**BITCOIN PRICE PREDICTION USING LSTM NEURAL NETWORK**

### PROJECT OVERVIEW

In this project, deep learning model is built to predict the bitcoin price. The idea was to create a framework capable of analysing live data and helping stockholders to invest correctly. The programmer takes data from real-world and goes through set of reshaping data that gets should be given to algorithm of Machine learning as an input.

## Introduction

Bitcoin uses encryption and anonymity for cryptography. Bitcoins are entirely virtual, unlike traditional currencies. In order to forecast the price of Bitcoin, there is a need to implement the machine learning algorithm to forecast the future price.

Therefore, Long-Short Term Memory(LSTM) is implemented to predict the Bitcoin price to achieve maximum accuracy with lower error rate.

### Bitcoin

Bitcoin is a cryptographic cash that is used for advanced installments worldwide. It gains popularity by the end of year 2017 when its price has increased most significantly and was worth to 1600 US dollar for 1 bitcoin. Bitcoin is the most famous cryptographic money in the world and is exchanged in more than 40 trades that accept more than 30 different currencies worldwide. Although its market capitalization developed grew from 2013 to 2017 at an unfathomable pace, people are increasingly becoming aware of its existence. Although more than 18 million Bitcoins are available for use, there are only 6,674 Bitcoin ATMs around the globe. The concept of cryptocurrency has revolutionized the way we think about money. The steady growth in acceptance and widespread use has greatly increased its utility in real-world applications.

In contrast to fiat money, what makes a bitcoin special is that it is decentralized, and no single foundation or bank governs the bitcoin organization.

Since 2009, numerous cryptocurrencies have been invented but bitcoin was the first to be introduced as a cryptocurrency. Bitcoin is a combination of 4 key innovations:

* A ledger for public transactions.
* A collection of guidelines for validating independent transactions and currency issuance.
* A framework for achieving decentralized global consensus on legitimate.

The concept of E-Cash was first proposed by David Chaum in 1982. To perform online transaction, to reduce the use of paper currency and to do online payment directly.

* Genesis is the first ever block of bitcoin, that was mined in 2009.
* Digital secure Signatures.
* Not requiring the use of third party.
* Proof-of-Labor
* Hashing

Bitcoin behaves like a traditional currency. It can be sold, bought, and traded for other currencies at specialized currency exchange. With no physical existence, they are completely virtual.

#### Double Spending Problem:

A single digital token in digital cash schemes is only a file that can be duplicated and invested twice. In order to avoid double spending, a centralized trust party is always required. This is the reason why digital currency never succeed before.

Bitcoin has solved this double spending problem by making transactions transparent and decentralized distributed public ledger that reduces the effectiveness of central authority.

#### Properties of Blockchain

* **Decentralized:** peer-to-peer ledger of balances
* **Immutable:** Transactions are permanent and cannot be changed.
* **Fungible:** Bitcoin preserves its value equally.
* **Permission less:** Anyone can participate in the network.
* **Divisible:** Down to eight decimal sections.
* **Scarcity:** A limit of 21 million coins.
  + - 1. **How Bitcoin Works?**

Bitcoins are completely virtual unlike traditional currencies. In transactions coins are implicit that pass value. User own keys in bitcoin network that holds bitcoin ownership. To unlock value and spend it by moving it to new owner, user sign transactions with keys. In a digital wallet, keys are often stored. The only prerequisite to spend bitcoin is possession of keys, putting the control in the hands of each user.

#### Key Concepts

1. **Hashing**

A hash function is called digest of messages. It is often referred as automated fingerprint because there is no way any other string can be represent by this digest.

For hash formation, blockchain uses SHA256. It produces a 256- bit hash code.

1. **Public-key encryption**

The transaction is first encrypted with a private key when an information is transmitted in a blockchain and then exchanged on a network.

1. **Mining**

Mining is a method that produces a hash code satisfied by blockchain. The mining involves block number, nonce, previous block hash code, and some other data to generate hash code possess some low qualities and attain some difficult levels. Mining is a time-consuming process which is done by all the nodes present in the network.

* + - 1. **Bitcoin’s Generation**

The generation of bitcoin is based on the efforts of 10 minutes. A block is created and transmitted to the network after every 10 minutes. The node that mined that block would get 12.5 bitcoins and transaction fees. So, after every 10 minutes, it produces a new block and 12.5 new bitcoins. Currently, 17-17.5 crore bitcoins have been generated and there is more to be generated.

For any kind of currency, it is important to restrict its supply. The supply of bitcoin is limited to 21 million.

### Blockchain

Blockchain is a decentralized, distributed, immutable public transaction ledger that used cryptography to protect and block transaction.

In simple words, Blockchain is a time-stamped collection of permanent data records that are not just owned by any single individual but are maintained by clusters of computers. Using cryptographic principles, each of these data blocks is encrypted and bound to each other.

“Blockchain is distributed ledger” this means that it has no central power; it is decentralized.

**Process:** When transaction is performed by a user, transaction enters the network nodes. Then the recipient receives, verifies, and validates the transaction and then documents it. These transactions are immutable.

Blockchain has not implemented any new technology rather it uses existing technologies and combine them in such a way that makes blockchain effective.

