# Ethereum and Binance Coin Price Prediction using GANs

**A report on**
**Deep Learning Lab Project**
**[CSE-3271]**

Submitted By
**Pranay Jaju - 210962146**
**Liam Bopanna M.M - 210962178**

MANIPAL
ACADEMY *of* HIGHER EDUCATION
*(Institution of Eminence Deemed to be University)*
INSPIRED BY LIFE

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**
**MANIPAL INSTITUTE OF TECHNOLOGY,**
**MANIPAL ACADEMY OF HIGHER EDUCATION**
**April 2024**

# Ethereum and Binance Coin Prediction using GANs

[1]Pranay Jaju, [2]Liam Bopanna M.M
[1]CSE & Manipal Institute of Technology, India
[2]CSE & Manipal Institute of Technology, India
[1]pranay.jaju@learner.manipal.edu; [1]liam.muthanna@learner.manipal.edu

***Abstract*— Deep learning is an exciting topic and has been utilized in many areas owing to its strong potential. For example, it has been widely used in the financial area which is vital to the society, such as high-frequency trading, portfolio optimization, fraud detection and risk management. In this paper, it develops a predictive model that can accurately forecast the future prices of Ethereum and Binance using Generative Adversarial Network (GAN) with Gated Recurrent Units (GRU) used as a generator that inputs historical price and generates future price and Convolutional Neural Network (CNN) as a discriminator to discriminate between the real price and generated price. The performance of the proposed model is evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE), and the results are compared with traditional time-series forecasting technique, ARIMA. The findings of this research have the potential to provide valuable insights for traders, investors, and cryptocurrency enthusiasts, enabling informed decision-making and risk management strategies in the volatile cryptocurrency market.***

***Keywords* — Ethereum (ETH), Binance (BNB), Generative Adversarial Networks (GANs), Gated Recurrent Units (GRUs), Convolution Neural Network (CNNs), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE)***

## I. INTRODUCTION

The first stone in the creation of the cryptocurrency market was the white paper published by Nakamoto[1] explaining the creation and operation of a new digital currency which has the particularity of being decentralized and does not require the intermediation of any financial institution. Among the myriad of cryptocurrencies, Ethereum and Binance Coin have garnered significant attention due to their innovative features and widespread adoption. Ethereum, introduced by Vitalik Buterin[2], is a decentralized platform that enables the creation of smart contracts[3] and decentralized applications (dApps)[4] and the second-largest cryptocurrency by market capitalization. Binance Coin, on the other hand, is the native cryptocurrency of the Binance exchange, one of the largest and most popular cryptocurrency trading platforms globally.

Predicting the price movements of cryptocurrencies has been a challenging task due to their inherent volatility, complex dynamics, and susceptibility to various factors, including market sentiments, regulatory changes, and technological advancements. Traditional time-series forecasting methods, such as autoregressive integrated moving average (ARIMA)[5],[6] and exponential smoothing[7]-[9], have shown limitations in capturing the non-linear and complex patterns present in cryptocurrency price data.

In recent years, deep learning techniques have gained prominence in various domains, including finance and economics, due to their ability to model intricate relationships and extract valuable insights from large datasets. Generative Adversarial Networks (GANs)[10], a class of deep learning models introduced by Ian Goodfellow and his colleagues in 2014, have demonstrated remarkable performance in generating synthetic data and capturing complex distributions.

This research aims to leverage the power of GANs to develop a predictive model for Ethereum and Binance Coin prices. By training the GAN architecture on historical price data, the proposed model seeks to capture the underlying patterns and dynamics governing the price movements of these cryptocurrencies. The generated synthetic data can then be used to train a forecasting model, enabling accurate predictions of future prices.

The introduction should provide a comprehensive overview of the research problem, highlighting the significance of predicting cryptocurrency prices and the limitations of traditional forecasting methods. It should also briefly introduce the concept of GANs and their potential applications in the cryptocurrency domain.

## II. LITERATURE REVIEW

The research paper proposes a novel approach using Generative Adversarial Networks (GANs) to predict the prices of Ethereum (ETH) and Binance Coin (BNB), two major cryptocurrencies. By leveraging the adversarial training process of GANs, the model learns to capture the underlying patterns and generate realistic price forecasts. The GAN models are trained on historical price and volume data, and their performance is evaluated against traditional forecasting methods and simpler deep learning techniques. The results demonstrate the superiority of the GAN-based approach in terms of various evaluation metrics, such as RMSE and MAPE.

