Investigation on the effects of sampling periods on the accuracy of time-series prediction models for Bitcoin pricing

To what extent is the accuracy of LSTM neural Networks for the prediction of bitcoin pricing influenced by time frequencies?

A Computer Science Extended Essay

Words: 3897

TABLE OF CONTENTS

Introduction **3**

Background Information **4**
Deep Learning 4
Long-Short-Term-Memory (LSTM) 6
Representation of a LSTM network 7
Sigmoid Layer 7

Experiment Methodology **10**
The Datasets used 11
Processing the Datasets for use 11
Variables 12
Accuracy 12
Preparing the three LSTMs 12
The Experimental Procedure 13
Graphical Representation 15
Data Analysis 16
Analyzing Real-Time data 16
Analyzing Time Frequencies 17
Making sense of the differences in Accuracy of Minute, Monthly or Daily Predictions for Bitcoin Price 17
Comparing the Resultant Datasets 19

Conclusion **21**

Bibliography **23**

Appendix A **26**

Appendix C **36**

Appendix D **36**

Appendix E **38**

Appendix F **39**

Introduction

As a newly developed and widely accepted electronic alternative to traditional trade methods, cryptocurrencies (cryptos) have established significant economic ramifications for both developing nations and the global economy at large (Worldcoin). However, this ever-expanding financial industry is marked by high volatility and persistently high price swings, the reason for which a market focusing on their predictability has formed (Graubard & Eaddy). LSTM models are seen as a crucial element by services and researchers aiming to assist investors towards the right decisions, as they are capable of efficiently capturing sequence patterns as well as long and short-term data dependencies (Schmidhofer). While extremely nonlinear time-series problems may be addressed by this sophisticated deep learning model, it has been shown that these algorithms often give erroneous crypto projections. A possible cause for this phenomenon, which is to be addressed within this paper, is the lack of scientific literature evaluating the extent to which the accuracy of LSTM neural Networks for the prediction of bitcoin prices is influenced by time frequencies, which highlights a disregard towards one of the most relevant independent variables.

With a global market share of 2.1 Trillion USD (Coinmarketcap), Cryptos are considered an essential component of our economy despite many potential investors being held back by the constant price fluctuations (McCluskey). This is supported by **bitcoin**, the main digital currency and test subject for this research, having lost around 30% of its value within one day and being subject to constant price variability since November 2021 (Morris). Various technologies surrounding cryptocurrency trade and prediction are emerging, however, a median accuracy touching up 55-65% (Springer) evidences room for potential improvement. My research aims to fill this gap by offering a better understanding of the impact time-frequency has on algorithmic accuracy, as it is one of the most disputed independent variables within neural networks of any kind (Ellis).

This research is worthy of investigation as the findings can aid programmers in bettering existing and future LSTM neural networks by placing importance on the testing of time series, which in the bigger picture creates more stability for investors of any sort, by alerting them in advance of the plausible occurrences in such an unpredictable field.

This paper seeks to investigate the extent to which a configurable property in LSTM neural networks affects its performance, specifically its data analysis capacity and its resultant accuracy in bitcoin price predictions. To investigate the influence, three identical LSTMs models were programmed and were each trained with different time series gained from public datasets; the subsets being minute, hourly and daily frequencies. Their individual predictions spanned a 30-day time-long period with the scope of emerging patterns in their performance to then analyze. The results were analyzed in terms of their logical and mathematical justifications to determine the accuracy of the predictions.

Background Information

Deep Learning

Recurrent Neural Networks are based on deep learning which is a data analysis method that automates the construction of analytical models. It is the most advanced branch of Machine Learning and is based on the idea that systems can learn from data, identify patterns on their own and make decisions with minimal human intervention (Selig, J.). With enough data, the system may solve machine learning issues and learn the proper representation without the requirement for data pre-processing, as is the case with conventional machine learning techniques.

In other words, Deep Learning is a learning technique in which artificial neural networks are exposed to vast amounts of data, so that they are capable of learning long-term dependencies, particularly in issues involving sequence prediction (intellipaat). In this experiment, an LSTM could be given a collection of stock price values of the previous days and the internal layers would analyze their properties and attempt to generate a behavioral sequence to predict the next day's price. This process is referred to as training where each layer calculates the values for the next one, in order to process the information in an ever more complete way (Rakshith Vasudev). If the program is successful, it will eventually be able to recognize patterns within the studied groups' behavior to fetch correct estimates of future references. From the accuracy rate of the predictions, which are based on experimental and theoretical evidence, the developers can determine whether their proposed training framework assures the "suitability" for the real-world usage of said time-series deep learning model.

Recurrent Neural Network (RNN)

Basic neural networks are constrained by their inability to establish persistent learning (Dongens). One of the allures of RNNs for this experiment is the potential for them to make connections between prior knowledge and the work at hand. An RNN in its simplest form is represented by an artificial neural network that reports as input the output data of the previous step (See Figure 1). The input consists not only of the current data but also of the output result obtained in the previous phase (r2rt). It is made up of identical feedforward neural networks, referred to as "RNN cells," one for each instant or step in time. These cells can be composed because they run on their own output. They are also capable of processing outside information and generating outside output (See Figure 2). In other words, these cells, which represent the network's recurrent component due to their looping, can be conceptualized as numerous replications of the same network that communicate with one another by passing messages.

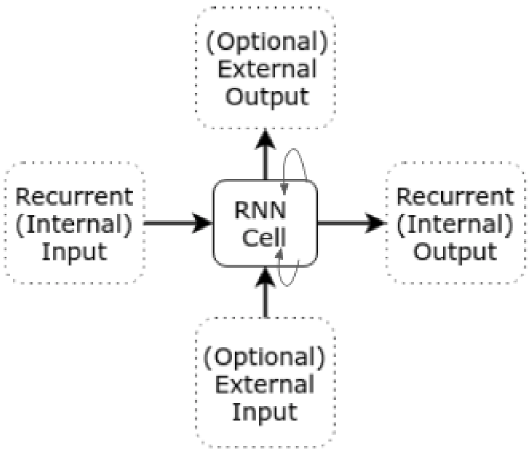
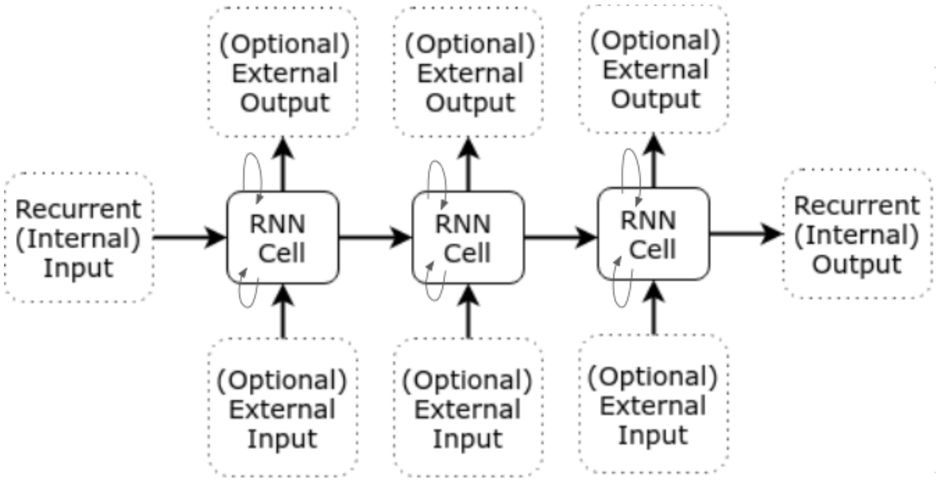