* Ledger Accounting
* Cryptography
* Computer Network Technology

**Shaking the System.**

“Blockchain and cryptocurrencies are shaking the system.”

Industries that Blockchain will disrupt.

1. **Banking and Payments**

Bank plays an intermediate role. You first transfer it to the bank and then the bank transfers it to the particular person when you make the transaction to transfer money to another person.

This intermediate scheme is absolutely terminated by Blockchain. It operates on peer-to-peer software. In this scheme, there is no middleman.

1. **Supply Chain Management**

Consider an example of purchasing a product from a retail store. Now you do not know whether or not this item is authentic, did it really come from where it was made, how much time it took in whole supply chain systems. You are not going to know these things because you have no means of getting to know them. Blockchain provides you with knowledge of all these things so you can check the product quickly.

1. **Education Records**

Educational documents can easily be changed when an organization manages them centrally. They cannot be altered if these records are available on the blockchain.

#### How Blockchain Works?

Blockchain commonly uses hashing function. In hashing unction, a string of characters of any length is specified as an input in the hashing function and gives a fixed string length as an output.

Advantage of this function is that if you give the exact same input again, it will give you the same exact output but if there is some sort of alteration, even if a comma is added into a document, it will give you a completely different output.

Ledgers are maintained in a blockchain network on every device in an online network operated by in-centralized volunteers all over the world.

Each machine will have a copy of the ledger that shows the ownership of coins that is visible and known to all. He/She would simply announce to the computer network his/her intention to submit coins to someone else anytime a person decide to make a transaction. Upon hearing the announcement, all those machines will change their inner ledger by deducting coins from her and adding coins to the other guy.

With the help of hashing function, the identities of people are disguised on a public ledger. The identity information of each person is thus reduced by anonymizing the participant to an unrecognizable 20 digits. The input needed to produce the code is only known to the owner.

In order to ensure that everything begins from the same version of the transaction, the second thing is that ledgers are easily compared between computers. Blockchain is transparent and publicly accessible and user information is available to the public. The blockchain’s goal is to make transactions transparent and maintain the user’s personal anonymity.

#### Creation of Block

When a transaction made in a network, it floods in that network. A node where transaction is created sends it to the other 8 nodes in a network. Those 8 nodes receive the transaction and verify it. After verification, node stored transactions in its pool of transactions and then sent it to other 8 additional nodes. This flow of information Is called flooding/gossip protocol/broadcasting.

Once data is reached to all nodes in a network, then each node will pick that transaction and then creates its block. The development of block involves presence of transaction in a transaction pool of code, block timestamp, previous block hash, nonce number, and some other data.

Now the hash generation process called proof-of-work must be performed by every node. This hash code is the real requirement of blockchain which must have some blockchain-defined attributes.

Blockchain imposes some restrictions and set different levels to generate a hash code for each block. If block generation takes time more than 10 minutes, then level of difficulty is increased and if takes time less than 10 minutes then difficulty level is increased. In a network, a node that executes the block first makes an announcement and all other nodes interrupt their execution process. The block will be sent to all other nodes in a network by the one who has generated the block. The block is obtained and verifies by all other nodes. If the block is valid, they will record it in their ledger. In addition to that, other network transactions often take place such that each node generates its block, and the transactions are received at each node. It a continuous process.

#### Formation of Blockchain

Hash code of latest block in used by the next block, block number, nonce number to form a chain in blockchain. In 1st block, previous hash code used is always 0. It is also called genesis block. The next block would use the hash code of 1st block as the previous hash and so on.

If the blockchain is created, it is not possible to modify the information in any of the previous block to produce its fresh hash code.

Suppose a blockchain of 5 blocks in it. Each block has already been created and has some details. Block number, nonce number, 0 as previous hash and some other data are available in 1st block. The hash of the 1st block is then used in 2nd block as a previous hash attached with other relevant details. With the other block, same process goes on. All blocks are now completely generated and satisfied by blockchain. Now you want to change some data in block 2 to create its new hash. The mining process will be unable to find new hash, so it is difficult to do so. The next block will not accept this even if the new hash is created. It is for the reason that blockchain is immutable.

So, if you want to modify some block in blockchain, by making some modifications, you want to do mining and then transmit it to the network. Firstly, it will not be approved by the network. Secondly the mining of that block would render invalid the next block Then the next block needs to be verified too.

Changing any new block would require 51% of the network capacity but it will still be almost impossible to do so.

1. **Design and Methodology**

Design and Methodology section discuss the details of the tools and technologies, design steps of Bitcoin Price Prediction Project and the methodology adopted during the implementation of the project. This includes the simulation design, and algorithms of Machine Learning model.

* 1. **Tools and Technologies**

The technology and tools in this chapter apply primarily to applications that can be used to build or support the Bitcoin Price Prediction App. We think of technology today only as computer related tools, but technology includes all tools that improve instruction delivery, and these tools have evolved and will evolve over time.

* + 1. **Technology**

Technology is the sum of techniques, skills, methods, and processes used to manufacture products or services or to achieve goals, such as science study. Awareness of procedures, methods, and the like can be technology, or it can be incorporated in machines to enable operation without comprehensive understanding of their operations. Systems (e.g. machines) are referred to as technology systems or technological systems which apply technology by taking an input, adjusting it according to the use of the system, and then producing outcome.

