2	136.65	28870.82	169.91	-0.4403	68
25	3	90.56	14316.9	119.65	-0.0959	74
25	4	49.93	5066.57	71.18	-0.5433	75
50	1	102.26	25818.3	160.68	-0.0134	69
50	2	109.41	20112.07	141.82	-0.0033	66
50	3	79.21	13127.21	114.57	-0.0048	78
50	4	29.7	3284.38	57.31	-0.0004	82
100	1	108.18	27599.07	166.13	-0.0833	66
100	2	131.15	30760.18	175.39	-0.5345	103
100	3	106.82	22021.14	148.4	-0.6856	66
100	4	64.77	7958.87	89.21	-1.4243	90

Table B.95: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	180.54	66450.36	257.78	-0.1874	71
2	2	135.54	31466.2	177.39	0.0915	96
2	3	128.18	30420.62	174.42	-0.0758	73
2	4	81.83	11409.24	106.81	-0.495	69
5	1	172.59	56901.83	238.54	-0.0168	70
5	2	148.3	33290.98	182.46	0.0388	75
5	3	105.22	27270.79	165.14	0.0356	70
5	4	72.13	12383.46	111.28	-0.6226	70
10	1	170.09	59374.0	243.67	-0.061	67
10	2	151.45	36164.55	190.17	-0.0441	66
10	3	115.48	29614.6	172.09	-0.0473	77
10	4	56.2	6968.35	83.48	0.0869	91
25	1	217.93	78045.16	279.37	-0.3946	66
25	2	280.33	110224.47	332.0	-2.1824	66
25	3	192.52	63928.39	252.84	-1.2608	75
25	4	251.62	97334.52	311.98	-11.7539	68
50	1	179.55	63601.0	252.19	-0.1365	71
50	2	150.06	38045.98	195.05	-0.0985	79
50	3	146.75	43000.4	207.37	-0.5207	66
50	4	60.57	9189.81	95.86	-0.2042	71
100	1	184.73	59237.12	243.39	-0.0585	72
100	2	153.3	39359.01	198.39	-0.1364	67
100	3	138.39	37580.74	193.86	-0.329	81
100	4	84.21	12048.02	109.76	-0.5787	77

Table B.96: Result metrics for the $VolumeSent_{td}$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.4 BERT Models Results

B.4.1 The Non-Time Differenced Dense-BERT Model $DenseBERT_c$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	894.11	1085422.4	1041.84	-1.0526	80
2	2	828.02	833165.75	912.78	-4.1144	81
2	3	814.53	760993.6	872.35	-6.802	77
2	4	706.67	512698.97	716.03	-37.504	66
5	1	811.0	865055.25	930.08	-0.6359	84
5	2	822.53	822916.4	907.15	-4.0515	71
5	3	773.21	694852.25	833.58	-6.1238	69
5	4	208.96	56895.08	238.53	-3.2729	92
10	1	741.09	701637.6	837.64	-0.3268	66
10	2	825.31	828094.6	910.0	-4.0833	83
10	3	781.37	707920.75	841.38	-6.2578	77
10	4	102.28	15207.36	123.32	-0.1421	119
25	1	643.05	630454.06	794.01	-0.1922	81
25	2	824.55	826679.25	909.22	-4.0746	79
25	3	806.19	747478.2	864.57	-6.6634	86
25	4	273.84	88298.59	297.15	-5.6313	88
50	1	679.66	558116.4	747.07	-0.0554	80
50	2	824.67	826899.7	909.34	-4.076	66
50	3	745.78	650842.6	806.75	-5.6726	85
50	4	174.07	42792.99	206.86	-2.2138	95
100	1	849.81	1124591.5	1060.47	-1.1267	96
100	2	212.09	61422.92	247.84	0.623	71
100	3	57.08	6271.36	79.19	0.9357	68
100	4	46.72	2852.59	53.41	0.7858	99

Table B.97: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	1055.55	1577417.8	1255.95	-1.983	83
2	2	994.96	1152155.4	1073.39	-6.1026	86
2	3	975.64	1049273.1	1024.34	-9.7725	80
2	4	982.7	978985.7	989.44	-72.7186	87
5	1	1055.38	1576886.8	1255.74	-1.982	84
5	2	999.06	1160335.5	1077.19	-6.1531	75
5	3	975.21	1048441.6	1023.93	-9.764	73
5	4	963.42	941456.8	970.29	-69.8926	66
10	1	1051.82	1565494.5	1251.2	-1.9605	67
10	2	129.4	23936.76	154.72	0.8524	101
10	3	51.8	4844.8	69.6	0.9503	69
10	4	19.48	907.59	30.13	0.9317	82
25	1	157.82	39101.99	197.74	0.9261	134
25	2	91.42	12354.63	111.15	0.9238	87
25	3	47.34	4356.81	66.01	0.9553	69
25	4	38.34	2220.81	47.13	0.8328	83
50	1	130.76	25330.6	159.16	0.9521	137
50	2	91.89	12630.04	112.38	0.9221	74
50	3	52.45	5068.53	71.19	0.948	70
50	4	20.51	1140.1	33.77	0.9141	89
100	1	86.38	12241.88	110.64	0.9768	77
100	2	81.28	10687.14	103.38	0.9341	84
100	3	49.9	4781.2	69.15	0.9509	66
100	4	19.01	935.37	30.58	0.9296	99