[20]This paper employs Recurrent Neural Network (RNN)[11] and Long Short-Term Memory (LSTM)[12] models with 10-fold cross-validation for Bitcoin price forecasting. The authors retrieve data from CoinMarketCap, perform data cleaning and normalization, and create additional features like price volatility and daily highs. The proposed RNN with LSTM models outperform traditional techniques like Linear Regression and Random Forest, achieving mean absolute errors (MAE) of 0.1518 and 0.0043, respectively. However, the study is limited to Bitcoin price prediction, and the findings may not generalize to other cryptocurrencies or financial assets. Additionally, the high volatility of cryptocurrency markets poses challenges for accurate price forecasting, as sudden market fluctuations or external events may not be fully captured by the models.

[21]This paper proposes a hybrid cryptocurrency price prediction scheme that combines Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)[13] deep learning models to forecast the prices of Litecoin and Monero cryptocurrencies accurately within 1, 3, and 7-day window sizes. The authors collected daily cryptocurrency data, pre-processed it using Min-Max normalization, split it into training and testing datasets, and trained the hybrid model on the training data. The model achieved reasonable RMSE values for 1-day and 3-day predictions but struggled with 7-day predictions, highlighting the limitations of these methods in capturing the high volatility and unpredictable nature of cryptocurrency prices over longer time horizons due to external factors and the resemblance of cryptocurrency prices to a random walk process.

[22]This paper proposes a novel hybrid model that combines Hidden Markov Models (HMMs)[14] for descriptive analysis and Long Short-Term Memory (LSTM) networks for predictive forecasting of cryptocurrency prices, specifically Bitcoin. The model incorporates raw 2-minute frequency Bitcoin data from Coinbase, along with extracted features such as order and trade data, and technical indicators. Genetic Algorithms (GA)[15] were used for optimal hyperparameter tuning for the LSTM component. The proposed HMM-LSTM model outperforms traditional ARIMA and regular LSTM models, achieving lower RMSE and MAE values for 1-step and 2-step ahead predictions. However, the study acknowledges

limitations, including the potential complexity and interpretability challenges of the hybrid model, as well as questions regarding its applicability to other cryptocurrencies or market conditions.

[23]The research investigates the efficacy of a transformer-based neural network for forecasting Ethereum cryptocurrency prices, with the hypothesis that prices are strongly correlated with other cryptocurrencies and sentiments. The proposed model incorporates multiple features such as volume, sentiment data from Twitter, Reddit, and news outlets, and prices of correlated cryptocurrencies like Polkadot and Cardano. The sentiment analysis is performed using the FinBERT model, and the data is pre-processed accordingly. A regular transformer encoder architecture with stacked blocks and global pooling is employed. The model surpasses artificial neural networks (ANNs) and multilayer perceptrons (MLPs) on some metrics, achieving RMSE of 0.0716 and 0.189, and MAPE of 14.91 and 18.84 for scenarios with and without sentiment data, respectively. However, the study acknowledges limitations, including the use of a regular transformer instead of more advanced models such as Autoformer and TimeGPT-1 and limited sources for sentiment data collection.

[24]This paper proposes an innovative model that combines signal decomposition and deep learning techniques to address the challenges of forecasting non-linear and non-stationary time series data. The model employs Generalized Autoregressive Conditional Heteroskedasticity (GARCH)[16] for learning volatility in time series changes, followed by Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) for data decomposition into Intrinsic Mode Functions (IMFs). Graph Convolutional Networks (GCN) are then applied to effectively learn the features of the decomposed data. The proposed model is demonstrated on three different time series datasets: Air Quality, Energy, and Traffic, achieving RMSE of 16.64 and MAPE of 35.32 for the Air Quality dataset. However, the study acknowledges limitations, including the introduction of GARCH being particularly invaluable for handling volatility in financial series data, and the challenges of model complexity, computational demands, and the risk of overfitting.