* + - 1. **Time-Series Model**

Time series is a dynamic field of research that has attracted research community attention over the last few decades. The main aim of time series modelling is to carefully collect and rigorously study the past observations of a time series to develop an appropriate model which describes the inherent structure of the series. This model is then used for the sequence to create future values i.e. to make predictions. Therefore, time series prediction can be called as the “Act of forecasting the future by understanding the past”. In order to match an effective model for the underlying time series, due to the critical importance of time series forecasting in various practical fields such as industry, economics, finance, science, and engineering, etc. proper care should be taken. It is clear that an accurate time series forecast relies on a suitable model fitting. Over many years, researchers have made a great deal of effort to develop effective models to improve the accuracy of forecasting. As a result, in literature, different significant time series forecasting models have been evolved.

* + - 1. **Predictions**

For new data, predictions are made when the actual result may not be known until some future date. The future is expected but almost always all past results are handled equally. Maybe to solve the idea of “concept drift” with some very minor temporal dynamics, such as only using the last year observations rather than all available data.

A significance difference in predictions is that the future is totally unavailable and only from what has already occurred must be predicted.

* + - 1. **Time Series Forecasting**

A time series dataset is different. The time series introduces an explicit dependency of order between observations: a dimension of time. This additional aspect is both a weakness and mechanism. That provides a source of additional data. A set of observations taken sequentially in time is a time series. Forecasting requires taking historical data-fit models and using them to estimate future observations.

In general, the purpose of time series analysis is twofold: to understand or model the stochastic processes that give rise to an observed series and to estimate or to predict the future values of a series based on the history of that series.

The capability of the forecasting model of a time series is determined by its accuracy in predicting the future. This is also at the cost of being able to understand why a particular prediction was made, confidence intervals, and the underlying causes behind the issues are much better understood.

* + 1. **Tools**
       1. **Python**

In the late 1980s, ‘Guido Van Rossum’ launched Python and the first version was released in 1991. It is a high level-language used for widely useful programming and is used on various frameworks/platforms such as web development(server-side), software development and scripting of systems.

The python is one of the most renowned artificial intelligence programming languages and thus appreciates the countless number of custom functional libraries generated by its more popular network.

Despite the fact that the presentation of related languages, such as Python, is worse for many computational tasks compared to low-level programming languages, extension libraries such as “NumPy”, “SciPy” have been created that extend “Fortran” and “C” inferior layer used for faster and vectored functions on multi-dimensional screens.

The majority of the types of algorithms of the Machine Learning(ML) and Deep Learning are used in this project, therefore, the language that is most popular for these types of applications is the software of this project. The reason why Python is popular for applications for machine learning is that it has a basic syntax that resembles the English language, so it is easier to write programs with fewer lines compared to some other programming languages for complex calculations.

Python has a broad variety of libraries for data loading, visualizations, statistic, natural language processing, image processing, etc. It also has machine learning libraries, with Scikit-learn, TensorFlow, and Theano are being the most common, open-source and available.

* + - 1. **Google Collaboratory**

Google Collaboratory or ‘Collab’ is a free cloud service hosted by Google to support research on Machine Learning and Artificial Intelligence, where the obstacle to learning and progress is always the need for enormous computing resources. Colab enable anyone through the browser to write and execute arbitrary python code, and is particularly well-suited to machine learning, data analysis and education. More technically, Colab is hosted Jupyter notebook that needs no configuration to use, thus providing free access to GPUs, including device resources.

* **Benefits of Google Colab:**

The Colab is fairly robust in its setup, in addition to being easy to use, and does much if heavy lifting for developers.

* Support Python 2.7 and Python 3.7
* Free GPU acceleration
* Pre-installed libraries: All major Python libraries like TensorFlow, Scikit-learn, Matplotlib among many others are pre-installed and ready to be imported.
* Built on top of Jupyter Notebook
* Collaboration feature (works with team just like Google Docs): Google Colab allows developers to use and share Jupyter notebook among each other without having to download, install, run, or anything other than a browser.
* Supports bash commands.
* Google Colab notebooks are stored on the drive.
  + - 1. **Android Studio:**

Android studio is the official Integrated Development Environment (IDE) based in the IntelliJ Idea for the development of android applications. In Android Studio, individual development requires a minimum single section of source code records and resource files. Module classes adopts:

* Android application modules
* Library modules
* Google App engine modules

Android Studio displays your work documents in the Android Project view, as a matter of course. This view is sorted out by modules to provide easy access to the main source files of your project.

* **Attributes of Android:**
* A versatile Gradle-Based build framework
* A simple property rich emulator
* The integrated environment in which all android devices can be created
* Apply push code and asset changes to your operating function without any resuming the application
* Wide instruments and context for research
* Lint instruments for seizing execution, compatibility, class harmony, and various problems.
  1. **Design Steps**

The Design steps in this chapter define the steps followed by the students to build the application or the flow carried out to develop Bitcoin Price Prediction Project. This section consists sequence of steps. The steps are written sequentially, but we have jumped back and forth for our project. The Project is built by following high-level steps given below:

Step 1. Define Problem

Step 2. Prepare Data

Step 3. Evaluate Model

Figure 1 Design Steps of The Project

* + 1. **Define Problem**

This step is concerned with learning enough about the project in order to select the prediction task framing. Is it classification or regression, for instance, or some other higher-order problem type?