Figure 1: diagrammatic representation of a Singular RNN cell. Adapted from “Understanding, Deriving and Extending the LSTM - R2RT”



Bc

Figure 2: diagrammatic representation of three composed RNN cells. Adapted from “Understanding LSTM Networks -- colah's blog”

Long-Short-Term-Memory (LSTM)

A RNN suffers from long-term memory loss (Dongens), as it calculates the output based on what it remembers from the step prior rather than taking into account its database as a whole. One of the aspects making an LSTM fit for this research is the model’s ability to learn from long-time sequences and retain their memory. The inner workings of an LSTM are discussed below.

Representation of an LSTM network

Similarly to RNNs, LSTM layers are made up of various cells. Each LSTM cell will only consider a current column of its inputs, and also the previous column's LSTM cell's output. Normally, LSTMs receive an entire matrix for input, with each column corresponding to an element that predates the subsequent column (Luciano Strika). The purpose is for each LSTM cell to have two different input vectors: its own input column and the output of the LSTM cell before it, which provides some context for the preceding input column.

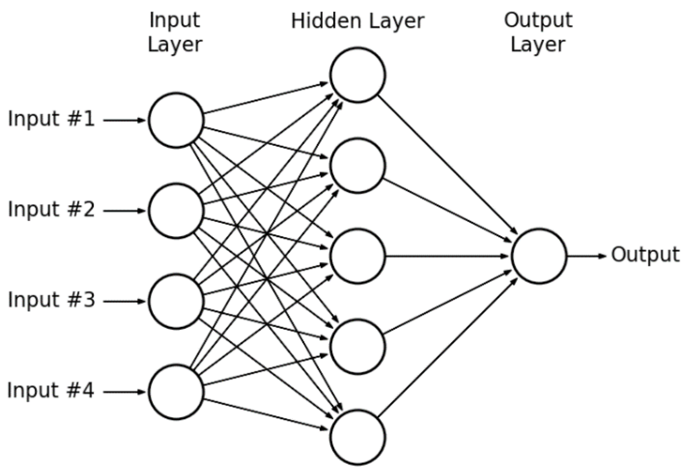


Figure 5: A hypothetical example of a Multilayer Perceptron Network. Copied from Zahran Mohamed

Sigmoid Layer

An LSTM can modify the cell state by removing or adding information. This process is carefully controlled via gates which consist of a pointwise multiplication process and a layer of sigmoid neural networks. To indicate how much of each component should be allowed through, the sigmoid layer generates integers between zero and one (See Figure 3). Information is deemed lost if the multiplication yields a result of 0. Similarly, if the value is 1, the data is retained. This will aid the network in the experiment in learning which data can be lost and which should be kept (Singhal G.). Three of these gates are present in an LSTM to safeguard and regulate the cell state. The entire process is described in Figure 4.

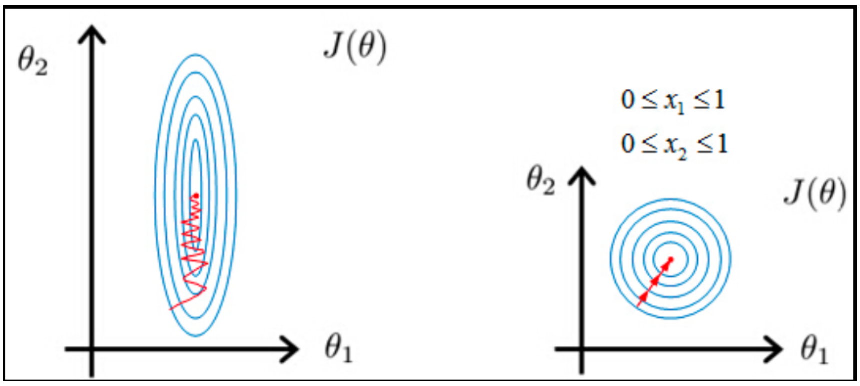


Figure 3: diagrammatic representation aimed at data range for algorithmic optimization. Copied from Andrew Ng

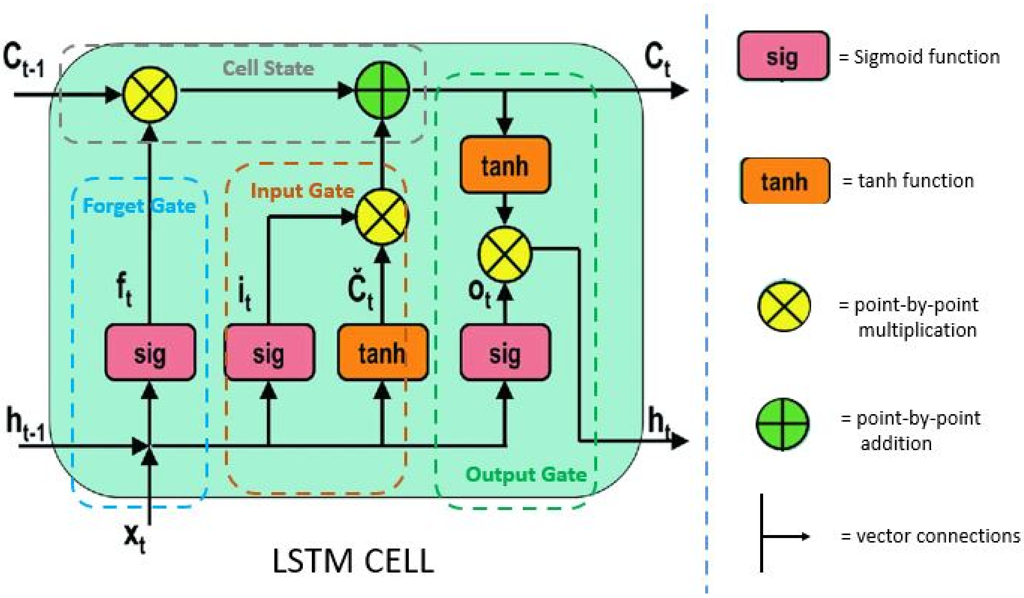


Figure 4: diagrammatic representation of an LSTM network. Copied from Singhal G.