Table B.98: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	1010.72	1433413.1	1197.25	-1.7115	69
2	2	1046.69	1256646.8	1121.0	-6.8008	71
2	3	1026.16	1150001.8	1072.38	-10.8562	85
2	4	966.14	946687.44	972.98	-70.5752	76
5	1	1013.33	1441585.5	1200.66	-1.7269	73
5	2	1042.59	1248076.4	1117.17	-6.7476	68
5	3	1006.73	1110549.1	1053.83	-10.4494	75
5	4	1038.12	1090925.1	1044.47	-81.4804	78
10	1	83.1	12050.84	109.78	0.9772	66
10	2	70.34	8135.04	90.19	0.9495	76
10	3	46.01	4908.62	70.06	0.9494	80
10	4	26.2	1443.82	38.0	0.8908	93
25	1	80.8	11721.83	108.27	0.9778	104
25	2	109.53	15568.46	124.77	0.9034	76
25	3	34.23	2913.7	53.98	0.97	92
25	4	17.73	954.84	30.9	0.9278	85
50	1	80.25	11008.1	104.92	0.9792	68
50	2	89.39	11197.37	105.82	0.9305	67
50	3	37.95	3255.46	57.06	0.9664	75
50	4	17.7	972.39	31.18	0.9265	72
100	1	112.84	20461.04	143.04	0.9613	70
100	2	98.99	13649.26	116.83	0.9153	80
100	3	46.99	4808.94	69.35	0.9504	81
100	4	49.23	3469.16	58.9	0.7377	81

Table B.99: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	963.08	1291626.5	1136.5	-1.4467	75
2	2	1051.84	1266261.6	1125.28	-6.9193	76
2	3	1039.77	1177049.6	1084.92	-11.27	67
2	4	1077.06	1173186.9	1083.14	-88.3542	75
5	1	135.51	24967.44	158.01	0.9527	90
5	2	110.02	16847.41	129.8	0.8946	85
5	3	52.26	5147.86	71.75	0.9463	69
5	4	33.08	2190.7	46.8	0.8331	89
10	1	136.98	30289.68	174.04	0.9426	78
10	2	73.64	9497.76	97.46	0.9406	68
10	3	58.1	6097.07	78.08	0.9364	72
10	4	25.05	1447.7	38.05	0.8897	67
25	1	75.23	10320.42	101.59	0.9805	95
25	2	70.77	8371.27	91.49	0.9476	66
25	3	43.45	3940.84	62.78	0.9589	72
25	4	18.81	1011.45	31.8	0.923	68
50	1	133.34	30365.14	174.26	0.9425	72
50	2	69.65	8926.58	94.48	0.9442	89
50	3	44.71	4372.38	66.12	0.9544	94
50	4	23.36	1283.75	35.83	0.9022	70
100	1	108.03	17777.18	133.33	0.9663	84
100	2	123.36	20840.98	144.36	0.8697	84
100	3	55.65	6025.4	77.62	0.9372	80
100	4	30.75	1883.78	43.4	0.8565	75

Table B.100: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	176.69	46413.92	215.44	0.9113	93
2	2	64.93	8218.19	90.65	0.9484	74
2	3	46.06	4529.88	67.3	0.9533	88
2	4	30.85	1942.15	44.07	0.8517	76
5	1	141.35	34266.62	185.11	0.9346	78
5	2	78.01	11297.98	106.29	0.9291	70
5	3	62.58	8100.23	90.0	0.9165	80
5	4	33.2	2244.92	47.38	0.8285	69
10	1	95.72	15599.28	124.9	0.9702	75
10	2	71.53	9705.32	98.52	0.9391	87
10	3	58.27	6560.95	81.0	0.9324	93
10	4	30.18	1956.78	44.24	0.8506	77
25	1	84.77	13308.57	115.36	0.9746	78
25	2	68.41	9501.53	97.48	0.9404	69
25	3	49.76	5507.71	74.21	0.9432	79
25	4	23.84	1491.67	38.62	0.8861	72
50	1	98.77	16341.96	127.84	0.9688	75
50	2	74.99	10197.25	100.98	0.936	74
50	3	52.62	6140.41	78.36	0.9367	77
50	4	33.16	2169.01	46.57	0.8343	67
100	1	117.74	20671.06	143.77	0.9605	69
100	2	86.44	13178.61	114.8	0.9173	90
100	3	65.97	7800.6	88.32	0.9196	89
100	4	35.27	2591.75	50.91	0.8021	69

Table B.101: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	139.29	29795.59	172.61	0.9433	67
2	2	97.02	15355.33	123.92	0.9021	92
2	3	76.83	12439.07	111.53	0.8683	77
2	4	41.99	3323.41	57.65	0.7421	69
5	1	122.82	25848.84	160.78	0.9508	68
5	2	112.97	22002.11	148.33	0.8598	76
5	3	84.8	13834.33	117.62	0.8536	111
5	4	53.75	5374.17	73.31	0.5829	68
10	1	99.29	18154.88	134.74	0.9654	71
10	2	102.33	18583.47	136.32	0.8816	72
10	3	84.73	14740.19	121.41	0.844	77
10	4	62.44	5850.7	76.49	0.5459	82
25	1	134.24	26795.89	163.69	0.949	73
25	2	113.61	23907.17	154.62	0.8476	70
25	3	85.68	16082.74	126.82	0.8298	76
25	4	55.73	5658.96	75.23	0.5608	83
50	1	162.54	40108.24	200.27	0.9237	67
50	2	113.15	22850.76	151.16	0.8544	83
50	3	71.67	11110.04	105.4	0.8824	69
50	4	44.64	3708.21	60.9	0.7122	66
100	1	165.97	40746.65	201.86	0.9224	68
100	2	109.68	19899.53	141.07	0.8732	69
100	3	77.14	11590.27	107.66	0.8773	70
100	4	60.87	6656.1	81.58	0.4834	67

Table B.102: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	158.93	41695.83	204.2	0.9208	72
2	2	127.44	27117.42	164.67	0.8206	72
2	3	134.93	27425.08	165.61	0.6859	69
2	4	77.66	10117.09	100.58	0.2382	67
5	1	145.04	41409.78	203.49	0.9213	68
5	2	136.4	30199.1	173.78	0.8002	76
5	3	150.82	35386.74	188.11	0.5948	79
5	4	92.66	14812.9	121.71	-0.1154	66
10	1	187.96	59968.08	244.88	0.8861	98
10	2	141.99	32449.88	180.14	0.7853	81
10	3	159.11	46073.6	214.65	0.4724	75
10	4	81.64	11224.14	105.94	0.1548	89
25	1	177.44	47331.98	217.56	0.9101	79
25	2	153.99	40016.29	200.04	0.7353	70
25	3	141.45	43254.96	207.98	0.5046	70
25	4	101.61	17735.6	133.18	-0.3355	91
50	1	167.36	43955.06	209.65	0.9165	96
50	2	117.45	22768.22	150.89	0.8494	68
50	3	110.6	22351.35	149.5	0.744	68
50	4	88.46	13040.37	114.19	0.018	66
100	1	181.0	54814.32	234.12	0.8959	66
100	2	168.09	44616.04	211.23	0.7048	71
100	3	140.34	33662.77	183.47	0.6145	67
100	4	205.54	53674.92	231.68	-3.0418	68