It includes gathering the information that is believed to be helpful in making a forecast clearly identifying the shape the forecast would take. This step also includes taking a close look at data, as well as perhaps using summary statistics and data visualization to explore the data.

* + 1. **Prepare Data**

Data Preparation is the process of readying data for the training, testing, and implementation of an algorithm in machine learning. Prior to being used to match and validate a machine learning model, the raw data must be pre-processed. This phase is known as “Data Preparation”, in a predictive modelling project, although it goes by many other names such as “data wrangling”, “data washing”, “data pre-processing”, and “feature engineering”.

It is a multi-step process involving collection of data, cleaning and pre-processing, feature engineering, feature scaling, and labelling. Such measures play an important role in machine learning’s overall efficiency as they build on each other to ensure that a model meets standards.

**2.2.2.1. Data Collection:** The process of identifying or constructing sufficient data to be used for training a machine learning model is data collection.

**2.2.2.2. Data Pre-Processing:** The act of cleaning and preparing your data for training is data pre-processing. This involves the arrangement and formatting, standardization, and handling of missing information.

**2.2.2.3. Feature Engineering:** Feature Engineering is the method for developing data enhancement features. Feature Engineering enables you to define the most important information in your dataset and to get most out of it using domain expertise.

**2.2.2.4. Data Labelling:** Data labelling is the key component of machine learning data preparation since it determines from which parts of the data the model can learn. Data is divided into two sets: one for the purpose of training the algorithm and other for evaluation. In order to ensure adequate testing, non-overlapping subset of the data for training and assessment sets were selected.

**2.2.3. Evaluate Model**

The evaluation of machine learning models on the dataset deals with this step. We build a robust test harness that is used to validate our model so that we can trust the results we get.

This includes tasks such as choosing performance metrics to assess a model’s ability, and a resampling technique to divide the data into training and test sets to simulate how to use the final model.

A single train-test split of the data can be performed for fast estimates of model performance, or for a very large dataset. This phase also includes tasks to make the most of well-performing models, such as hyperparameter tuning.

* 1. **Methodology**

the methodology explains the milestones to be reached by developer to achieve the mentioned target. The approach illustrates the system’s overall preparation schedule. It is a guide and gives them clues on what needs to be achieved first, so that the criteria for post-steps are met.

Data Collection

Data Normalization

Feeding Input to Model

Training Algorithm

Evaluation Metrics

Figure 2 Methodology of The Project

The methodology of our project is carried in such a way that the first phase is the data collection. We have acquired real-time bitcoin values from Cryptocompare through an API. After acquiring this time series daily recorded data at different instances of time, it must be normalized and smoothened. For this, we have implemented the minmax technique of normalization. After this, data is smoothened over the entire time span. The next step is to select the parameters from an available feature of data to be supplied to the network for prediction. After feature selection of features, the sample inputs will be fed to the model. It is possible to consider the variations in bitcoin values just as pattern. The pattern can be either grown up, down or to stay within a certain margin of the cost of the previous day. The next available option is the number of layers and per layer, the number of neurons. The model, therefore, will perform a pattern helped LSTM neural networks to correctly predict the bitcoin value. It is possible to achieve accuracy with the model after the final prediction.

* + 1. **Data Collection**

Using a rest API call, we collected data from CryptoCompare website. The API returns data at time interval of 15 minutes from May,2019 to present day. The obtained data is loaded into a data frame.

Figure 3 Process Flow of Data Collection

The stored data contains the features which are tabulated below:

|  |  |
| --- | --- |
| Features | Description |
| Open Price | It is market open price for currency of that day. |
| Close Price | It is the market close price for currency for that day. |
| High Price | It is the highest price of currency for that day. |
| Low Price | It is the lowest price for currency of that day. |
| Volume | The volume of currency that is being traded for that day. |

Table 1 Features of Data

* + - 1. **CryptoCompare API**

A record of all cryptocurrencies available in the market are held by CryptoCompare website. They are keeping the list of all the transactions by documenting the quantity of circulating the coins. The number of coins that have been exchanged till the last minute. They have updated their records on an ongoing basis as they receive feeds from different cryptocurrency exchanges. CryptoCompare provides with real-time as well as historical data for Bitcoin prices.

CryptoCompare has established the core concepts to its data services; they are:

1. Ease of integration
2. Broad, rich datasets
3. Exceptional Quality

CryptoCompare has massive array of market knowledge for digital assets, bundled into 73+ powerful API endpoints including real-time via web stock and API services, enables users to manage multiple exchange market data feeds from a single, clean, and standardized source.

**What’s Available:** Trade data for all exchange markets, level 2 order book data, exchange tickets, CryptoCompare Aggregate Index, OHLC Candles, full volume for all markets and all exchanges and full volume for all Top Tier exchange markets( in the new CryptoCompare Exchange Benchmark, rated AA, A or B).

* + 1. **Data Normalization**

The next step is the normalization of data. Basically, we perform this step to maintain consistency i.e. reduce or eliminate the duplicate data, negligible points, and other redundancies. As part of data preparation for machine learning, normalization is used technique.