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

Forget Gate Operation

Choosing whatever information from the cell state to discard is the first step for an LSTM. This occurs in the forget gate, hence a sigmoid layer. The sigmoid function receives data from the current input X(t) and the hidden state h(t-1). The values that Sigmoid produces range from 0 to 1. It draws a conclusion regarding the necessity of the old output's portion (by giving the output closer to 1). The cell will eventually use this value of f(t) for point-by-point multiplication. (Singhal G)

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Input Gate Operation

The next 2 steps decide how to update the cell status, the input gate does the subsequent processes. First, the second sigmoid function receives two arguments: the current state X(t) and the previously hidden state h(t-1). Transformed values range from 0 (important) to 1. The tanh function will then receive identical data from the hidden state and current state. The tanh operator will build a vector (C(t)) containing every possible value between -1 and 1 in order to control the network. The output values produced by the activation functions are prepared for multiplication on a point-by-point basis.

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

Cell State Operation

The input gate and forget gate have provided the network with sufficient information to provide an output. This output, however filtered, will be based on the cell state. A sigmoid layer will be run to determine which portions of the cell state will be output. The forget vector f multiplies the previous cell state C(t-1) (t). Values will be removed from the cell state if the result is 0. The network then executes point-by-point addition on the output value of the input vector i(t), updating the cell state and creating a new cell state C. (t).

Experiment Methodology

Primary experimental data is the main source of data in this paper. 3 LSTMs were programmed (code can be found in appendix A, heavily adapted from (Pathompong Yupensuk) due to the code's strong customizability in terms of the time frame and the throughout explanations within the comments) and were each trained with past bitcoin closing prices recorded on public datasets varying in time frequency. Their respective results in the testing phase were then recorded with the intention to study the influence different sampling periods held on predictive accuracy. Given the scarcity of secondary data sources available to address the study topic in this paper, I settled for an experimental methodology that gives researchers a significant amount of flexibility to control the independent variables.

Nonetheless, there are technical restrictions to this methodology; namely the exclusion of datasets surrounding the causation of the price volatility. Real-time and past datasets tracking bitcoin's value help the LSTM determine a trend in the consumers' demand which largely depends on the overall perception (many investors tend to sell when they notice a price drop and vice versa). However external factors such as cyberattacks, governmental regulations, and anomalous real-world occurrences (ex wars, pandemics) which strongly coincide with cryptocurrencies' volatility cannot be included as parts of the training or testing phases.

Time restrictions represent another limitation as it hindered the possibility of improving the LSTM models' capacity in fetching patterns.

The Datasets used

All the datasets used for the LSTMs' training are from Yahoo! Finance: A financial platform covering all available data on Bitcoin-USD price since 2014. The archive's peculiarity being the inclusion of minute-level time frequencies, too niche to interest investors, but representing a crucial feature for the sake of this research. The training set consisted of data from January 1st 2017 (a year in which bitcoin's price briefly reached a new all-time high of $19,783.06 (Smith)) to the 31st of July 2022, while the testing set consisted of data from 1 August 2022 to 31 August 2022. Three datasets were collected, each with a different sampling period; 1 minute, 1 hour and 1 day. It is worth noticing that the three utilized datasets contained values that include the lifting of the COVID-19 pandemic restrictions as well as the Ukrainian and inflation crisis, which are events characterized by their volatility and deviations from the regular behavior as well as structural breaks.

Processing the Datasets for use

The first step was to obtain the data through the installation of a library known as yfinance, as Yahoo Finance decommissioned its own official API following widespread misuse of data. Once the finance library was installed, its history function renders it possible to download historical data and convert it to a CSV file which is compatible with Python.

Next, the raw data was normalized in order for it to be processed by the LSTM. To improve the algorithm's performance, the data must be converted so that each value falls between 0 and 1 which is possible through Python's preprocessing.MinMaxScaler() function (Brownlee).

Lastly, the data structure must be modeled into an adjacency list (adj function). This saves a lot of space because values are only stored for the edges (ex. [dayAhigh,dayAlow]). The dataset will note a set of Open, High, Low, and Volume values for each day for the number of historical data points that were utilized to make bitcoin predictions.

Variables

Accuracy

The accuracy was measured on the predictions obtained throughout the testing phase only. The accuracy is evaluated by the correlation coefficient which was calculated by dividing the covariance by the product of the two variables' standard deviations; in this case, predicted results vs actual results. A prediction counts as correct if the LSTM could predict if the following day's price value of Bitcoin was increasing, decreasing, or remaining stagnant. The accuracy scales are based on three separately trained LSTMs, all of which evaluate the same 30-day testing period.

Preparing the three LSTMs

The three LSTMs programmed were identical in structure and were trained with data occurring throughout the same 2068 day time span; the only difference being the time frequencies tracked by the individual datasets. The homogeneousness of the code ensures that the patterns observed were not model-specific. A detail to make note of is the difference in the amount of information each of the artificial networks had available, with the minute-level frequencies dataset including 2,977,920 data points, the hourly frequency equalling 49,632, and the daily frequencies, of course, amounting to the least with 2068. The following flowchart (Figure 7) illustrates the structure of the LSTMs programmed: including the Bitcoin dataset being input the preparation of the data, the steps in processing, and the resulting final output layer.

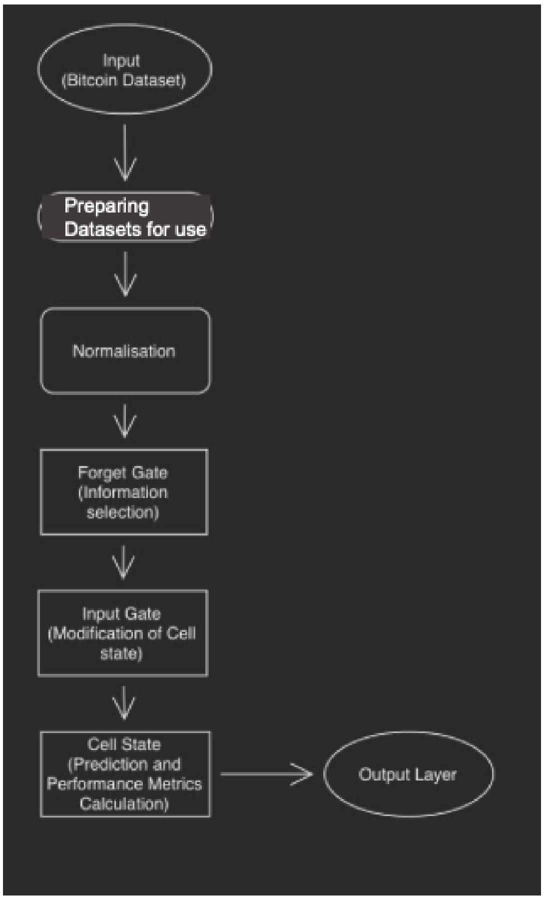


Figure 7: Visual representation of the LSTM model

The Experimental Procedure

Each of the three LSTMs was trained with a dataset spanning over the same 2068 days, each tracking a different time-frequency. Other than the data points, the training process remained relatively homogenous with each LSTM consisting of three layers and applying the same use dropout rate of 20% to combat overfitting during training.