Table B.103: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	187.54	76032.98	275.74	0.8506	84
2	2	202.71	64950.56	254.85	0.5382	70
2	3	195.4	64777.18	254.51	0.2168	68
2	4	166.18	44900.24	211.9	-2.1629	81
5	1	210.81	71503.5	267.4	0.8595	72
5	2	173.31	42235.9	205.51	0.6997	76
5	3	199.08	70086.16	264.74	0.1527	82
5	4	109.64	19393.54	139.26	-0.3661	87
10	1	252.0	103853.3	322.26	0.7959	73
10	2	214.64	80065.6	282.96	0.4307	73
10	3	143.37	52761.2	229.7	0.3621	71
10	4	175.39	51204.67	226.28	-2.607	69
25	1	243.91	95521.79	309.07	0.8123	69
25	2	226.96	75490.9	274.76	0.4633	90
25	3	174.74	68796.7	262.29	0.1682	66
25	4	192.12	59173.73	243.26	-3.1683	73
50	1	258.99	100834.0	317.54	0.8018	93
50	2	252.18	99161.68	314.9	0.295	68
50	3	185.37	62437.54	249.88	0.2451	78
50	4	158.36	40640.0	201.59	-1.8628	83
100	1	752.22	963893.44	981.78	-0.8943	65
100	2	312.03	126121.91	355.14	0.1033	140
100	3	163.37	53809.4	231.97	0.3494	66
100	4	100.75	19471.03	139.54	-0.3716	79

Table B.104: Result metrics for the $DenseBERT_c$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

B.4.2 The Time Differenced Dense-BERT Model $DenseBERT_{td}$

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	10.48	297.8	17.26	-0.0004	145
2	2	11.83	363.41	19.06	-0.0019	267
2	3	8.46	215.29	14.67	-0.0022	147
2	4	4.69	92.82	9.63	-0.0141	187
5	1	10.48	297.78	17.26	-0.0003	159
5	2	11.83	363.38	19.06	-0.0018	269
5	3	8.45	215.2	14.67	-0.0018	203
5	4	4.71	92.93	9.64	-0.0153	78
10	1	10.48	297.79	17.26	-0.0003	156
10	2	11.84	363.7	19.07	-0.0026	178
10	3	8.45	215.01	14.66	-0.0009	246
10	4	4.56	91.63	9.57	-0.0011	147
25	1	10.48	297.79	17.26	-0.0003	367
25	2	11.84	363.77	19.07	-0.0028	224
25	3	8.46	215.25	14.67	-0.002	89
25	4	4.59	91.85	9.58	-0.0035	208
50	1	10.48	297.81	17.26	-0.0004	131
50	2	11.83	363.2	19.06	-0.0013	108
50	3	8.45	214.99	14.66	-0.0008	176
50	4	4.56	91.64	9.57	-0.0012	72
100	1	10.48	297.82	17.26	-0.0004	153
100	2	11.83	363.14	19.06	-0.0011	236
100	3	8.45	215.01	14.66	-0.0009	103
100	4	4.58	91.74	9.58	-0.0023	80

Table B.105: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 10 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	17.51	907.57	30.13	-0.0028	68
2	2	19.83	1010.74	31.79	-0.0008	82
2	3	14.81	743.89	27.27	-0.0078	226
2	4	7.36	263.33	16.23	-0.0001	82
5	1	17.49	906.92	30.12	-0.002	98
5	2	19.84	1010.54	31.79	-0.0006	76
5	3	14.86	745.37	27.3	-0.0098	67
5	4	7.37	263.3	16.23	-0.0	74
10	1	17.55	909.32	30.15	-0.0047	67
10	2	19.83	1010.83	31.79	-0.0009	72
10	3	14.83	744.47	27.28	-0.0085	68
10	4	7.37	263.29	16.23	-0.0	82
25	1	17.54	908.8	30.15	-0.0041	67
25	2	19.84	1010.92	31.79	-0.001	70
25	3	14.81	743.9	27.27	-0.0078	75
25	4	7.37	263.29	16.23	-0.0	71
50	1	17.49	906.66	30.11	-0.0018	81
50	2	19.83	1010.61	31.79	-0.0007	75
50	3	14.83	744.66	27.29	-0.0088	74
50	4	7.37	263.31	16.23	-0.0001	87
100	1	17.49	906.99	30.12	-0.0021	70
100	2	19.84	1010.46	31.79	-0.0005	69
100	3	14.81	744.09	27.28	-0.008	69
100	4	7.37	263.31	16.23	-0.0001	69

Table B.106: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 30 minutes and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	25.14	1825.09	42.72	-0.0085	76
2	2	28.66	2064.76	45.44	-0.0004	68
2	3	20.04	1172.9	34.25	-0.003	77
2	4	9.87	466.44	21.6	-0.0001	68
5	1	25.03	1816.89	42.63	-0.004	79
5	2	28.65	2064.64	45.44	-0.0003	72
5	3	20.06	1174.76	34.27	-0.0046	93
5	4	9.87	466.42	21.6	-0.0	82
10	1	25.07	1820.16	42.66	-0.0058	78
10	2	28.69	2065.83	45.45	-0.0009	81
10	3	20.02	1170.64	34.21	-0.0011	66
10	4	9.87	466.44	21.6	-0.0	89
25	1	25.09	1821.63	42.68	-0.0066	87
25	2	28.62	2064.04	45.43	-0.0	87
25	3	20.04	1173.61	34.26	-0.0036	96
25	4	9.88	466.55	21.6	-0.0003	90
50	1	25.06	1819.52	42.66	-0.0054	73
50	2	28.66	2064.81	45.44	-0.0004	66
50	3	20.03	1172.68	34.24	-0.0028	81
50	4	9.87	466.42	21.6	-0.0	71
100	1	25.06	1819.3	42.65	-0.0053	70
100	2	28.66	2064.83	45.44	-0.0004	69
100	3	20.03	1172.53	34.24	-0.0027	67
100	4	9.87	466.42	21.6	-0.0	68