The aim of standardization is to adjust the numeric column values in the dataset to use a common scale, without distorting variations in the value ranges or losing information. For some algorithms, normalization is also necessary to model correctly.

By generating new values that preserve the general distribution and ratios in the source data, while maintaining values within a scale applied to all numeric columns used in the model, normalization avoids such kinds of issues.

For normalizing our data, we have used min-max normalization technique to transform the data between 0 and 1.

* + - 1. **Min-Max Normalization**

Min-Max normalization technique is the linear transformation of the original unstructured data. It scales the data from 0 to 1. Min-max normalization is the most common normalization technique. For each feature, he minimum value of that feature is transformed into 0, the maximum value is transformed into 1, and every other value is converted into a decimal between 0 and 1.

* + 1. **Training Algorithm**

The training method for an ML model includes the provision of training the data to learn from an ML algorithm (that is, the learning algorithm). The training data, known as target or target attribute, must contain the correct response.

In the training data, the learning algorithm seeks patterns that map the attributes of input data to the target (the response you want to predict), and it outputs an ML model that captures these patterns.

The ML model can be used to get predictions about new data that you do not know the target for.

You need to define the following in order to train the ML model:

1. Source of input training data source
2. Name of the data attribute that contains the projected goals
3. Necessary instructions on data transformation
4. Parameters of training to monitor the learning algorithm
   * + 1. **Long Short-Term Memory (LSTM) Model**

A type of Recurrent Neural Network, Long Short-Term Memory (LSTM) was designed by Hochreiter and Schmidhuber. LSTM networks are extension of Recurrent Neural networks, primarily implemented to deal with situations where RNN fails. Speaking of RNN, it is a network that operates on the present input by taking the previous output(feedback) into account and storing it for a limited period of time in its memory.

However, there are disadvantages to RNNs. First, for a longer period of time, it fails to store the information. RNNs are unable to handle these long-term dependencies. Second, the aspect of the meaning that needs to be carried forward and how much of the history need to be forgotten is not better managed. Other problems with RNNs are the exploding and disappearing gradients that occur by back tracking during a network’s training phase.

Thus, LSTM were introduced. It has been built such that the issue of the vanishing gradient is almost entirely eliminated, while the model of training is left unchanged.

* **Structure of LSTM**

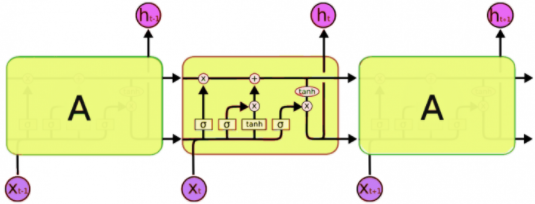


Figure 4 Structure of LSTM Model

A typical LSTM consists of various blocks of memory called cells. There are two states that are being shifted to the next cell: the state of the cell and the hidden state. The memory blocks are responsible for recalling stuff, and three key mechanisms, called gates, are used to control this memory. Each cell is discussed below in detail:

**1. Forget Gate:**

A forget gate is responsible for removing details from the state of the cell. By multiplying a filter, the information that is no longer is needed for the LSTM to understand things or information that is of less value is removed. This is required for efficiency of the LSTM network to be optimized.

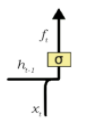


Figure 5 Forget Gate

This gate takes h\_t-1 and x\_t as inputs.

The hidden state of the previous cell or the output of the previous cell is h\_t-1 and the input at that specific time point is x\_t. the weight matrices multiply the given inputs, and a bias is applied. The sigmoid function is subsequently added to this value. This sigmoid function outputs a vector corresponding to each number in the cell state, with values varying between 0 and 1. The sigmoid function is essentially responsible for determining which values to preserve and which to discard.

Equation 2: Forget Gate

If a ‘0’ is output in the cell state for a specific value, it means that the forget gate needs the cell state to fully forget that piece of information. Likewise, a ‘1’ implies that the forget gate wishes to recall the whole piece of data. This output vector from the sigmoid function is multiplied by the cell state.

**Input Gate:**

For the addition of information to the cell state, the input gate is responsible. As seen from the diagram below, this addition of information is essentially a three-step procedure.

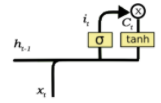


Figure 6 Input Gate

1. By involving a sigmoid function, controlling what values need to be added to the cell state. Basically, this is similar to the forgotten gate and acts as a filter for all h\_t-1 and x\_t data.
2. Creating a vector that includes all possible values that can be added to the cell state . Using tanh function, which outputs values from -1 to +1.
3. Multiply the value of the regularity filter (sigmoid) to the generated vector (tanh function) and then, through the addition operation, add this useful information to the cell state.

Equation 3: Input Gate Layer Equation

**Output Gate:**

The output gate performs this job of selecting useful information from the current cell state and presenting it as an output. Here is its composition:

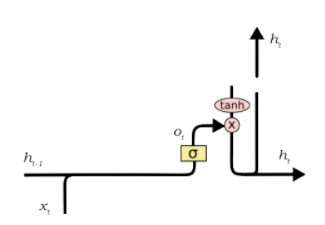


Figure 7 Output Gate

The operation of an output gate can be broken down to three steps.

