Bitcoin Price Prediction using LSTM Autoencoder Regularized by False Nearest Neighbor Loss

Chahat Raj[1], Manojit Chattopadhyay[2]
[1]chahatraj58@gmail.com, [2]mchattopadhyay@iimraipur.ac.in
[1]Department of Information Technology, Delhi Technological University, Delhi, India
[2]Information Technology and Systems, Indian Institute of Management Raipur, Chhattisgarh, India

Abstract: We implement deep learning for predicting bitcoin closing prices. Identifying two new determiners, we propose a novel LSTM Autoencoder using Mean Squared Error (MSE) loss which is regularized by False Nearest Neighbor (FNN) algorithm. The method results in reduced error rates when compared to traditional forecasting algorithms and is statistically validated. This research contributes by developing a robust algorithm that accurately determines the fluctuation directions in bitcoin prices and results in values closer to the actual prices.

Keywords: LSTM, Autoencoder, False Nearest Neighbor, Bitcoin, Regularizer

JEL Classification: G1, C6, C12

1. Introduction

Deep learning techniques brought a massive revolution in predicting stock prices, equities, mutual funds, gold, and silver, amongst other financial instruments. Nonlinearity and high volatility made the prediction difficult for financial time series (Chen et al., 2020; Rezaei et al., 2021). The trend is shifting towards predicting prices of cryptocurrencies, especially Bitcoin (Nakamoto, 2008), which has gained popularity after a multi-fold surge in its worth in a short span of six months, a phenomenon that rarely occurs in financial markets. These fluctuations have severely raised the necessity of bitcoin price predictions to mitigate market risks like bitcoin trend prediction (Cavalli & Amoretti, 2021).

In the related domain of finance few Deep learning based Long Short-Term Memory (LSTM) algorithm has been known to use for successful prediction task (McNally et al., 2018; Maknickien and Maknickas, 2012; Ahmed et al., 2020; Nelson et al., 2017; Rezaei et al., 2021). In stock market forecasting, traditional statistical and artificial intelligence methods are mainly applied that involved single value prediction. In contrast, the latest deep neural network models are applied to develop models for multiple inputs and multiple outputs based on the LSTM network (Ding & Qin, 2020). However, accurate real-time market predictions have not yet been achieved because of financial markets' volatility and chaotic nature.

Liu et al. (2020) investigated various determiners impacting bitcoin prices. However, the impact of new-age determiners such as behavioral investment and the number of blockchain wallet users has not been discussed. Bitcoin prices remained steady below a threshold of USD 15000 until October 2020, even when the number of blockchain wallet users was increasing quadratically. From October 2020 till April 2021, the latter's value rose exponentially, while a multi-fold increase in bitcoin's price was observed (Statista, 2021). This relationship is displayed in Figure 1. Studies show that every increase in bitcoin price is more likely to generate a two-fold increase in bitcoin price than observing a decrease (Phillips et al., 2018). This helps us understand the increase in blockchain wallet users as the bitcoin price increased, followed by a saturation (no significant increase) in the growth of blockchain wallet users that caused the price to fall from a peak of USD 60000 to USD 35000 during April 2021 to May 2021. This rapid investment and disinvestment pattern, which we term as behavioral investment, has a two-way relationship with high short-term fluctuations in the price of cryptocurrencies such as bitcoin. We identify this relationship and establish two new fluctuation determiners in this paper: behavioral investment and the number of blockchain wallet users.