Table B.107: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 1 hour and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	36.07	3739.64	61.15	-0.0069	133
2	2	42.16	4291.15	65.51	-0.0001	86
2	3	29.38	2162.45	46.5	-0.0	67
2	4	12.84	559.72	23.66	-0.0	111
5	1	36.08	3741.03	61.16	-0.0073	77
5	2	42.17	4291.27	65.51	-0.0001	91
5	3	29.37	2162.76	46.51	-0.0002	75
5	4	12.84	559.73	23.66	-0.0	81
10	1	36.08	3740.46	61.16	-0.0071	133
10	2	42.15	4291.03	65.51	-0.0001	98
10	3	29.37	2162.72	46.51	-0.0001	71
10	4	12.84	559.72	23.66	-0.0	90
25	1	36.08	3740.85	61.16	-0.0072	176
25	2	42.14	4290.92	65.51	-0.0	66
25	3	29.38	2162.47	46.5	-0.0	76
25	4	12.84	559.87	23.66	-0.0003	79
50	1	36.05	3738.47	61.14	-0.0066	87
50	2	42.15	4291.01	65.51	-0.0001	90
50	3	29.38	2162.51	46.5	-0.0	71
50	4	12.84	559.79	23.66	-0.0001	88
100	1	35.99	3733.6	61.1	-0.0053	130
100	2	42.12	4290.77	65.5	-0.0	78
100	3	29.38	2162.5	46.5	-0.0	79
100	4	12.84	559.72	23.66	-0.0	88

Table B.108: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 2 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	45.25	6170.09	78.55	-0.0116	126
2	2	53.55	6324.53	79.53	-0.0002	78
2	3	34.73	3209.49	56.65	-0.0001	108
2	4	14.88	871.41	29.52	-0.0007	72
5	1	45.26	6171.32	78.56	-0.0118	168
5	2	53.54	6324.47	79.53	-0.0002	69
5	3	34.71	3209.39	56.65	-0.0	70
5	4	14.88	871.4	29.52	-0.0007	80
10	1	45.1	6151.54	78.43	-0.0085	91
10	2	53.45	6323.19	79.52	-0.0	81
10	3	34.64	3209.35	56.65	-0.0	79
10	4	14.85	870.93	29.51	-0.0002	75
25	1	45.12	6153.39	78.44	-0.0088	110
25	2	53.44	6323.19	79.52	-0.0	90
25	3	34.65	3209.33	56.65	-0.0	82
25	4	14.85	870.92	29.51	-0.0002	80
50	1	45.11	6153.25	78.44	-0.0088	174
50	2	53.51	6323.83	79.52	-0.0001	105
50	3	34.7	3209.35	56.65	-0.0	103
50	4	14.86	871.09	29.51	-0.0004	78
100	1	45.0	6138.72	78.35	-0.0064	74
100	2	53.48	6323.4	79.52	-0.0	72
100	3	34.69	3209.31	56.65	-0.0	71
100	4	14.86	871.01	29.51	-0.0003	71

Table B.109: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 3 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	68.99	12215.52	110.52	-0.0198	66
2	2	88.1	15453.16	124.31	-0.159	67
2	3	51.36	6240.7	79.0	0.0094	76
2	4	38.31	3945.84	62.82	-1.3545	69
5	1	68.61	12148.39	110.22	-0.0141	75
5	2	101.75	19549.0	139.82	-0.4662	68
5	3	51.09	6294.05	79.34	0.0009	106
5	4	34.17	3809.83	61.72	-1.2733	71
10	1	69.25	12253.02	110.69	-0.0229	79
10	2	81.77	13334.68	115.48	-0.0001	71
10	3	51.68	6300.02	79.37	-0.0001	78
10	4	21.65	1676.9	40.95	-0.0006	71
25	1	68.79	12184.44	110.38	-0.0172	66
25	2	81.91	13337.04	115.49	-0.0003	66
25	3	51.84	6299.67	79.37	-0.0	74
25	4	21.82	1679.69	40.98	-0.0023	97
50	1	68.44	12110.06	110.05	-0.0109	66
50	2	81.69	13333.55	115.47	-0.0001	82
50	3	51.92	6299.84	79.37	-0.0	66
50	4	21.7	1677.55	40.96	-0.001	66
100	1	68.18	12030.81	109.69	-0.0043	66
100	2	81.02	13342.63	115.51	-0.0007	67
100	3	51.09	6310.63	79.44	-0.0017	70
100	4	21.43	1676.48	40.94	-0.0003	76

Table B.110: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 6 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	108.86	26542.49	162.92	-0.0419	72
2	2	122.31	24131.06	155.34	-0.2038	77
2	3	151.55	41939.19	204.79	-2.2102	71
2	4	93.94	16771.88	129.51	-4.1089	83
5	1	103.58	26377.43	162.41	-0.0354	66
5	2	111.66	19840.73	140.86	0.0102	67
5	3	115.95	23197.22	152.31	-0.7756	78
5	4	58.8	8108.14	90.05	-1.4698	68
10	1	101.74	25811.3	160.66	-0.0132	67
10	2	133.8	27012.18	164.35	-0.3476	66
10	3	142.46	32458.47	180.16	-1.4845	71
10	4	31.13	3362.47	57.99	-0.0242	91
25	1	102.68	26061.57	161.44	-0.023	68
25	2	111.22	20074.13	141.68	-0.0014	66
25	3	79.04	13144.59	114.65	-0.0062	69
25	4	30.34	3288.33	57.34	-0.0017	71
50	1	102.26	25817.67	160.68	-0.0134	69
50	2	109.41	20111.99	141.82	-0.0033	66
50	3	79.21	13127.2	114.57	-0.0048	70
50	4	29.7	3284.42	57.31	-0.0005	66
100	1	108.41	27658.53	166.31	-0.0857	68
100	2	116.78	21084.44	145.2	-0.0519	79
100	3	94.48	18125.0	134.63	-0.3874	93
100	4	75.18	9488.63	97.41	-1.8903	88