After applying the tanh function to the cell state, to construct a vector, scaling the values to a range of -1 to +1.

Make a filter using the h\_t-1 and x\_t values so that the values that need to be output form the vector generated above can be managed. Once again, this filter employs a sigmoid function.

Multiply the value of this regulatory filter to the vector produced in step 1 and send it to out to the next cell state as an output, as well as to the hidden state.

Equation 4: Equation of Output Gate Layer

* + - 1. **Dropout Regularization**

Dropouts are used in making the neurons more robust and hence allowing them to predict the trend without focusing on any one neuron.

Deep Neural Networks are very efficient machine learning systems with a wide number of parameters. Overfitting, however, is a serious issue for such networks. Large networks are often slow to use, making overfitting difficult to deal with by integrating the predictions at test time of several different large neural networks. Dropout is a technique used to resolve this problem. During training, the main concept is to randomly drop units from the neural network. This avoids too much co-adapting by units. Dropout samples from an exponential number of different ‘thinned’ networks during training. At test time, by simply using a single non-thinned network with smaller weights, it is simple to estimate the effect of combining the predictions of all these thinned networks. This greatly decreases overfitting and provides substantial advantages over other methods of regularization.

* + - 1. **Optimization**

The most important ingredient in the recipe of machine learning algorithms is optimization. This starts with specifying some type of loss function/ cost function and ends with using one or other optimization routine to minimize it. The choice of optimization algorithm will make a difference in hours or days between having a good accuracy. Optimization applications are infinite and are a widely studied subject in both industry and academia.

One of the main hyperparameters undergoing optimization is the learning rate. The learning rate determines if any parts of the data will be missed by the model. If the learning rate is high, more subtle aspect of data will be missed by the model. If the learning rate is high, more subtle aspects of the data could be overlooked by the model. If it is tiny, then for real-world applications, it is desirable. The rate of a learning has a major impact on SGD. It can be difficult to set the correct value of learning rate. Adaptive approaches were immediately proposed for this tuning.

In DNNs, the adaptive variants of SGD have been used extensively. The exponential averaging is used by methods such as AdaDelta, RMSProp, and Adam to provide efficient updates and simplify the calculation.

* + - * 1. **Adam Optimizer**

The Adam optimization algorithm is an extension of stochastic gradient descent that has recently seen wider use in computer vision and natural language processing for deep learning applications.

Adam is an algorithm for optimization that can be used to adjust iterative network weights based on training data instead of the classical stochastic gradient descent technique. Adam optimizer combines the advantages of Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp).

Adam also uses the average of the second moment of the gradients (the uncentered variance) instead of adapting the parameter learning rates based on the average first moment(the mean) as in RMSProp.

In particular, an exponential moving average of the gradient and squared is determined by algorithm and the parameters beta1 and beta2 control the decay rates of these moving averages.

* + 1. **Evaluation Metric**

Mean Absolute Error (MAE) is used as evaluation metric. The reason behind choosing MAE over Root Mean Squared Error (RMSE) is that MAE is more interpretable. RMSE does not describe average error alone and hence is much more difficult to understand. Since we want model to be readily explained even to the non-technical audience, MAE looks like a better choice.

* + - 1. **Mean Absolute Error (MAE)**

Mean Absolute Error is a model evaluation metric used with regression models. The mean absolute error of a model with respect to a test set is the mean of the absolute values of the individual prediction errors on over all instances in the test set. Each prediction error is the difference between the true value and predicted value for the instances. [13]

Equation 5: Equation for Mean Absolute Error

1. **Results and Discussions**

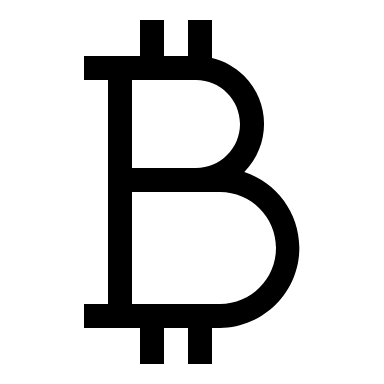
The outcome of this project, that includes the graphs, figures and present detailed discussions and analysis of the obtained result.

* 1. **Experimental Results of Model**

The model is built using sequential time series algorithm LSTM of RNN. The real-time data was collected from CryptoCompare website through an API. The data was first normalized so that model can better learn during training. For normalization, we choose min-max scalar that scale the data between 0 and 1. The data was then split into two sets: namely training dataset and testing dataset. the training dataset contains 80% of the data and the remaining 20% data was used as testing dataset. After splitting the data, the data was reshaped into arrays so that they can be fed into the model as an input. The neural network is comprised of a LSTM layer followed by 20% Dropout layer and Dense layer with linear activation function. The model is compiled using Adam as optimizer and Mean Squared Error as loss function. The model was trained on 100 epochs with a batch size of 32

.