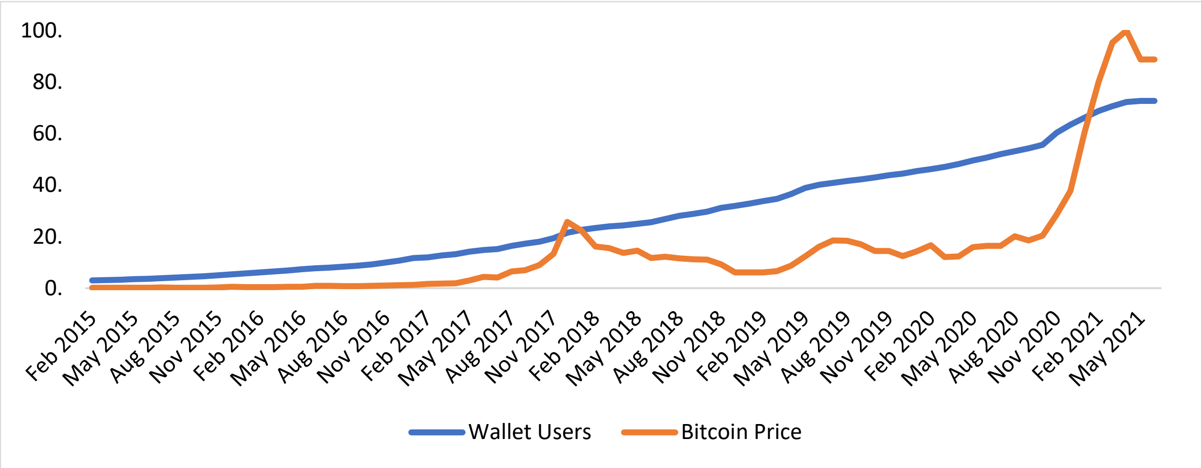

Figure 1: Relationship between bitcoin price and number of blockchain wallet users

To improve forecasting performance arising out of such fluctuations, we utilize a price convergence method incorporating the False Nearest Neighbor (FNN) algorithm introduced by Kennel et al. (1992). Gilpin (2020) adapted this method on time-series data to reconstruct strange attractors. The idea of the proposed method is derived from the fact that in an unpredictable environment such as the financial markets, deterministic conditions are non-linear and largely scattered in nature. The considerable factors are complex, a large portion of which is outliers, thus making the projection noisy. These multi-dimensional representations are linearly presented using Delay Coordinate Embedding (DCE). The necessary condition for the technique implies that the number of dimensions is known, which is not a plausible condition in financial applications. This is when FNN's use becomes indispensable, where the embedding dimension of time-series data is determined, thus overcoming the above-stated limitation of DCE.

This paper presents an LSTM autoencoder model that uses FNN as a regularizer to a traditional loss function. The method bridges a necessary gap in the existing research by bringing the bitcoin fluctuation deterministic factors into linear space, reducing noisy outliers. The deep learning method considers the critical real-world factors, where the neurons in the network automatically perform feature extraction. The use of LSTM in our approach is supported by the fact that it offers highly retentive memory with the use of forget gates. This allows the network to remember necessary factors for a long time and selectively ignore unnecessary information. Autoencoder's use is eminent for the reconstruction of information in time-series data. Overall, the method considers selectively necessary information from a multi-dimensional space (financial market factors), reconstructs them using an autoencoder, and performs prediction by exploiting the memory power of LSTMs. This justifies the importance of the proposed method. This novel method improves prediction accuracy compared to traditional forecasting algorithms, with a reduced error rate and supplements precision in bitcoin price prediction.

2. Data and Methodology

The dataset used in this research for bitcoin price prediction is procured from the coindesk website (Coindesk, 2021), similar to Demir et al. (2018), Dyhrberg (2016), and Katsiampa (2017). This dataset contains 24-hour bitcoin prices for 2791 days (1/10/2013 to 23/5/2021). We utilize the daily bitcoin closing prices to forecast future prices. Predictions are calculated for immediate next-day bitcoin prices to several days' forecasts by altering the timestep count. The prices are scaled into a range of [0,1] to minimize the magnitude impact of significant figures. Bitcoin closing price trend for 2791 observations is represented in Figure 2.