The training results were recorded and respective accuracy was each calculated and archived. Once the model had completed the training, the test data was used to predict the price value and compare it with the real-world result by calculating mean squared error (MSE). The final results were then to be inverted with the Python built-in scale converter, so the prices were no longer scaled in the [0, 1] range.

Experiment Results

Tabular representation of relevant Statistics

Table 1 feature a summary of some of the most relevant descriptive statistics surrounding the 3 LSTM models' experimental results and real-time price values. The full set of raw data is available in Appendix B and the results of their respective calculation can be found in Appendix E.

	Real life	Minute-Level Frequencies	Hourly-Level Frequencies	Daily Frequencies
Mean/ Average	22493.84	22608.71	22593	22637.77
Minimum	19659	19603	19713	19983
Maximum	24434	24980	24868	24920
Standard Deviation	1488.80	1489.46	1268.10	1448.21
Number of True Positives	-	19.35% (6/31)	22.58% (7/31)	22.58%(7/31)
Number of True Negatives	-	29.03% (9/31)	41.94%(13/31)	29.03%(9/31)
False Positives	-	32.26% (10/31)	19.35% (6/31)	25.81%(8/31)
False Negatives	-	19.35% (6/10)	16.13% (5/31)	22.58%(7/31)
Accuracy ratio	-	48.39%(15/31)	64.52% (20/31)	51.61%(16/31)

Table 1: Descriptive statistics for predictive and the real-time bitcoin price values

Graphical Representation

To better understand the trends of classification within the performance, the data has been represented in pie charts which are depicted in Figures 8, 9 and 10.

Pie charts evidence the relationship between predictive accuracy and inaccuracy (calculated through the true accuracy ratio whose maximum achievable accuracy is 1 simulating an accurate prediction of a drop or rise). The percentages of the total are represented through the wedges of the graph.

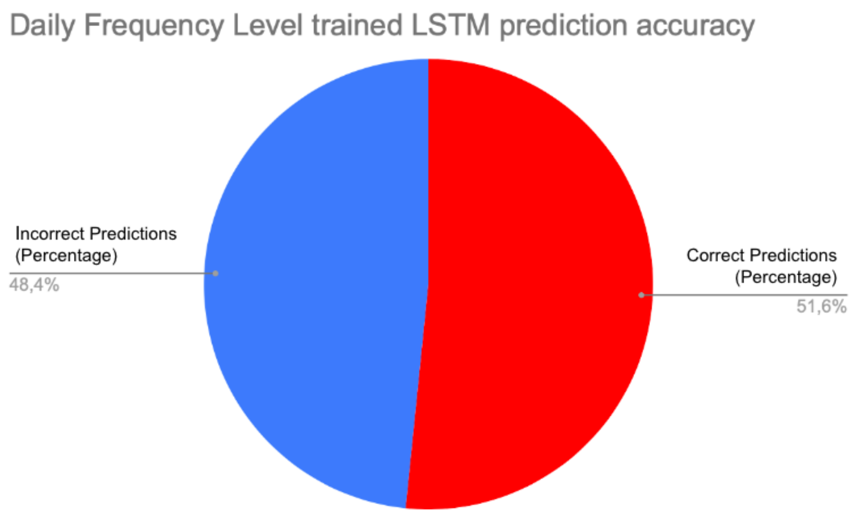


Figure 8: Daily Frequency Level trained LSTM prediction accuracy Pie chart

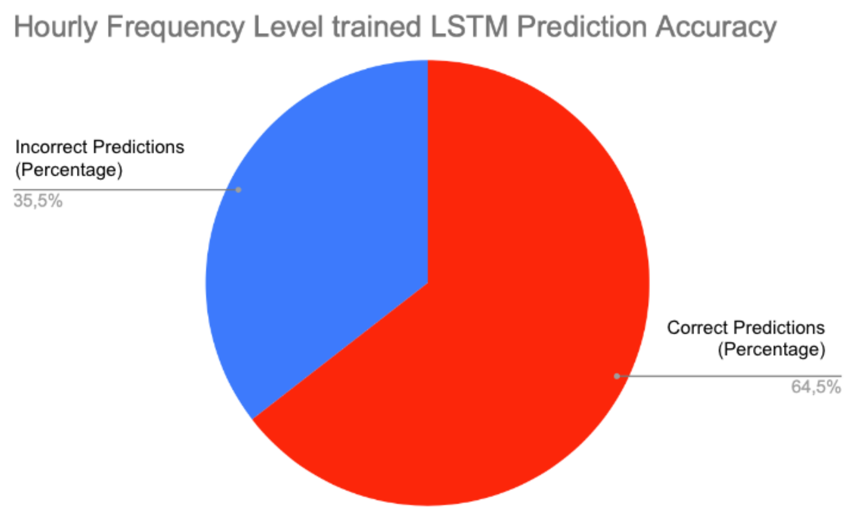


Figure 9: Hourly Frequency Level trained LSTM prediction accuracy Pie chart

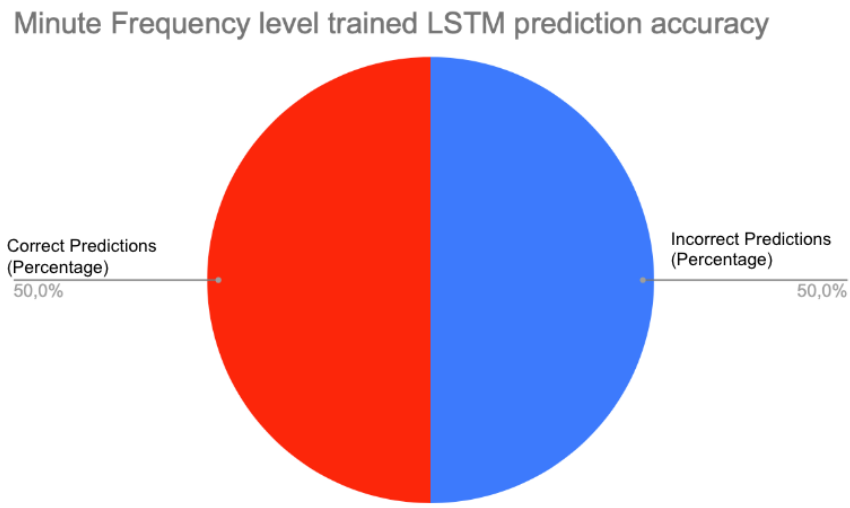


Figure 10: Minute Frequency Level trained LSTM prediction accuracy Pie chart

Data Analysis

Analyzing Real-Time data

To better understand the resultant predictions it is crucial to take note of the seasonality that was analyzed (See Appendix C for a graphical representation).