[25]The study proposes a novel deep learning framework that combines Wavelet Transforms (WTs)[17],[18], Stacked Autoencoders (SAEs)[19], and Long Short-Term Memory (LSTM) for stock price forecasting. The WT is used to decompose the stock price time series and eliminate noise, followed by SAEs to generate deep high-level features for price prediction. These denoised high-level features are then fed into an LSTM model to forecast the next day's closing price. The framework is applied to six major stock indices: CSI 300, Nifty 50, Hang Seng, Nikkei 225, S&P 500, and DJIA, across different market conditions from 2010 to 2016. The proposed WSAEs-LSTM model achieved impressive returns of 45% to 64% for all the analysed markets over the 6-year period, with a MAPE of 0.011 for S&P 500 and DJIA. However, the study acknowledges limitations, including the potential for better hyperparameter tuning and the computationally intense and complex nature of the model, which may hinder scalability.

[26] The study introduces SeqGAN, a sequence generation framework aimed at addressing two significant challenges encountered when applying GANs for generating sequences. Firstly, GANs are designed for generating real-valued, continuous data but face difficulties in directly generating discrete token sequences. Secondly, GANs can only evaluate the score or loss for an entire sequence once it has been generated; for a partially generated sequence, it is non-trivial to balance its current score and the future score of the complete sequence. By modelling the data generator as a stochastic policy in reinforcement learning (RL), SeqGAN circumvents the generator differentiation problem by directly performing gradient policy updates. The RL reward signal is derived from the GAN discriminator's evaluation of a

complete sequence and is propagated back to the intermediate state-action steps using Monte Carlo search. On synthetic data, SeqGAN significantly outperformed maximum likelihood methods, scheduled sampling, and PG-BLEU and superior performance compared to baseline real-world tasks like poem, speech language, and music generation as assessed by various metrics including human judgements.

[27] This is a systematic literature review on the usage of GANs in time series. The paper gives a background of GANs and their usage in the fields of image generation, natural language processing (NLP) and now how they are being applied to time series data. The review systematically analyses a wide range of research articles, focusing on GAN architectures specifically designed for time series data. The included studies cover various applications such as medical/physiological data generation, financial time series prediction, traffic flow imputation, music generation, and patient record generation. The review identifies several GAN architectures tailored for time series data, including SeqGAN[26], QuantGAN[28], Continuous RNN-GAN (C-RNN-GAN)[29], TimeGAN[30], Conditional Sig-Wasserstein GAN (SigCWGAN)[31], Sequentially Coupled GAN (SC-GAN)[32], Synthetic Biomedical Signals GAN (SynSigGAN)[33]. These architectures have been applied to diverse datasets such as EEG, ECG, EHRs, PPG, EMG, speech data, financial indices (S&P 500, Dow Jones), ETFs, and synthetic sets.

While there have been studies focused on predicting the prices of major cryptocurrencies like Bitcoin, there is a lack of research specifically targeting Ethereum and Binance Coin, two of the most widely traded cryptocurrencies. Additionally, the application of Generative Adversarial Networks (GANs), a powerful deep learning technique, remains unexplored in the domain of cryptocurrency price prediction. Primary objective of this research is to develop and evaluate a novel GAN model tailored for accurately predicting the prices of Ethereum (ETH) and Binance Coin (BNB).

## III. **METHODOLOGY**

In this section, we outline the methodology employed for developing a Generative Adversarial Network (GAN) for the purpose of predicting prices of Ethereum and Binance Coin. The section is broken down into 4 parts – A. Data Collection and Pre-Processing, B. GAN Architecture, C. Training Process, and D. Evaluation Strategy.

### Data Collection and Pre-Processing

The data collection process for this project involved gathering historical price data from Yahoo Finance. The data thus gathered comprised of the open, close, high, low, volume, and adjusted close values for a 5-year period from April 1$^{th}$ 2019 to April 1$^{th}$ 2024 for Ethereum and April 1$^{th}$ 2019 to April 1$^{th}$ 2024 for Binance Coin, giving a total of 1828 data points for each crypto.

The research employed a Variational Autoencoder (VAE) as a dimensionality expansion

The research employed a comprehensive feature engineering and data preparation process to enhance the representational capacity and capture intricate patterns within the dataset. After splitting the original six-attribute data into training and testing sets using an 80-20 split, it is normalized using Min-Max Normalization. Then a Variational Autoencoder (VAE) was utilized to project the training data into a higher 10-dimensional latent space, potentially extracting complex nonlinear relationships and intricate patterns hidden in the lower dimensions. The learned latent features from the VAE were then concatenated with the original dataset, resulting in a 16-feature representation. To incorporate temporal dependencies, a sliding window approach with a length of 3 days was employed, where the model utilized three consecutive days' worth of data to predict the price for the subsequent day. This enriched 16-feature dataset, structured using the sliding window technique, was the used as the input data to the GAN model.