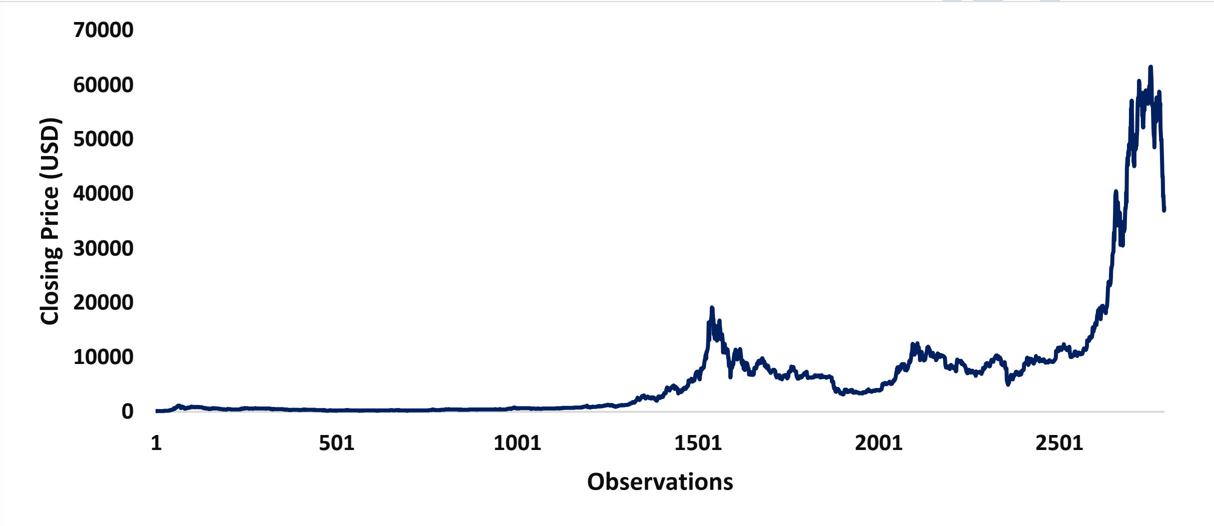

Figure 2: Bitcoin price trend

2.1 LSTM Autoencoder

Recurrent Neural networks (RNN) are primarily used in sequential data applications. LSTM is prevalent in this domain as it offers feedback connections in addition to feedforward transmissions. It is built as a self-supervised learning algorithm where it can learn from the previous sequential inputs. It can retain and store information across large data sequences possessing the ability to remember past patterns. This makes it efficient for prediction tasks.

An autoencoder is a neural network containing a single hidden layer capable of learning from compressed input representations. It performs compression of information until the model's midpoint and reconstructs it back into the input data. The input vector $x \in [0,1]_d$ with dimension, d is mapped into a hidden vector $y \in [0,1]_d$. Here y is the compressed input representation at the midpoint bottleneck. The layers performing this compression constitute the encoder $f_\emptyset$. The mapped representation y is then mapped back into the input space as a vector $z \in [0,1]_d$ using a decoder $g_{\emptyset'}$. The vectors y and z are calculated using Eq. (1) and Eq. (2).

$$y = f_\emptyset(x) = s(W_x + b) \tag{1}$$

$$z = g_{\emptyset'}(y) = s(W'_y + b') \tag{2}$$

The error encountered during the reconstruction phase is reduced by optimizing $\emptyset$ and $\emptyset'$ using a loss function L as Eq. (3).

$$\emptyset, \emptyset' = \underset{\emptyset,\emptyset'}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, z^{(i)}) = \underset{\emptyset,\emptyset'}{argmin} \frac{1}{n} \sum_{i=1}^{n} L(x^{(i)}, g_{\emptyset}(f_{\emptyset}(x^{(i)}))) \tag{3}$$

2.2 False Nearest Neighbor Regularizer

Suppose in a d-dimensional space, neighbors of a point on a trajectory are close enough, supposedly overlapping, but are largely separated in a d+1 dimensional space. In that case, these are referred to as false neighbors. Such points co-exist only in a d-dimensional space. According to the traditional FNN algorithm, the correct embedding dimension d_E is obtained when FNN converges to zero as d is increased. In an m-dimensional space, to calculate the false neighbor count F_m, the input batch-size, B and the number of nearest neighbors, K are treated as hyperparameters as in Eq. (4). K is chosen upon deciding the optimum number of neighbors overlapping or close enough to be sufficiently informative. K is dependent on B setting $K = \max(1, \lceil 0.01B \rceil)$.

$$F_m = \frac{1}{KB} \sum_{k=1}^{K} \sum_{b=1}^{B} F_{kbm} \tag{4}$$

The false nearest neighbor loss, L_{FNN}, is then calculated using F_m, and the batch activations $h \in R^{B \times L}$ of a latent layer with L units as in Eq. (5).