While volatility is a definite characterization of bitcoin, summer of 2022 stood out for its crypto market crash which was spurred by momentary de-risking from Wall Street due to many investors feeling pessimistic about the economy amid surging inflation (DeMatteo). It’s illustrated by August's bitcoin price value averaging 22493.84 USD in contrast to the all-year-round one being 29137.5 USD. It is therefore important to keep in mind that the lstm's accuracy for this testing sample in particular, could have been negatively influenced by third-party events such as the war, and shifting monetary policy in the various countries, this way resulting in some unpredictable behaviors. While not stationary, the price value remained relatively stable for the first 19 days, with the hugest drop occurring on day 10 to 11 counting 745 units. Subsequently to this period a loss of 2299 data points is recorded. As visible from the negative trendline alleged in Appendix C, the month of August was one which resulted in monetary loss.

Analyzing Time Frequencies

The results in Table 1 indicate that running an LSTM with different time frequencies throughout its training and testing phase strongly influences the predictions' accuracy. No pattern can be observed as the hourly frequency model, which was to act as an intermediary, performed best. This is demonstrated by its possession of the most approximate average with 22593 points against the actual one which measures 22493.84 and the 64.52% accuracy ratio which exceeds the daily (51.61%) and minute level one (48.39%). The causation could lie in finding a compromise in the amount of data fed into the neural network. Especially when considering the pre-programmed 20% drop rate which for the minute-level frequencies with 2,401,920 data-points could have been insufficient, in contrary to the sole 1618 data points the daily-frequency model had available, which in turn learnt to round up values excessively to make sense of the capital gaps occurring within a day's time span. The standard deviation indicates dispersion of distribution of values, similarly to the real-time data all predictive values are relatively scattered. Unlike the previous two criteria in which the hourly LSTM performed best, the predicted dispersion which assimilates the real world one the most was presented by the minute-level frequency LSTM. Overall, the outcomes seem to state that the more moderate the samples' intervals are, that is to say they're neither too narrow or distant, the more the algorithm's accuracy will improve.

Making sense of the differences in Accuracy of Minute, Monthly or Daily Predictions for Bitcoin Price

Remembering that neural networks represent extremely vast functions that increase their predictive capacity over time is necessary to make sense of the seemingly paradoxical relationship between time frequency based training and accuracy. The number of factors in a function that affect its input increases with the number of datapoints, therefore the LSTM that is frequency-based at the minute-level may produce accuracy from those that are frequency-based at the hourly or daily levels. The minute-level frequency function includes more parameters and is denoted for its complexity caused by the processing of excessive amounts of repetitive data, but this does not necessarily make it better; rather, it enhances the likelihood that it will perform better.

Time frequencies simply serve to infer data, neural networks do not necessarily require a large number of them to be effective (as seen with the hourly level frequency predictions which touch up a 64% accuracy rate). The goal of training is to teach the network how to create its own rules based on what it observes. It is therefore possible for a network to maintain accuracy even if the number of datapoints it has access to is reduced, provided that the given samples occur within the same time period and provide data accurate enough for usage. This is because whatever information one time frequency infers can still be suggested from a combination which varies in periodicity, granted the neural network processes the input from those kernels properly.

The lower accuracy of the daily frequencies (51%) suggests a significant reduction in data quantity in the subsampling stride causes accuracy to decline. The network's ability to receive entire information is impacted by a greater subsampling stride. Because broader intervals in time frequencies is less comprehensive and useful, the number of datapoints and a proximity in time capable of making the network understand what thought pattern is occuring, become important elements in accuracy when the information is less complete and therefore less comprehensive. In order to make up for this severe lack of information, more datapoints (and hence which were recorded within closer time periods) will be required. Nonetheless, it is also true that a network could experience larger accuracy dips due to data overflow as seen with minute level frequencies where the similarity index within various of consecutive points was so high the network seemed to struggle sorting out what was to be filtered.

In addition, the accuracy ratings changed according to the chosen evaluation method. Considering solely the numerical proximity between the predictions and actual bitcoin price values, then the hourly frequency based LSTM, followed by minute was best at accurately predicting and then daily. This outcome was evaluated through the similarity in trendlines (shown in Appendix D) and monetary mean.

The most common metric for classification for predictive tasks is accuracy, which quantifies the proportion of predictions in a dataset that match the actual predicted results.

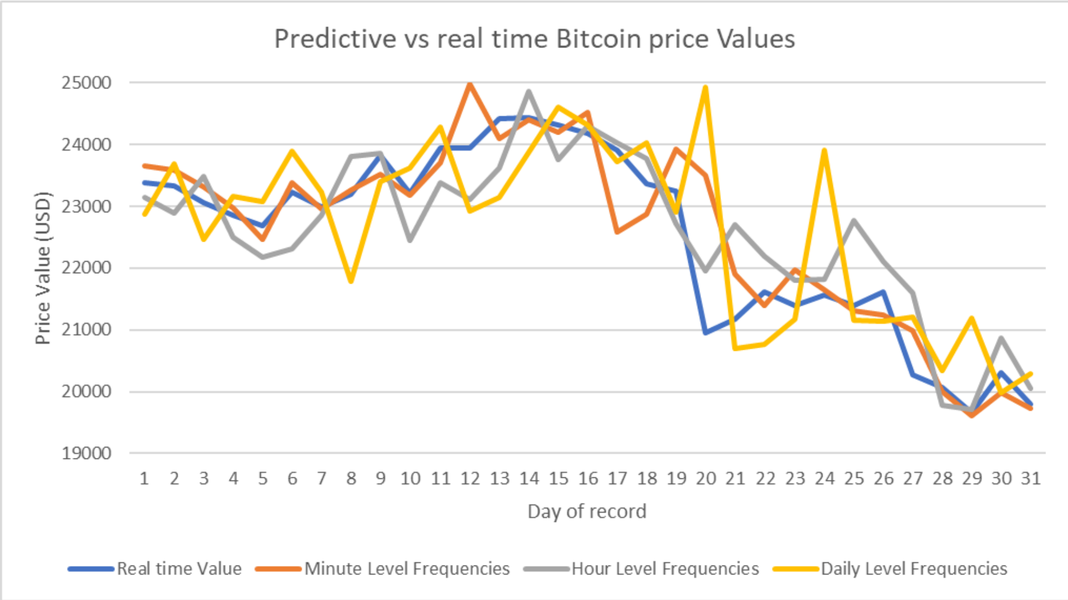
The formula states: **accuracy = correct predictions / all predictions.**

For that I used three error matrices (attached in appendix E) for each of the three LSTM models, each containing 31 predictions in a 2-class classification problem, hence whenever an increase or decrease in price value would occur on the following day. Interestingly, according to this classification parameter the hourly-frequencies based model retained the predictions with the most accuracy, followed by daily and with minute frequencies tailing behind. This methodology, while erroneous for not taking into account the predictions' numeric differences (as shown between the minute vs daily frequencies), is favored by data scientists as rather than relying on certain assumptions about the data like time series stationarity or the existence of a Date field, as their full potential is demonstrated when recognising complex patterns from enormous datasets as well as their capacity of narrowing down the relevant information (Kutzkov). This last point appeals to investors whose interest lies in knowing the crypto's surge and fall tendencies.