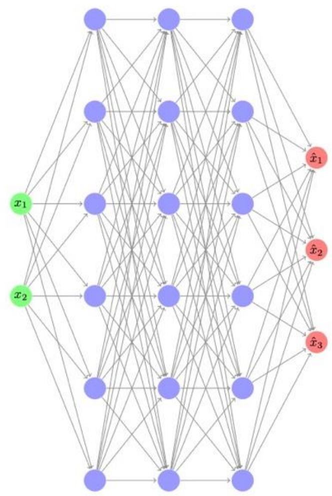

Fig 1. A simplified VAE with layers as (2,6,6,6,3)

## GAN Architecture

The Generator – Gated Recurrent Units (GRUs) were used as the generator for the GAN model. The input of the generator will be three-dimensional data: batch size, input-step, and feature, and the output will be two-dimensional data: batch size and output-step. For building up a generator with good performance, this model uses three layers of GRU, the number of neurons of 1024, 512, 256, each of which is followed by a Dropout layer with p=0.2. This is followed by three layers of Dense Network with 256, 128, 64 neurons, and the last layer will have the same neurons as the output step that will be predicted.

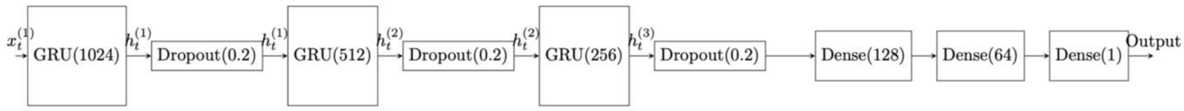


Fig 2. Generator's Architecture

The Discriminator – Convolutional Neural Networks (CNNs) were used as the discriminator for the GAN model. The input of the discriminator will be either from the original or fake dataset. It includes three 1D-convolutioon layers with 32, 64, 128 neurons followed by three layers of Dense Network with 220, 220, 1 neuron. The activation function used is Leaky Rectified Linear unit (ReLU) for each layer except the last which uses normal ReLU.

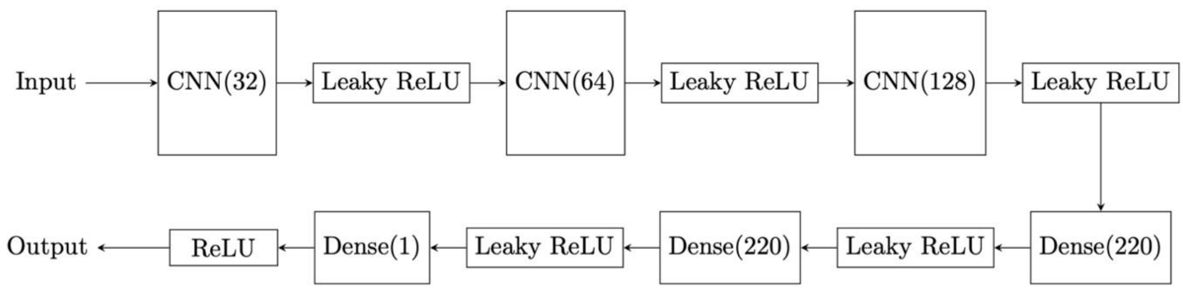


Fig 3. Discriminator's Architecture

### Training Process

The VAE was trained using a sum of KL-Divergence and Binary Cross-Entropy Loss. The GAN was trained using Wasserstein Distance as its loss. The Adam optimizer was chosen for its efficiency in handling sparse gradients and adaptive learning rates, with an initial rate of 0.000115. The model was trained for 100 epochs with the following key steps:

- First, the generator was used to produce fake output given the input which was concatenated with the real output.
- Simultaneously the real and fake data were fed to the discriminator and their outputs were obtained.
- The loss for the discriminator was calculated as the negative of the difference between the mean of discriminator's output for the real and fake data. The optimizer is updated.
- The loss for the generator was calculated as negative of the mean of the discriminator's output for the fake data. The optimizer is updated.

### Evaluation Strategy

The model's capability was assessed on unseen test set after the training phase. Performance metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) were calculated to evaluate the model comprehensively.