Table B.111: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 12 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

w	Fold NR.	MAE	MSE	RMSE	R2	Epochs trained
2	1	172.44	54399.54	233.24	0.0279	67
2	2	212.43	63443.02	251.88	-0.8317	69
2	3	285.16	132323.8	363.76	-3.6796	66
2	4	235.11	103506.34	321.72	-12.5626	84
5	1	169.36	56103.26	236.86	-0.0025	68
5	2	277.71	123071.33	350.82	-2.5533	70
5	3	160.5	43538.19	208.66	-0.5397	74
5	4	154.14	39324.43	198.3	-4.1528	72
10	1	173.05	56185.13	237.03	-0.004	66
10	2	262.25	95629.96	309.24	-1.761	92
10	3	213.72	65148.65	255.24	-1.304	68
10	4	163.9	35273.26	187.81	-3.6219	73
25	1	173.75	59777.38	244.49	-0.0682	66
25	2	148.54	35236.92	187.72	-0.0174	72
25	3	132.76	35269.58	187.8	-0.2473	83
25	4	47.47	7654.44	87.49	-0.003	71
50	1	189.91	64485.02	253.94	-0.1523	95
50	2	184.16	50657.77	225.07	-0.4626	71
50	3	189.05	52547.95	229.23	-0.8583	66
50	4	78.16	11217.13	105.91	-0.4698	68
100	1	184.78	59281.5	243.48	-0.0593	77
100	2	128.06	29202.32	170.89	0.1569	79
100	3	171.53	43716.54	209.09	-0.546	67
100	4	168.72	41181.18	202.93	-4.3961	74

Table B.112: Result metrics for the $DenseBERT_{td}$ model trained to predict the next interval's price with a time interval of 24 hours and a batch size of 64 for different window sizes and data folds. The fold NR. indicates the fold number from Table 3.11 and w is the window size.

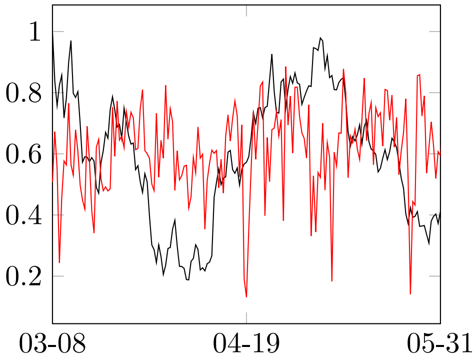
Appendix C

The Third Appendix

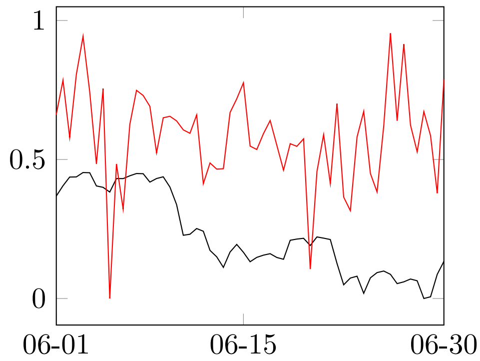
In this appendix, extra plots regarding the correlation between features can be found.