**Dataset Collection**



**Dataset Pre-Processing**

Data Cleaning

Formatting into Date and Time Format

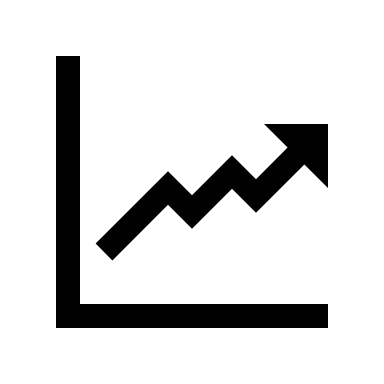
Selecting Target Feature

Normalization

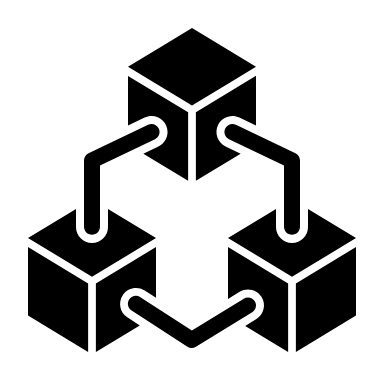


**Training Model**

**Predictions**



**Data Splitting**



**Accuracy**

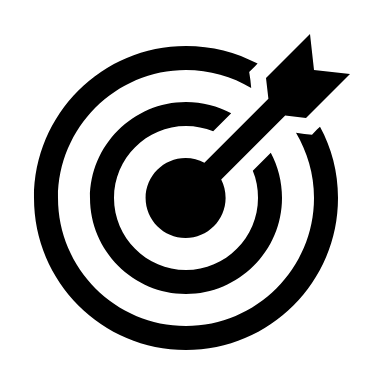


Figure 8 Workflow Diagram of System

* + 1. **Summary of Model**

The model is implemented using LSTM model. It is built on 3-stacked layers with a linear activation function. Each layer contains 500 neurons and dropout value as 0.2. The input data is split into the first 80% for training and last 20% for testing. We train the machine using data obtained from 400 days before the present day.

A screenshot of a cell phone

Description automatically generated

Figure 9 Summary of Model

* + 1. **Data Visualization**

From the figure 10 we can observe that from April 2019 the price rapidly goes up, in July 2019 it can be seen Bitcoin price has attained the highest peek i.e. above 12000 USD. After that, the curve slowly goes down up down and from January 2020 to April 2020 there is a clear dip in graph the lockdown period has also affected the Bitcoin price. In April 2020, the price again goes up. From April the price is continuously going up with slight fluctuations



Figure 10 Data Visualization

* + 1. **Performance Evaluation**

The performance is evaluated using MAE evaluation metrics and loss is evaluated using MSE.

A close up of a map

Description automatically generated

Figure 11 Loss during Training and Testing

Figure 11 shows that from 5th epoch the loss during training and testing of data is reduced to zero.

A close up of a map

Description automatically generated

Figure 12 Mean Absolute Error

Figure 12 represents the Mean Absolute Error during training and testing of data. In the above figure we can see the error rate is reduced below 0.05 after 5 epochs which show a good accuracy.

* + 1. **Resulting Graph**

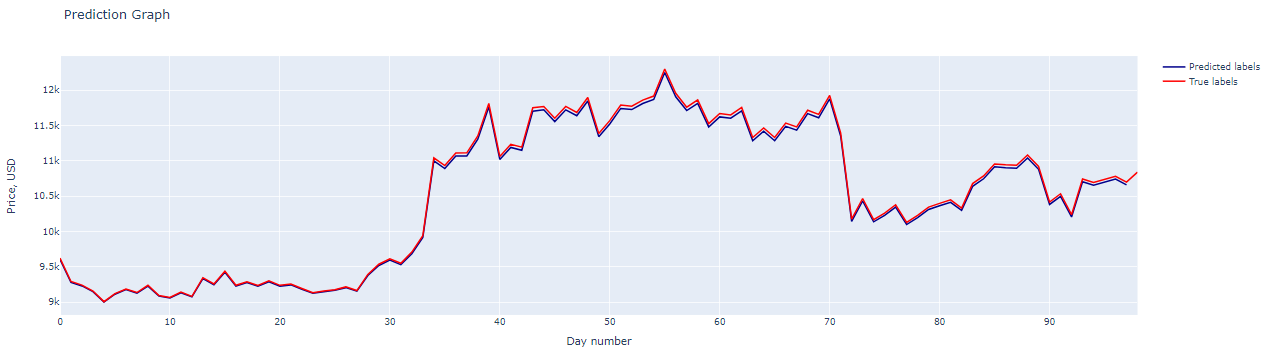
Figure 13 represents a graph that show the predicted values and actual values. As this model is implemented using real-time dataset, the graph starts from the latest date and gives a prediction for up to 100 next days. From the figure we can observe that the predicted labels are accurate to the true labels and showing 99% of accuracy. It is also observed that from 70th day to 80th day the bitcoin price reaches at its maximum price i.e. 12k US Dollars.

Figure 13 Prediction Graph

* + 1. **Experimenting with Parameters**

After tuning the model, by setting learning rate of adam optimizer to 0.01, activation sigmoid, mean squared error as metrics, and categorical\_crossentropy as loss, there seems a huge difference in training and testing validation, also the it left a great impact on prediction graph too.