$$L_{FNN} = \sum_{m=2}^{L} (1 - F_m) h_m^2 \tag{5}$$

Instead of using a traditional evaluation metric, like the MSE, to calculate reconstruction loss, we add the FNN loss with an adequate weightage λ to the standard metrics. This λ acts as a regularizer and brings the total loss to converge to zero, thus increasing the prediction accuracy. The loss function is given by Eq. (6), in which $\frac{1}{n}\sum_{i=1}^{n}(Y_i - Y_i)^2$ denotes the MSE, and λ takes variable values between 0 to 1. The weightage λ differs for each application depending upon the dataset and is optimized by iterative experiments.

$$Loss = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 + \lambda L_{FNN} \tag{6}$$

2.3 Proposed Architecture

We employ an LSTM autoencoder network regularized with FNN loss to generate bitcoin price forecasts. The neurons in this deep learning model consider the effect of determinants responsible for bitcoin price change. The model selectively chooses information highly influencing the price change and discards useless information. The architectural representation of the proposed network is provided in Figure 3. It consists of two modules: an encoder and a decoder. The encoder takes an input array in shape (n_input × features × timesteps). The number of features is set to be 1. The input is fed to an LSTM layer with 256 units. The output is fed to another LSTM layer that reduces the output feature size to 128. Next, we add a Dropout layer with a probability of 0.2 that solves the problem of overfitting. This output is the compressed feature vector of the fed input. The next layer in the sequence is a RepeatVector layer that is used for feature vector replication. This layer acts as a connection between the encoder and the decoder. In the decoder module, two LSTM layers are added in the sequence opposite the encoder. The first LSTM layer contains 128 units, and the second layer comprises 256 units. Another Dropout layer of probability 0.2 is added. The final layer is a TimeDistributed Dense layer with units equal to the number of features = 1.