C.1 Correlation Plots

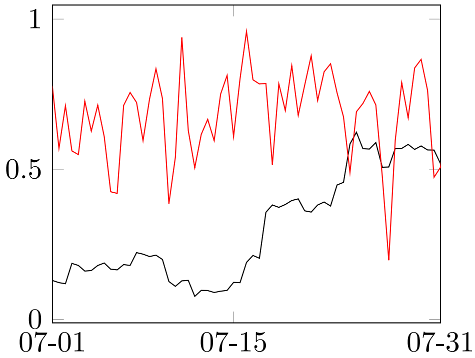
C.1.1 Sentiment Correlation



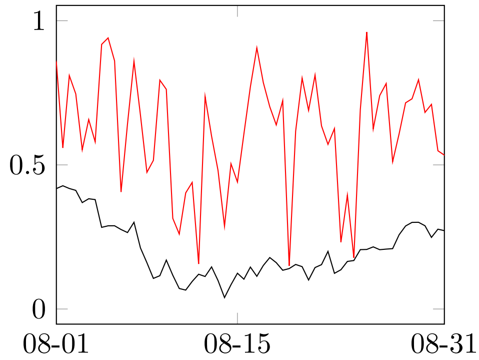
(a) Mart - May



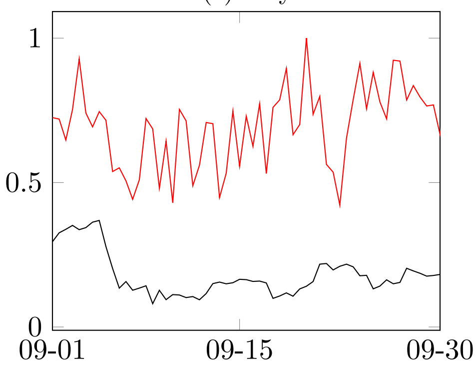
(b) June



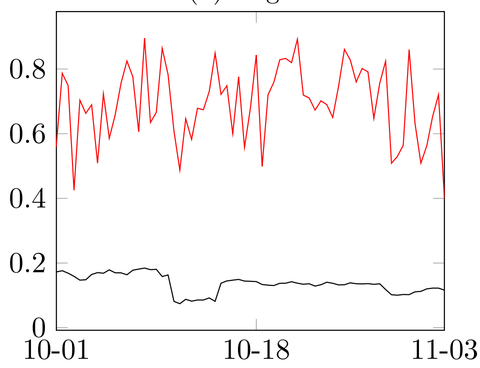
(c) July



(d) August



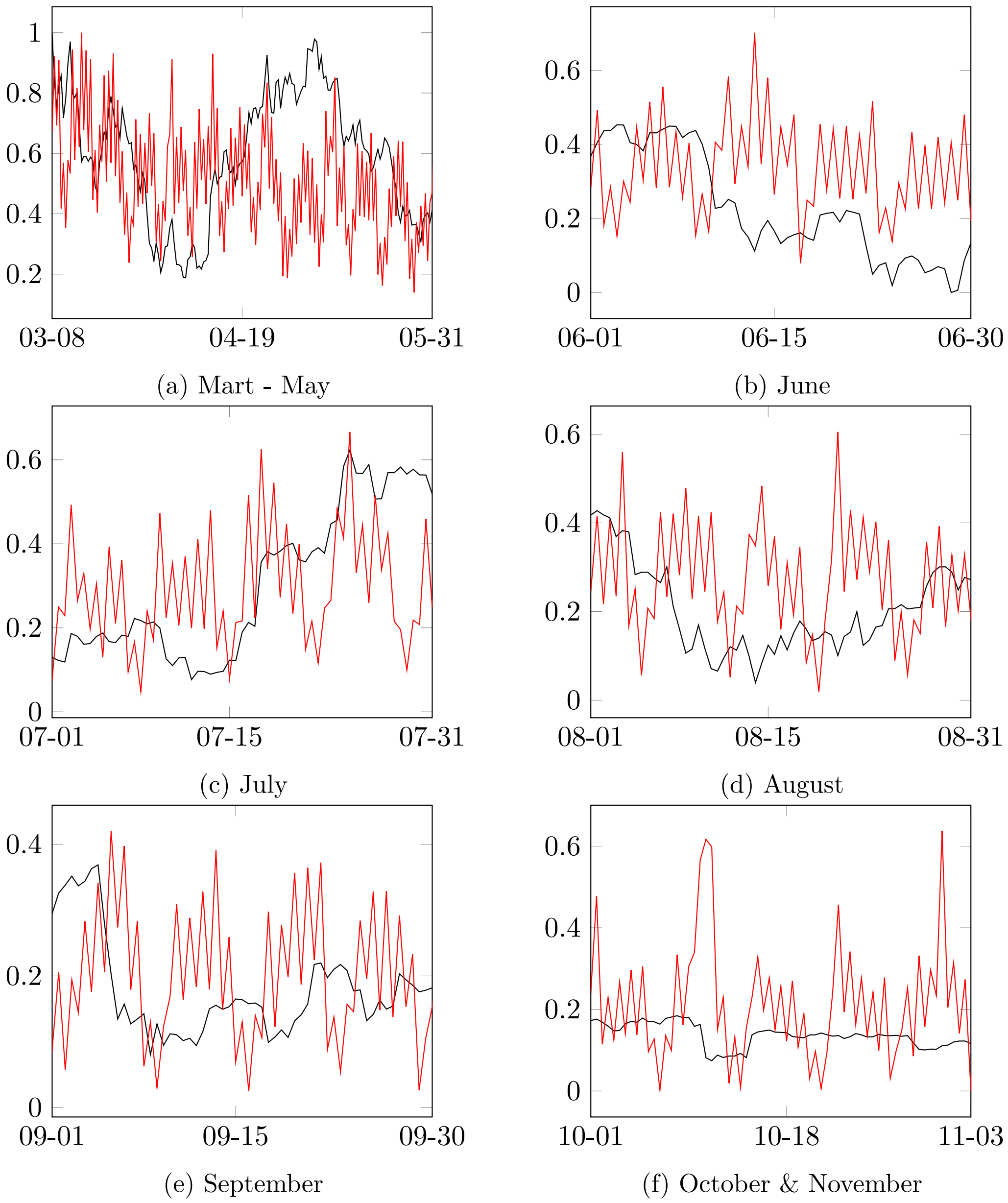
(e) September



(f) October & November

Figure C.1: The normalized actual price (black) and the normalized compound score (red) plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is different for each of the figures and is noted in their corresponding captions.