Comparing the Resultant Datasets

The LSTMs were found to have some degree of affinity for particular data sets throughout analysis. Despite not directly addressing the research topic, this is nonetheless interesting and has research implications, thus it deserves some consideration.

Pictured below is a line graph featuring all the predictive results for the three LSTM models, as well as the real-time bitcoin price values.



Predictive and real time Bitcoin price Values all included within one graph for the sake of comparison

All 3 LSTMs retained a sense of direction of the real time price values. This is demonstrated in both, a phase of relative stability, shown in the initial 14-day time period, where various points of the minute- and hour-level frequencies trained LSTMs collide with the real time value (emphasized within the 6-12 day time span). As well as more anomalous instances, demonstrated by the sudden change of pace occuring on the 15th and the sudden price drop on the 20th day, to which all the neural networks replied with the formation of a negative trend, successfully imitating the actual investors' thought pattern.

Nonetheless, the LSTM trained with hourly-level time frequencies performed best, followed by minute and then daily. Not only are the smaller time intervals capable of fetching a pattern closer to reality, but also lack ano c'èmalous data points. Meanwhile, the day-based predictive system is inclined to tend to sudden and far-fetched values compared to those which correspond to reality. These counter-current tendencies make up the biggest differential gaps seen on the 20th (with a 1850 data points difference) and on the 24th day (1386 data points difference-See Table 1).

Conclusion

In this research, the influence different time frequencies throughout the testing and training phases had on a LSTMs' performance were analyzed for the sake of adding onto the research. The outcomes are furnished with logical and mathematical justifications.

All three LSTM models' observed patterns differed across all datasets, suggesting that they are time frequency dependent rather than a fundamental property of LSTMs. The results show that moderate time intervals increase a LSTM's best performance. The reason being that neural networks working with particularly narrow breaks struggle managing and filtering the quantitative amounts of data transferred to them. Meanwhile, samples with track-records occuring on time periods too broad from one another confuse the algorithm which will have to make sense of inconsistent patterns resulting in predictions which are much more sudden in drops and rises and with a larger amount of anomalous points. However, while these predictions generally follow patterns, because neural networks are random, the consistency of these trends is potentially weakened.

The predictive accuracy, while strongly influenced by time frequencies, also depends on the interpretative approach taken. Though there is a general trend that more data points result in numerically more proximate results, and higher accuracy. Other methods solely base themselves on 2-class/parameter classification problems, identifying the correct predictions in terms of rises and drops. Both favored the hourly-level frequencies and swapped the other 2 models; this suggests both a trend containing an element of randomness, as well as a suggestion to operate with samples with more moderate intervals.

In order to iteratively enhance LSTMs performances, time-series focused neural network researchers and programmers can hopefully utilize this paper to help with the evaluation of choosing the data samples. This will hopefully lead to the discovery of further innovative approaches for the technological field and a reliable guidance for investors and analysts.

Bibliography

Graubard, Clay, and Andrew Eaddy. "Forecasting, Prediction Markets and the Age of Better Information." *CoinDesk: Bitcoin, Ethereum, Crypto News and Price Data*, 4 June 2022, www.coindesk.com/layer2/2022/06/04/forecasting-prediction-markets-and-the-age-of-better-information.

Bambrough, Billy. "Crypto Price Prediction: Bitcoin 'To Overtake' The Dollar By 2050 And Soar To $66,000 By The End Of 2021." *Forbes*, 2021, www.forbes.com/sites/billybambrough/2021/07/17/crypto-price-prediction-bitcoin-to-overtake-the-dollar-by-2050-and-soar-to-66000-by-the-end-of-2021/?sh=f278a5a4e9c8.

Morris, Chris. "Crypto volatility? Most owners will 'hodl' no matter what, says Deutsche Bank study." *Fortune*, 15 Feb. 2022, fortune.com/2022/02/15/crypto-volatility-owners-deutsche-bank.

DiLallo, Matthew. "Biggest Stock Market Crashes in History | The Motley Fool." *The Motley Fool*, 2022, www.fool.com/investing/stock-market/basics/crashes.

Selig, Jay. "What Is Machine Learning? A Definition." *Expert.ai*, 2022, www.expert.ai/blog/machine-learning-definition.

"What is LSTM? Introduction to Long Short Term Memory." *Intellipaat*, 2022, intellipaat.com/blog/what-is-lstm.

Vasudev, Rakshith. "Understanding and Calculating the number of Parameters in Convolution Neural Networks (CNNs)." *Medium*, 11 Feb. 2019, towardsdatascience.com/understanding-and-calculating-the-number-of-parameters-in-convolution-neural-networks-cnns-fc88790d530d.

Taylor and Francis. "Download Transformers for Machine Learning: A Deep Dive PDF by Taylor & Francis Group." *Search & download for 100.000.000++ of your favorites PDF*, 2022, pdfdrive.to/pdfs/transformers-for-machine-learning-a-deep-dive.

"Written Memories: Understanding, Deriving and Extending the LSTM - R2RT." *R2RT*, 2022, r2rt.com/written-memories-understanding-deriving-and-extending-the-lstm.html#what-rnns-can-do-choosing-the-time-step.

Sun, Y., and L. Zhu. "Multi-Input Convolutional Neural Network for Flower Grading." *Publishing Open Access research journals & papers | Hindawi*, 2017, www.hindawi.com/journals/jece/2017/9240407.

Ng, Andrew. www.andrewng.org. Accessed 25 Aug. 2022.

Strika, Luciano. "LSTM: How To Train Neural Networks to Write like Lovecraft." *Medium*, 24 June 2019,
towardsdatascience.com/lstm-how-to-train-neural-networks-to-write-like-lovecraft-e56e1165f514.

Zahran, Mohamed. "A hypothetical example of Multilayer Perceptron Network." *ResearchGate*, www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065.

DeMatteo, M. (n.d.). *Expert Bitcoin Price Predictions: From $10K to $28K in 2022, and Beyond*. NextAdvisor with TIME.
https://time.com/nextadvisor/investing/cryptocurrency/bitcoin-price-predictions/

Bloomberg - Are you a robot?(n.d.). Bloomberg - Are you a robot?
https://www.bloomberg.com/view/articles/2018-02-08/crypto-cynics-stand-to-profit-the-most

Schmidhofer, Juergen. *Our impact on the world's most valuable public companies (Google, Apple, Microsoft, Amazon, etc)*. 2017,
people.idsia.ch/~juergen/impact-on-most-valuable-companies.html.

Brownlee, Jason. "How to Scale Data for Long Short-Term Memory Networks in Python." *Machine Learning Mastery*, 2019,
machinelearningmastery.com/how-to-scale-data-for-long-short-term-memory-networks-in-python.

Dongens, Niklas. "A Guide to Recurrent Neural Networks: Understanding RNN and LSTM Networks." *Built In*, 2022, builtin.com/data-science/recurrent-neural-networks-and-lstm.