## IV. RESULTS AND DISCUSSION

The training of the GAN was carried out meticulously over 100 epochs, recording losses of the generator and discriminator at each epoch. The graphical representation of the model's performance shows a trend towards convergence of the losses of the generator and the discriminator, as seen in Fig.1 and Fig.2.

The RMSE and MAPE values for Ethereum and Binance Coin are provided in Table 1.

The graphs for train and test dataset and their respective predictions for Ethereum and Binance are provided in Fig.3 and Fig.4 respectively.

| **Data** | **RMSE** | **MAPE** |
|---|---|---|
| Train Data - Ethereum | 0.0392 | 8.7930 |
| Test Data - Ethereum | 0.0383 | 0.0821 |
| Train Data - Binance | 0.0519 | 48.8525 |
| Test Data - Binance | 0.0339 | 0.0707 |

Table 1. Results Table – RMSE and MAPE values

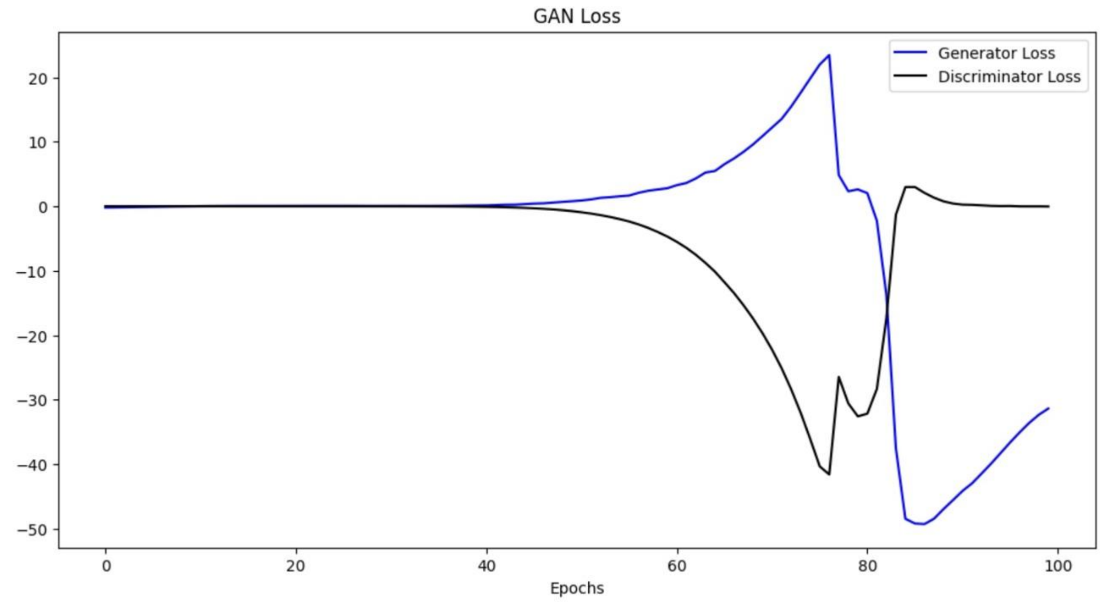


Fig 1. Model Loss during training for Ethereum

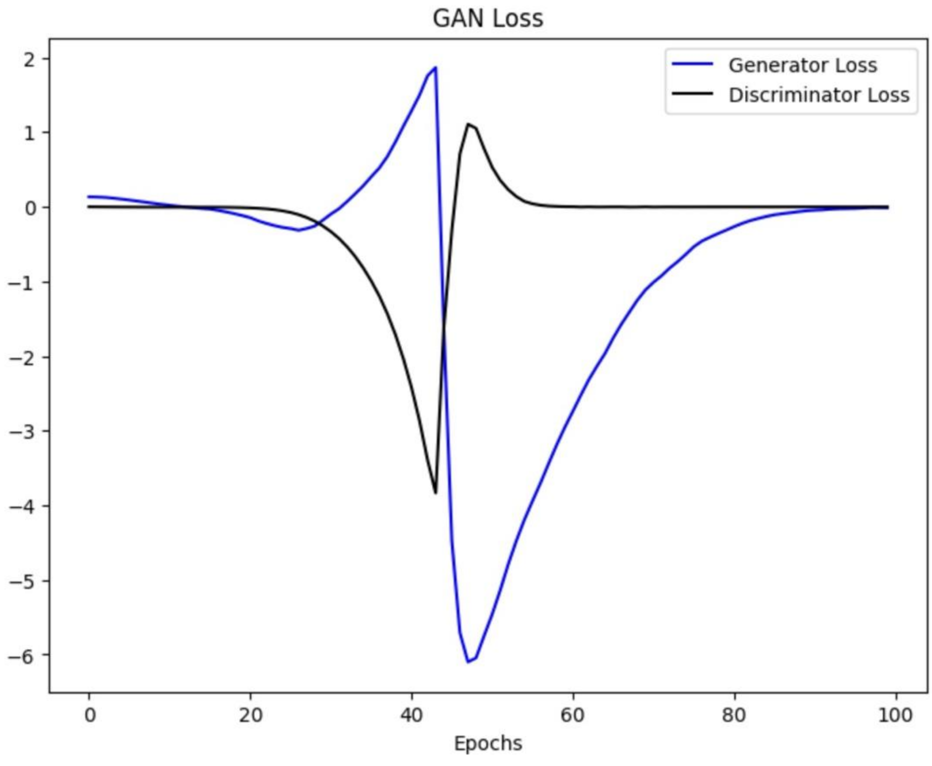


Fig 2. Model Loss during Training for Binance Coin

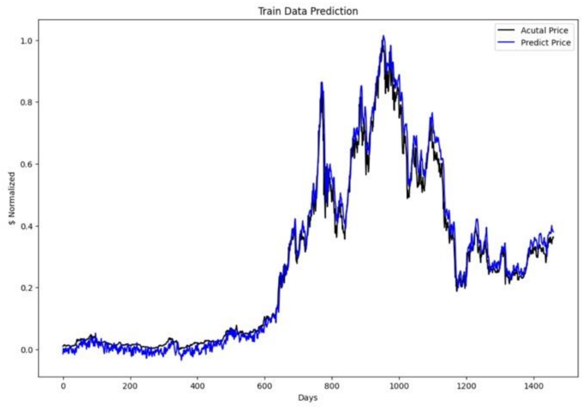


Fig 3a. Train Data with Prediction for ETH

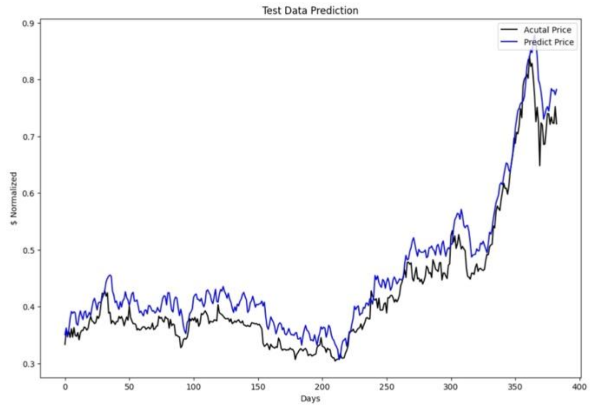


Fig 3b. Test Data with Prediction for ETH

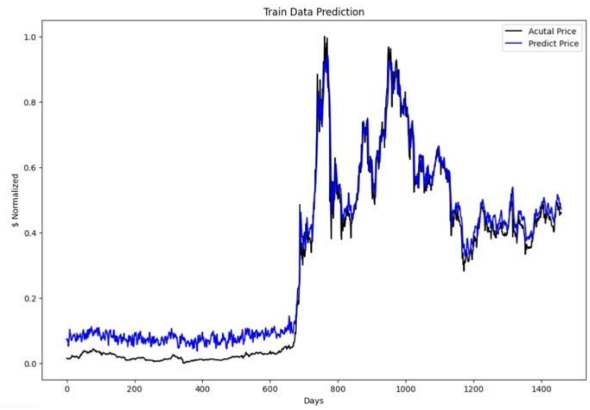


Fig 4a. Train Data with Predictions for BNB

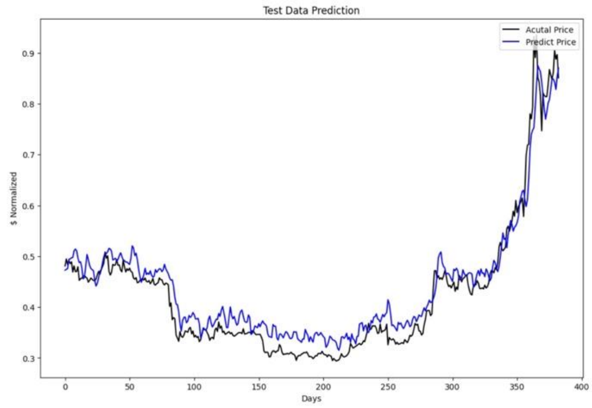


Fig 4b. Test Data with Predictions for BNB

## V. CONCLUSIONS

This paper proposed a GAN model with GRU as its generator and CNN as its discriminator. According to the experimental result, some conclusions have been made. Compared to the already existing research, using GRUs, LSTMs and other models, the proposed GAN model has outperformed these conventional models for single-look ahead prediction. However, hyperparameter tuning for such models is a challenging task and without suitable parameters, undesirable results may be achieved.