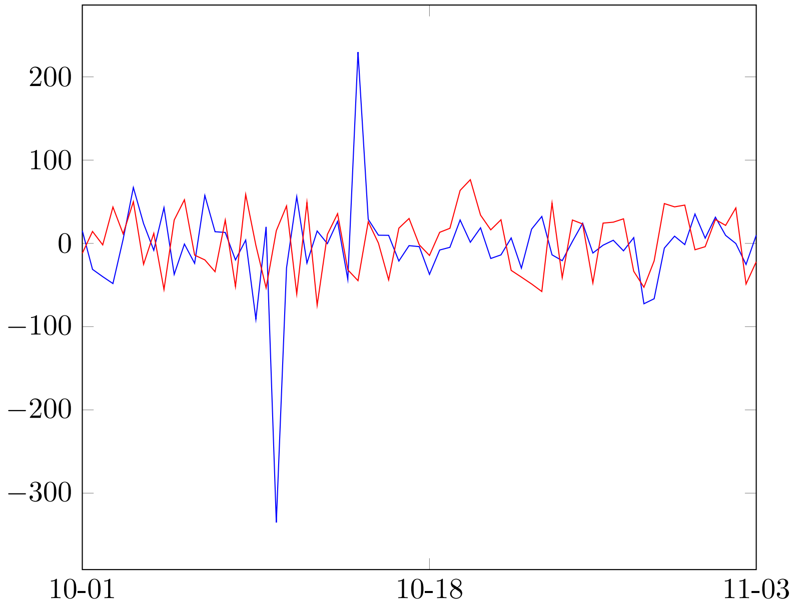
C.1.2 Volume Correlation



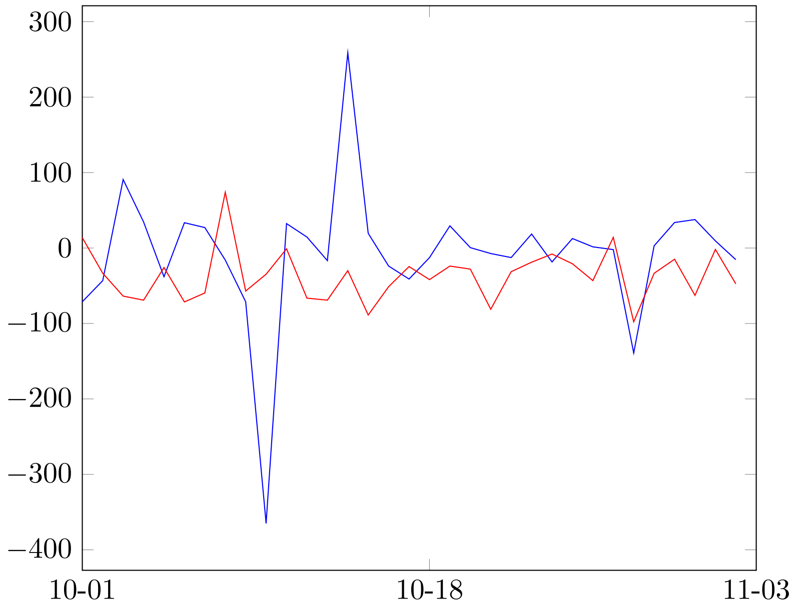
(a) Mart - May (b) June (c) July (d) August (e) September (f) October & November

Figure C.2: The normalized actual price (black) and the normalized filtered tweet volume (red) plotted over time with a time interval of 12 hours. These values are min-max normalized over the whole dataset. The plotted period is different for each of the figures and is noted in their corresponding captions.

C.2 Extra Plots



(a) $w = 25$, $i = 12$ hours and $RMSE = 72$.



(b) $w = 25$, $i = 24$ hours and $RMSE = 98$.

Figure C.3: $Base_{td}$ model's extra prediction plots.

Bibliography

[1] Bitcoin, now legal tender in el salvador, greeted with tech problems, volatility concerns. www.cbc.ca/news/world/el-salvador-bitcoin-1.6166977. 7 September, 2021.

[2] Bitinfocharts. www.bitinfocharts.com.

[3] coinmarketcap. www.coinmarketcap.com.

[4] Crypto data download. www.cryptodatadownload.com/.

[5] Github bertweet: A pre-trained language model for english tweets. https://github.com/VinAIResearch/BERTweet.

[6] Github vadersentiment. https://github.com/cjhutto/vaderSentiment#about-the-scoring.

[7] Google trends (2021). https://trends.google.com/. Accessed on October 20, 2021.

[8] hedonometer. www.hedonometer.org.

[9] Interpret the key results for correlation. https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/modeling-statistics/regression/how-to/correlation/interpret-the-results/.

[10] Microsoft surveys xbox users about potential bitcoin payment option. https://thepaypers.com/cryptocurrencies/microsoft-surveys-xbox-users-about-potential-bitcoin-payment-option--1247951. March 24, 2021.

[11] Trading volume. www.bitrates.com/guides/terminology/trading-volume.

[12] Twitter api9. https://developer.twitter.com/en.

[13] How cryptocurrency is different from stock exchange. https://timesofindia.indiatimes.com/business/cryptocurrency/how-cryptocurrency-is-different-from-stock-exchange/articleshow/85484613.cms, August 20, 2021.

[14] J. Abraham, D. Higdon, J. Nelson, and J. Ibarra. Cryptocurrency price prediction using tweet volumes and sentiment analysis. 2018.

[15] A. Aggarwal, I. Gupta, N. Garg, and A. Goel. Deep learning approach to determine the impact of socio economic factors on bitcoin price prediction. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–5, 2019.

[16] A. Altin. Github twitter bitcoin predictor thesis. `https://github.com/AytugAltin/TwitterBitcoinPredictor-thesis`.

[17] L. Ante. How elon musk's twitter activity moves cryptocurrency markets. 02 2021.

[18] F. Archilla. Tweets containing the word bitcoin 2017-2019. `www.kaggle.com/franciscoarchilla/tweets-containing-the-word-bitcoin-20172019/`.

[19] C. Atiyeh. Tesla backtracked on accepting bitcoin, but there are car dealers that take it. `www.caranddriver.com/news/a36434628/tesla-bitcoin-wont-accept/`, May 16, 2021.

[20] P. Calderon. Vader sentiment analysis explained. `https://medium.com/@piocalderon/vader-sentiment-analysis-explained-f1c4f9101cd9`, April 10, 2017.

[21] S. Cavalli and M. Amoretti. Cnn-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, 101:107065, 2021.

[22] J. Copeland. Zipfian distribution of cryptocurrency market capitalizations. `https://medium.com/@jamiecopeland_52423/zipfian-distribution-of-cryptocurrency-market-capitalizations-649bf31e2fb8`, October 6, 2017. 2017-10.

[23] A. Deebadi. Understanding impact of twitter feed on bitcoin price and trading patterns, 2020.

[24] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[25] V. Flovik. How (not) to use machine learning for time series forecasting: Avoiding the pitfalls. `https://towardsdatascience.com/how-not-to-use-machine-learning-for-time-series-forecasting-avoiding-the-pitfalls-19f9d7adf424`. June 6, 2018.

[26] J. Frost. How to interpret r-squared in regression analysis. `https://statisticsbyjim.com/regression/interpret-r-squared-regression/`.

[27] S. Ghelani. Breaking bert down. `https://towardsdatascience.com/breaking-bert-down-430461f60efb`, July 26, 2019.

[28] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[29] R. Horev. Bert explained: State of the art language model for nlp. `https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270`, November 10, 2018.

[30] C. Hutto and E. Gilbert. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014.