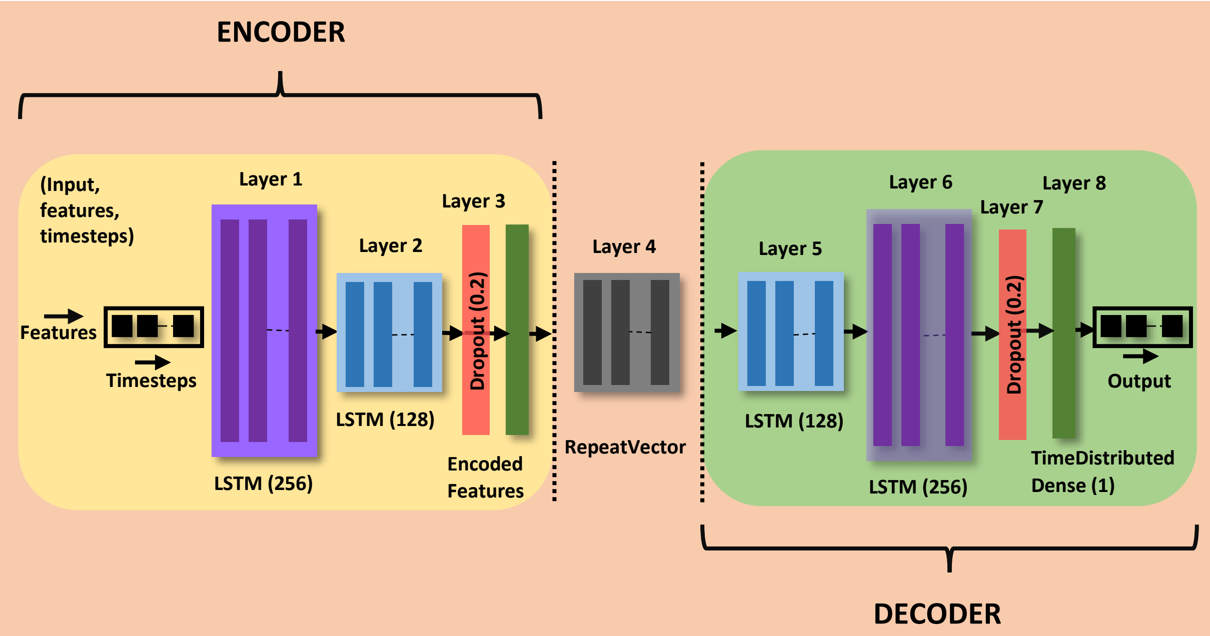

Figure 3: Proposed LSTM Autoencoder architecture

3. Numerical Experimentation

All experiments are carried out using Python 3 on Google Colab. The data points are normalized using sklearn's StandardScaler. We split the dataset with an 8:2 ratio into training and testing sets. The value for timesteps is variable for each experiment and depends on the number of days we wish to predict prices. The number of units in the latent layer, L, is set to 10 as proposed by the existing studies. The loss function for LSTM Autoencoder is the combination of MSE and FNN loss with λ as the adequate scaler weight set to 0.3. Rectified Linear Unit (ReLU) is used as the activation function for LSTM layers in both the encoder and decoder. The model uses an Adam optimizer which keeps its learning rate variable throughout the training phase. The batch size is 64, and the model is trained for 100 epochs. Table 1 lists out the values of all parameters used.

Table 1:Parameters and their values for experiments

Parameters	Value
Training Split	0.8
Batch Size (B)	64
Neighbors (K)	1
Latent layer units (L)	10
FNN regularizer weight (λ)	0.3
Epochs	100
Activation Function	ReLU
Optimizer	Adam
Language	Python 3
IDE	Google Colab

3.1 Results and Discussions

Root Mean Square Error (RMSE) is a widely accepted measure for determining time-series prediction accuracies. The error is calculated by the difference between the actual bitcoin prices and the predicted prices. RMSE, given by Eq. (7), is the average of these error values.

$$RMSE = \sqrt{\frac{1}{N}\sum(Y_i - \hat{Y}_i)^2} \quad (7)$$

The empirical results demonstrate that the proposed method is an optimum model for bitcoin price prediction. The prediction error values for varied timesteps are listed in Table 2. Figure 4 plots the bitcoin price trend for actual and predicted values. The training and testing sets contain 2232 and 559 observations, respectively. It is observable that the proposed method efficiently adapts to the price fluctuations and predicts the highs and lows of the market quite well. The curve for predicted values neatly overlaps the actual price curve during the training phase and shows minor deflections during the testing phase.

An ablation study is performed by removing the FNN loss component to highlight the effect and usefulness of the regularizer. Figure 5 depicts the ablation comparison graphically for the testing phase. It is observed that while the LSTM Autoencoder with only MSE loss can correctly predict the fluctuating directions, there is a vast gap between the curves that appears to be constant along the trajectory. This prediction gap can be treated as a constant *c* which the FNN regularizer overcomes and leads to convergence with the actual prices.

Table 2: RMSE scores for different number of prediction days

Horizon (days)	RMSE
1	0.5782
2	0.5788
3	0.5834
4	0.5893
5	0.5893
10	0.6352
20	0.6353
30	0.6472
60	0.6491

Table 3: Baseline Comparison

Model	Training RMSE	Testing RMSE
Vanilla LSTM (Shahi et al., 2020)	0.0185	2.0106
Stacked LSTM (Zanc et al., 2019)	0.0373	2.5313
SVM (Chen et al., 2020)	0.0311	1.5100
LR (Chen et al., 2020)	0.0465	2.0123
LSTM AE (Ablation Study)	0.0440	1.3056
LSTM AE + FNN	**0.0120**	**0.5782**

We compare our method with existing baseline methods for time-series prediction. We re-implement the existing works on our dataset to compare the performances of state-of-the-art models. Widely used machine learning algorithms for cryptocurrency price prediction include Support Vector Machine (SVM), Logistic Regression (LR), and Neural Networks. We use these methods for prediction power comparison as these are well-established traditional forecasting methods. The comparison is made with four popular prediction methods, namely, Vanilla LSTM (Shahi et al., 2020), Stacked LSTM (Zanc et al., 2019), SVM (Chen et al., 2020), and LR (Chen et al., 2020). The parameters of these models are the same as described in respective works, except for the loss function. All other methods use the MSE loss function. The comparison in Figure 6 highlights the accuracy and high convergence by the proposed architecture, which all other methods with the same configuration cannot achieve. Table 3 shows the comparison of RMSE values for all methods. The proposed method reaches the lowest error rates of 0.0120 and 0.5782 for training and testing phases. The model is superior to traditional machine learning algorithms because LSTM allows the deep learning model to automatically engineer real-world deterministic factors and utilize them for prediction, which is not the case for traditional methods. This accounts for the higher performance of the proposed method, overcoming the limitation of machine learning algorithms where such feature engineering becomes difficult due to the lack of deterministic datasets.