Ellis, Charles A. "A Systematic Approach for Explaining Time and Frequency Features Extracted by Convolutional Neural Networks From Raw Electroencephalography Data." *Frontiers*, 2022, www.frontiersin.org/articles/10.3389/fninf.2022.872035/full.

McCluskey, Megan. "How Crypto Investors Are Handling Plunging Prices." *Time*, 1 Feb. 2022, time.com/6141028/crypto-crash-investors.

Solomon, Martin. "Predictive Analytics Using Neural Networks." *LinkedIn: Log In or Sign Up*, 27 Apr. 2022,
www.linkedin.com/pulse/predictive-analytics-using-neural-networks-martin-solomon?trk=public_profile_article_view.

Kutzkov, Konstantin. "ARIMA vs Prophet vs LSTM for Time Series Prediction - neptune.ai." *neptune.ai*, 2022, neptune.ai/blog/arima-vs-prophet-vs-lstm.

Yupensuk, P. (2020, May 16). *Tuning LSTM to predict stock price in SET50 with lower than 5.5% error*. Medium.
https://medium.com/analytics-vidhya/tuning-lstm-to-predict-stock-price-in-set50-with-lower-than-5-5-error-e19b422304b7

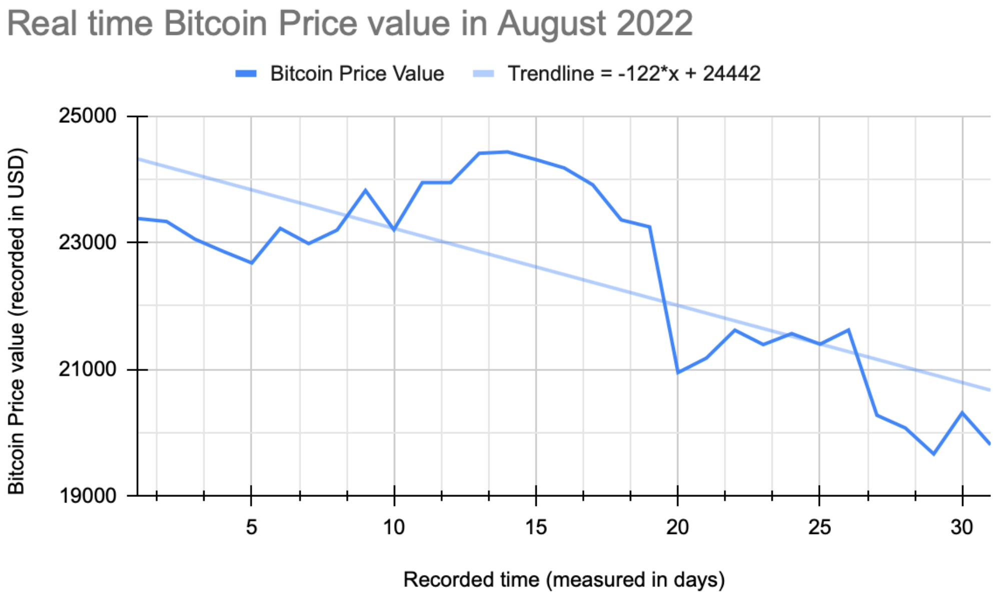
SAP Help Portal. (n.d.). SAP Help Portal. https://help.sap.com/docs/SAP_PREDICTIVE_ANALYTICS/41d1a6d4e7574e32b815f1cc87c00f42/9c144a376f004058b4e9fe56727359af.html?version=3.2&locale=en-US

Worldcoin. "Why Is Cryptocurrency Important? How Is It Revolutionizing the World? | Worldcoin." Worldcoin, 19 Sep. 2022, worldcoin.org/articles/why-is-cryptocurrency-important.

NOTE: I TOOK out appendix A and B

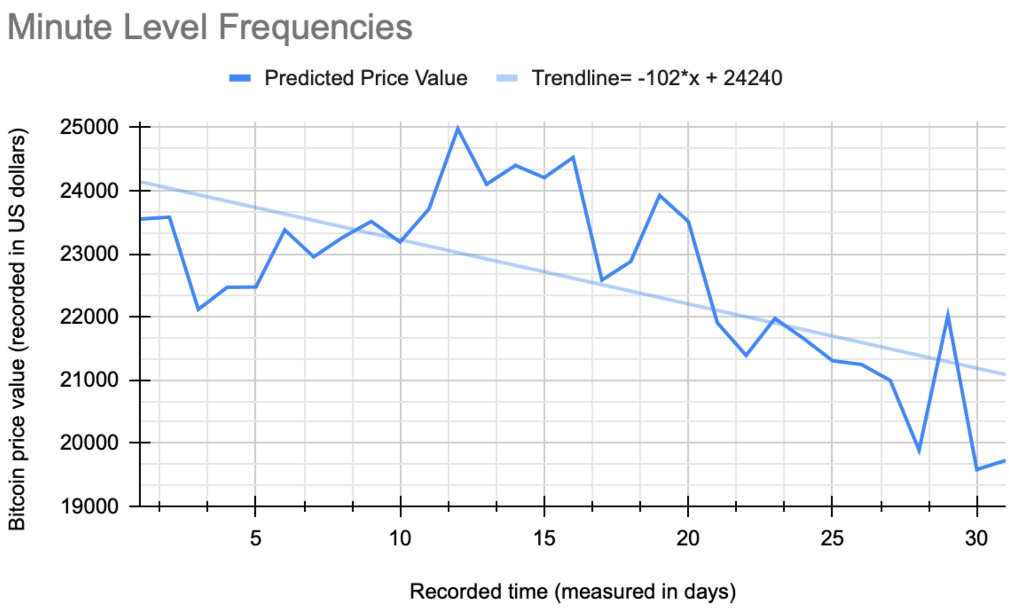
Appendix C

The following graphs depict the rough results of both prediction and real time data in line graphs