## VI. FUTURE WORK

Future research should focus on the development of hyperparameter tuning techniques. If the parameters are better tuned more accurately, the results may improve.

Another area for future research could be towards the development and implementation of GANs for multi-look ahead prediction.

Another future research could be towards the implementation of different models for generator and discriminator in the GAN model.

## REFERENCES


[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[2] V. Buterin, "A next-generation smart contract and decentralized application platform," white paper, 2014, vol. 3, no. 37, pp. 2-1.

[3] D. Macrinici, C. Cartofeanu, and S. Gao, "Smart contract applications within blockchain technology: A systematic mapping study," *Telematics and Informatics*, vol. 35, no. 8, pp. 2337-2354, 2018.

[4] K. Wu, Y. Ma, G. Huang, and X. Liu, "A First Look at Blockchain-based Decentralized Applications," 2019. Online. Available: arXiv:1909.00939 [cs.SE]

[5] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, UK, 2014, pp. 106-112, doi: 10.1109/UKSim.2014.67.

[6] Z. Tang, C. De Almeida, and P. A. Fishwick, "Time series forecasting using neural networks vs. Box-Jenkins methodology," *Simulation*, vol. 57, no. 5, pp. 303-310, 1991.

[7] E.S. Gardner Jr, "Exponential smoothing: The state of the art," *Journal of Forecasting*, vol. 4, no. 1, pp. 1-28, 1985.

[8] E.S. Gardner Jr, "Exponential smoothing: The state of the art—Part II," *International Journal of Forecasting*, vol. 22, no. 4, pp. 637-666, 2006.

[9] A.M. De Livera, R.J. Hyndman, and R.D. Snyder, "Forecasting time series with complex seasonal patterns using exponential smoothing," *Journal of the American Statistical Association*, vol. 106, no. 496, pp. 1513-1527, 2011.

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," 2014. Online. Available: arXiv:1406.2661 [stat.ML].