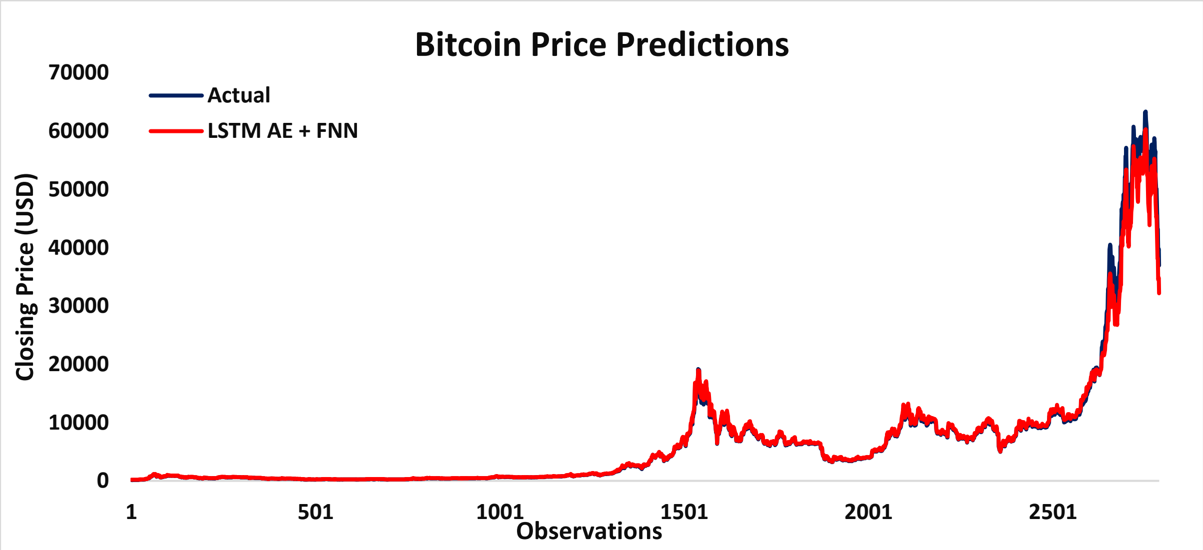

Figure 4: Actual and predicted values of bitcoin prices

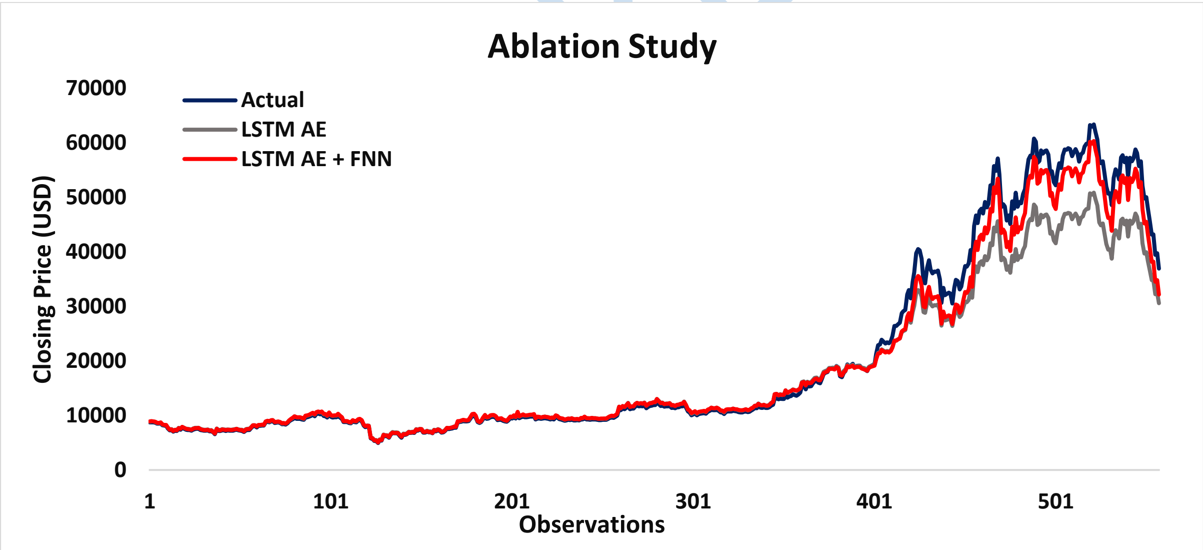

Figure 5: Comparison of predicted values with and without the FNN loss component on testing set (Grey line represents a simple LSTM Autoencoder with MSE loss, Red line represents LSTM Autoencoder with FNN loss)

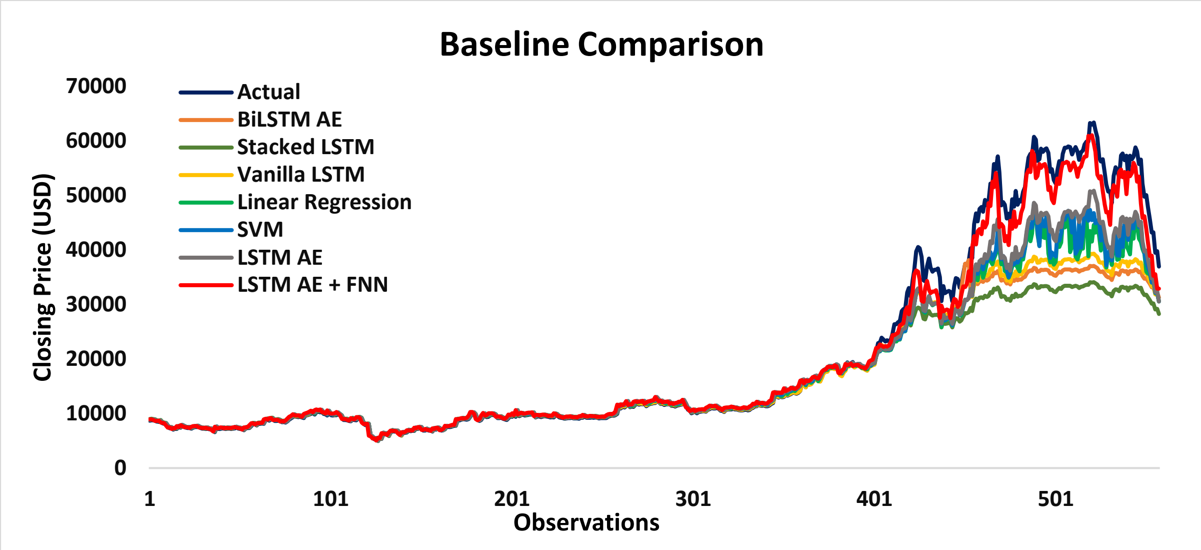

Figure 6: Baseline Comparison on the testing set

3.2 Test for Robustness

To test for the robustness of the proposed method, we perform tests for robustness using statistical pairwise t-tests and evaluate the prediction performance using two test parameters: p-value and t-statistic value. For each model, we compare the prices predicted with the actual bitcoin prices. For all the models, the p-value obtained is less than 0.05, indicating a significant difference in actual and predicted prices, and we reject the null hypothesis for each model. Table 4 shows t-statistic values obtained for each model. The t-statistic value is lowest for the proposed method compared to existing baseline methods, demonstrating that the proposed LSTM Autoencoder model with FNN regularizer is highly robust and accurate.