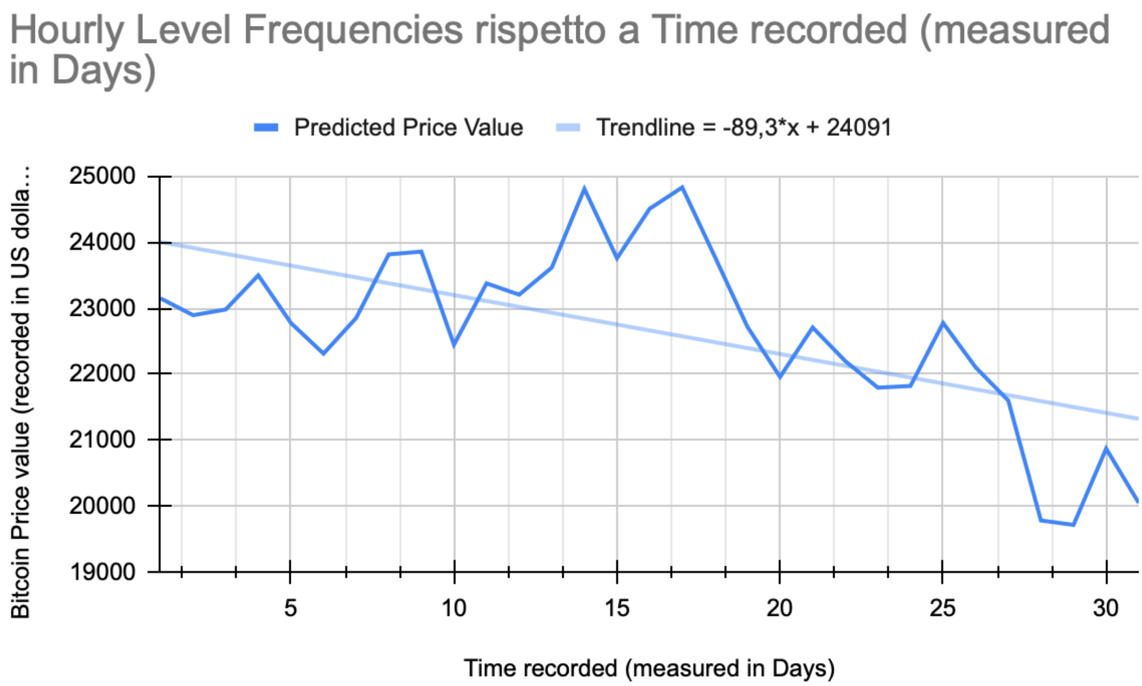


Line Graph illustrating real-time bitcoin Price value in August 2022

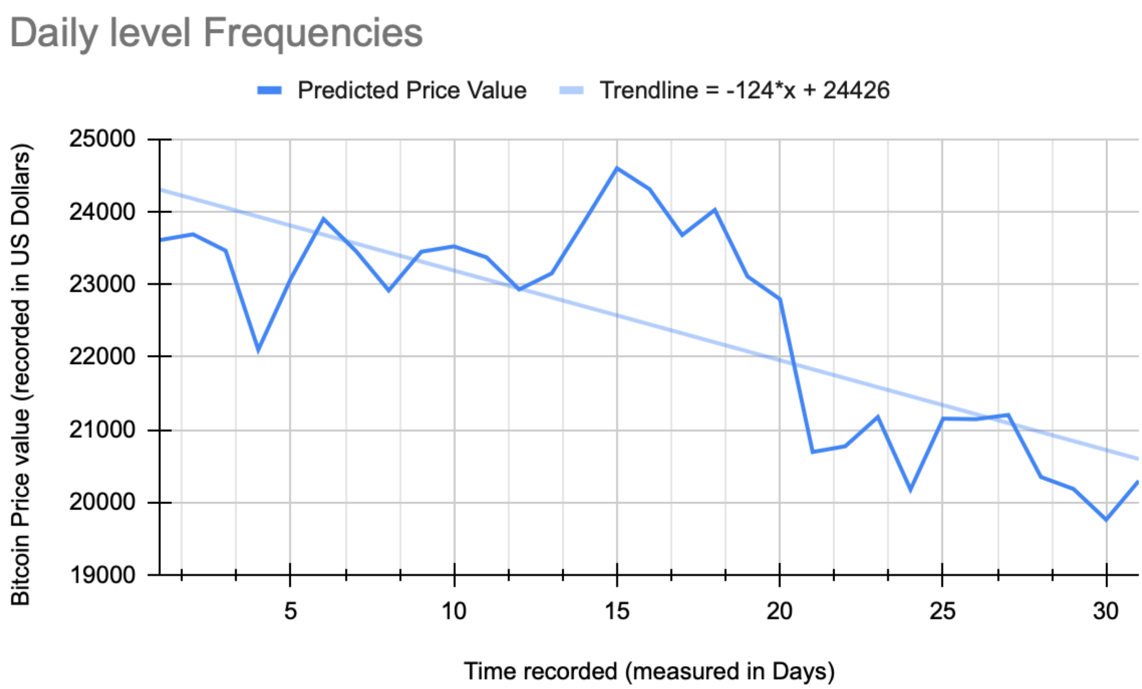
Appendix D



Minute Level Frequencies trained LSTM model's predictions



Hourly Level Frequencies trained LSTM model's predictions



Daily Level Frequencies trained LSTM model's predictions

Appendix E

	Real life	Minute-Level Frequencies	Hourly-Level Frequencies	Daily Frequencies
Mean/ Average	22493.84	22608.71	22593	22637.77
Minimum	19659	19603	19713	19983
Maximum	24434	24980	24868	24920
Standard Deviation	1488.8	1489.46	1268.1	1448.21
Skewness	-0.52	-0.62	-0.71	-0.38
Kurtosis	-1.02	-0.54	0.37	-1.2
Rises	12	11	12	18
Falls	19	20	19	13
Percentile representation of Ups	38.71%	35.48%	38.71%	58.06%
Percentile representation of Downs	61.29%	64.52%	61.29%	41.94%
Number of True Positives	-	19.35% (6/31)	22.58% (7/31)	22.58%(7/31)
Number of True Negatives	-	29.03% (9/31)	41.94%(13/31)	29.03%(9/31)
False Positives	-	32.26% (10/31)	19.35% (6/31)	25.81%(8/31)

False Negatives	-	19.35% (6/10)	16.13% (5/31)	22.58%(7/31)
Accuracy ratio	-	48.39%(15/31)	64.52% (20/31)	51.61%(16/31)

Tabular representation of various statistics surrounding the rough Data

Appendix F

The Error Matrix (Confusion Matrix), is a table that enables the comparison of the predicted value of the target variable with its actual value. The goal is to provide a representation of the precision of statistical classification to see how well the system performed. Each row corresponds to the observations in the real class, whereas each column corresponds to the observations in the anticipated class (SAP help Portal).

Total	Positive Targets Predicted	Negative Targets Predicted
Actual Positive Targets	Number of correctly predicted positive targets (True Positive =TP)	Number of actual positive targets that have been predicted negative (False Negative = FN)
Actual Negative Targets	Number of actual negative targets that have been predicted positive (False Positive = FP)	Number of correctly predicted negative targets (True Negative = TN)

Table Instructs on how to read the Error Matrix, Copied from "SAP help Portal"

Total (31)	Positive Targets Predicted	Negative Targets Predicted
Actual Positive Targets	19.35% (6/31)	19.35% (6/10)
Actual Negative Targets	32.26% (10/31)	29.03% (9/31)

Minute-Level Frequencies Error Matrix

Total (31)	Positive Targets Predicted	Negative Targets Predicted
Actual Positive Targets	22.58% (7/31)	16.13% (5/31)
Actual Negative Targets	19.35% (6/31)	41.94%(13/31)

Hourly-Level Frequencies Error Matrix

Total (31)	Positive Targets Predicted	Negative Targets Predicted
Actual Positive Targets	22.58%(7/31)	22.58%(7/31)
Actual Negative Targets	25.81%(8/31)	29.03%(9/31)

Daily-Level Frequencies Error Matrix