[31] S. Hyrynsalmi, S. Hyrynsalmi, and K. Kimppa. *Blockchain Ethics: A Systematic Literature Review of Blockchain Research*, pages 145–155. 08 2020.

[32] R. Islam, Y. Fujiwara, S. Kawata, and H. Yoon. Analyzing outliers activity from the time-series transaction pattern of bitcoin blockchain. *Evolutionary and institutional economics review*, 16(1):239–257, 2019.

[33] T. Kaizoji and M. Miyano. Zipf's law for share price and company fundamentals, 2017.

[34] O. Kraaijeveld and J. D. Smedt. The predictive power of public twitter sentiment for forecasting cryptocurrency prices. *Journal of International Financial Markets, Institutions and Money*, 65:101188, 2020.

[35] R. Kumar. All you need to know about bert. `https://www.analyticsvidhya.com/blog/2021/05/all-you-need-to-know-about-bert/`, May 27, 2021.

[36] L. Li, A. Arab, J. Liu, J. Liu, and Z. Han. Bitcoin options pricing using lstm-based prediction model and blockchain statistics. In *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 67–74, 2019.

[37] Y. Li and W. Dai. Bitcoin price forecasting method based on cnn-lstm hybrid neural network model. *The Journal of Engineering*, 2020, 01 2020.

[38] E. Loper and S. Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.

[39] D. Ma and H. Tanizaki. On the day-of-the-week effects of bitcoin markets: international evidence. 9(4):455–478, 2019.

[40] Y. Ma. Nlp: How does nltk.vader calculate sentiment? `https://medium.com/ro-codes/nlp-how-does-nltk-vader-calculate-sentiment-6c32d0f5046b`, February 5, 2020.

[41] D. C. Mallqui and R. A. Fernandes. Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques. *Applied Soft Computing*, 75:596–606, 2019.

[42] A. Mittal, V. Dhiman, A. Singh, and C. Prakash. Short-term bitcoin price fluctuation prediction using social media and web search data. In *2019 Twelfth International Conference on Contemporary Computing (IC3)*, pages 1–6, 2019.

[43] S. Mohapatra, N. Ahmed, and P. Alencar. Kryptooracle: A real-time cryptocurrency price prediction platform using twitter sentiments, 2020.

[44] M. A. Naeem, I. Mbarki, M. T. Suleman, X. V. Vo, and S. J. H. Shahzad. Does twitter happiness sentiment predict cryptocurrency? 2020.

[45] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Decentralized Business Review*, page 21260, 2008.

[46] D. Q. Nguyen, T. Vu, and A. T. Nguyen. Bertweet: A pre-trained language model for english tweets, 2020.

[47] C. Olah. Understanding lstm networks. `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`, August 27, 2015.

[48] S. S. Öztürk and M. E. Bilgiç. Twitter & bitcoin: are the most influential accounts really influential? *Applied Economics Letters*, 0(0):1–4, 2021.

[49] T. Pano and R. Kashef. A complete vader-based sentiment analysis of bitcoin (btc) tweets during the era of covid-19. *Big Data and Cognitive Computing*, 4(4):33, 2020.

[50] D. R. Pant, P. Neupane, A. Poudel, A. K. Pokhrel, and B. K. Lama. Recurrent neural network based bitcoin price prediction by twitter sentiment analysis. In *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, pages 128–132, 2018.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[52] A. pratap Singh. Evaluation metrics for regression models- mae vs mse vs rmse vs rmsle. `https://akhilendra.com/evaluation-metrics-regression-mae-mse-rmse-rmsle/`, March 20, 2019.

[53] E. Richards. What is the difference between cell state denoted c(t) and the output of the state denoted by h(t) in a recurrent neural network? `https://www.quora.com/What-is-the-difference-between-cell-state-denoted-c-t-and-the-output-of-the-state-denoted-by-h-t-in-a-recurrent-neural-network`, September 4, 2019.

[54] J. Royal. 12 most popular types of cryptocurrency. `www.bankrate.com/investing/types-of-cryptocurrency/`, December 10, 2021.

[55] D. Shen, A. Urquhart, and P. Wang. Does twitter predict bitcoin? *Economics Letters*, 174:118–122, 2019.

[56] S. Shrivastava. Cross validation in time series. `https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4`, January 14, 2020.

[57] A. P. Singh. Are multinationals enthusiastic about bitcoin? these 5 major companies are exploring the cryptocurrency frontier. `www.ndtv.com/business/are-multinationals-enthusiastic-about-bitcoin-here-are-5-big-companies-that-accept-the-cryptocurrency-2520197`, August 30, 2021.

[58] N. Smuts. What drives cryptocurrency prices? an investigation of google trends and telegram sentiment. *Performance Evaluation Review*, 46(3):131–134, 2019.

[59] F. Steinmetz, M. von Meduna, L. Ante, and I. Fiedler. Ownership, uses and perceptions of cryptocurrency: Results from a population survey. *Technological forecasting & social change*, 173, 2021.

[60] E. Stenqvist and J. Lönnö. Predicting bitcoin price fluctuation with twitter sentiment analysis.

[61] M. Thakur. Realized volatility. `www.wallstreetmojo.com/realized-volatility/`.

[62] B. Tu, L. Wei, Y. Jia, and J. Qian. Using baidu search values to monitor and predict the confirmed cases of covid-19 in china:–evidence from baidu index. *BMC infectious diseases*, 21(1):1–12, 2021.

[63] P. van Kooten. Python library: contractions. `https://pypi.org/project/contractions/`.

[64] M. Yao. 10 leading language models for nlp in 2021. `https://www.topbots.com/leading-nlp-language-models-2020/`, May 11, 2021.

[65] K. Żbikowski. *Application of Machine Learning Algorithms for Bitcoin Automated Trading*, pages 161–168. 03 2016.

[66] X. Zhang, F. Lu, R. Tao, and S. Wang. The time-varying causal relationship between the bitcoin market and internet attention. *Financial Innovation*, 7(1):1–19, 2021.