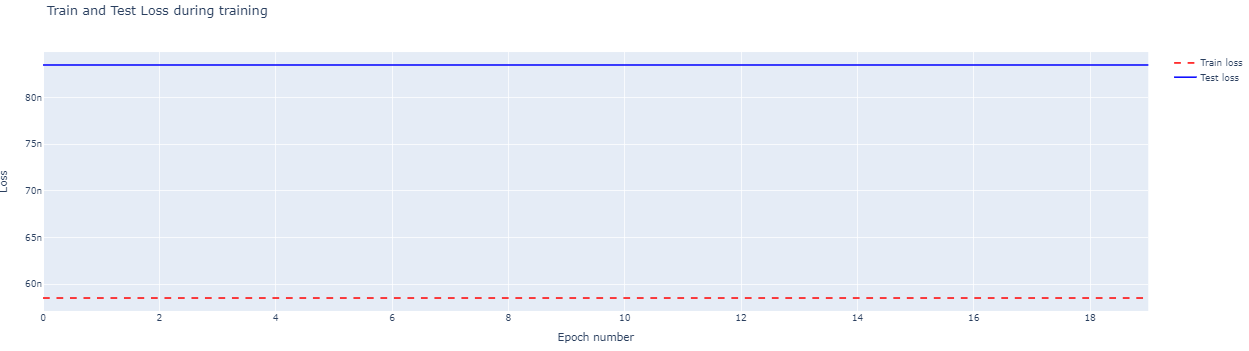
* + - 1. **Loss Occurred After Experimenting with Parameters**

Figure 14 Loss occurred after experimenting during Training and Testing

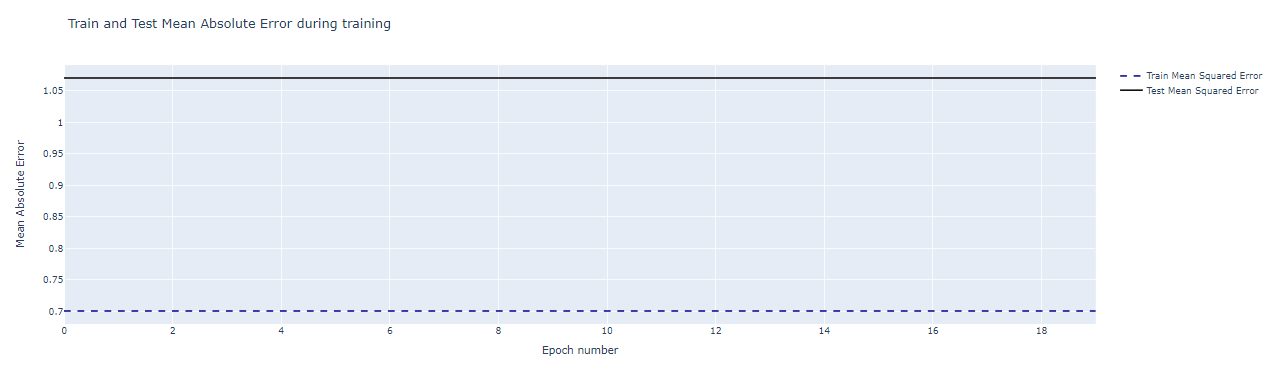
Figure 14 and 15 shows that after setting hyperparameters with values as mentioned above then there occurs huge loss during training and testing. Also Mean Squared Error Value increases too. For efficient and accurate model, the loss value should be as minimum as possible.

Figure 15 Mean Squared Error occurred after experimenting during Training and Testing

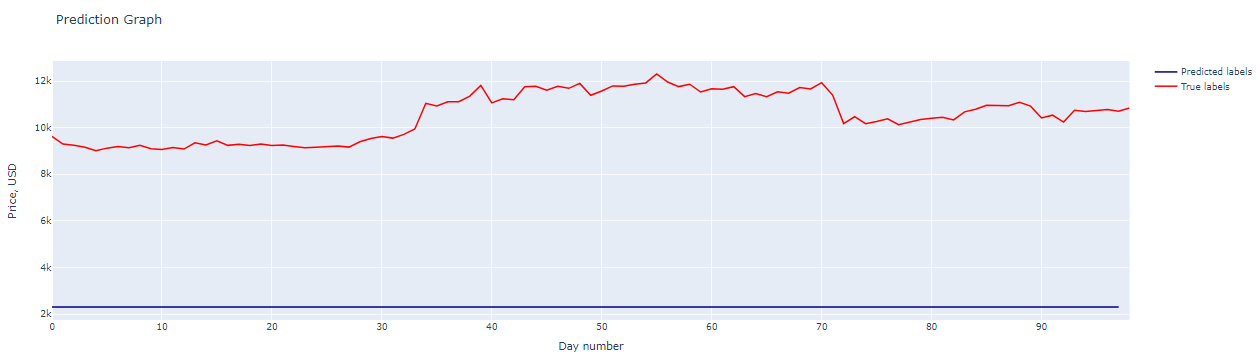
* + - 1. **Resulting Graph After Experimenting with Parameters**

Figure 16 Resulting Graph after experimenting with Parameters

From the above figure it can be assured that the parameters that were selected 1st turns out best for the model by giving the highly accurate prediction, whereas the parameters which were set for the experimental analysis turns out to be too bad for model. Thus, from these experiments, it is proved that the parameters we set before gives a great accuracy. And if we set the parameters as we mentioned after then we get the poor performance of our model. Also, a huge loss occurred during training and testing phase that leads to the poor prediction for future price of Bitcoin.