Table 4: Test for robustness

	Vanilla LSTM	Stacked LSTM	Bi-LSTM AE	SVM	LR	LSTM AE	LSTM AE + FNN
Actual	12.13	11.94	11.83	12.06	12.17	12.01	8.08

4. Conclusion

This research work contributes significantly to the domain of Bitcoin price prediction by employing a novel LSTM Autoencoder prediction model that uses FNN as a regularizer to the loss function. We identify two deterministic factors that impact bitcoin price fluctuations: behavioral investment and blockchain wallet users. These factors have significantly contributed to the variations yet have not been identified and established by existing works. Adding to the list of determinants by Liu et al. (2020), we narrow down the research gap in the literature. Due to many factors involved, which are scattered very far apart in multi-dimensional space, we overcome the existing gaps by proposing our novel LSTM method, which involves feature engineering by automatically selecting real-world deterministic factors. The proposed method significantly outperforms traditional prediction algorithms and provides robust results. We observe a great deal of reduction in prediction error owing to the FNN component. This research contributes to developing effective deep learning methods for time-series prediction of financial commodities such as bitcoin's price. However, the lack of a substantial amount of data limits the performance of deep learning models. Prospects of this work include developing better-performing methods that would bring the predicted prices in convergence to actual prices.

References

Chen, Z., Li, C., & Sun, W. (2020). Bitcoin price prediction using machine learning: An approach to sample dimension engineering. *Journal of Computational and Applied Mathematics*, *365*, 112395.

Rezaei, H., Faaljou, H., & Mansourfar, G. (2021). Stock price prediction using deep learning and frequency decomposition. Expert Systems with Applications, 169, 114332.

Na, H., & Kim, S. (2021). Predicting stock prices based on informed traders' activities using deep neural networks. Economics Letters, 109917.

Chen, L., Pelger, M., & Zhu, J. (2020). Deep learning in asset pricing. Available at SSRN 3350138.

Nakamoto, S., 2008. Bitcoin: a peer-to-peer electronic cash system.

Cavalli, S., & Amoretti, M. (2021). CNN-based multivariate data analysis for bitcoin trend prediction. *Applied Soft Computing*, *101*, 107065..

McNally, S., Roche, J., Caton, S., 2018. Predicting the price of Bitcoin using machine learning. In: 2018 26th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP. IEEE, pp. 339–343.

Maknickien˙ e, N., Maknickas, A., 2012. Application of neural network for forecasting of exchange rates and forex trading. In: The 7th international scientific conference" Business and Management, pp. 10–11.

Ahmed, S., Hassan, S. U., Aljohani, N. R., & Nawaz, R. (2020). FLF-LSTM: A novel prediction system using Forex Loss Function. *Applied Soft Computing*, *97*, 106780.

Nelson, D.M., Pereira, A.C., de Oliveira, R.A., 2017. Stock market's price movement prediction with LSTM neural networks. In: 2017 International Joint Conference on Neural Networks (IJCNN) (1419-1426). IEEE.

Ding, G., & Qin, L. (2020). Study on the prediction of stock price based on the associated network model of LSTM. International Journal of Machine Learning and Cybernetics, 11(6), 1307-1317.

Liu, M., Li, G., Li, J., Zhu, X., & Yao, Y. (2021). Forecasting the price of Bitcoin using deep learning. *Finance Research Letters*, *40*, 101755.

https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/

Phillips, R. C., & Gorse, D. (2018, June). Mutual-excitation of cryptocurrency market returns and social media topics. In Proceedings of the 4th international conference on frontiers of educational technologies (pp. 80-86).

Kennel, M. B., Brown, R., & Abarbanel, H. D. (1992). Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Physical review A*, *45*(6), 3403.

Gilpin, W. (2020). Deep reconstruction of strange attractors from time series. *Advances in Neural Information Processing Systems*, *33*.

https://www.coindesk.com/price/

Demir, E., Gozgor, G., Lau, C.K.M., Vigne, S.A., 2018. Does economic policy uncertainty predict the Bitcoin returns? An empirical investigation. Finance Res. Lett. 26, 145–149.

Dyhrberg, A.H., 2016. Bitcoin, gold and the dollar–a GARCH volatility analysis. Finance Res. Lett. 16, 85–92. Ho, T.K., 1995. Random decision

Katsiampa, P., 2017. Volatility estimation for Bitcoin: A comparison of GARCH models. Econom. Lett. 158, 3–6.

Shahi, T. B., Shrestha, A., Neupane, A., & Guo, W. (2020). Stock Price Forecasting with Deep Learning: A Comparative Study. *Mathematics*, *8*(9), 1441.

Zanc, R., Cioara, T., & Anghel, I. (2019, September). Forecasting Financial Markets using Deep Learning. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)* (pp. 459-466). IEEE.