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Proceedings of the 8th Annual Conference of the Cognitive Science Society*, 1986.

[12] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1740, Dec. 1997. doi: 10.1162/neco.1997.9.8.1735.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," 2014. Online. Available: arXiv:1412.3555 [cs.NE]

[14] L. Rabiner and B. Juang, "An introduction to hidden Markov models," in *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4-16, Jan 1986, doi: 10.1109/MASSP.1986.1165342.

[15] J. H. Holland, "Genetic Algorithms," *Scientific American*, vol. 267, no. 1, pp. 66-73, 1992.

[16] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307-327, 1986. doi: 10.1016/0304-4076(86)90063-1.

[17] A. Graps, "An Introduction to Wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Jun. 1995. doi: 10.1109/99.388960.

[18] I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis," in *IEEE Transactions on Information Theory*, vol. 36, no. 5, pp. 961-1005, Sept. 1990, doi: 10.1109/18.57199

[19] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P. A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, 2010.

[20] S. Tandon, S. Tripathi, P. Saraswat and C. Dabas, "Bitcoin Price Forecasting using LSTM and 10-Fold Cross validation," *2019 International Conference on Signal Processing and Communication (ICSC)*, NOIDA, India, 2019, pp. 323-328, doi: 10.1109/ICSC45622.2019.8938251.

[21] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A Deep Learning-based Cryptocurrency Price Prediction Scheme for Financial Institutions," *Journal of Information Security and Applications*, vol. 55, p. 102583, 2020, ISSN: 2214-2126, doi: 10.1016/j.jisa.2020.102583.

[22] I. A. Hashish, F. Forni, G. Andreotti, T. Facchinetti and S. Darjani, "A Hybrid Model for Bitcoin Prices Prediction using Hidden Markov Models and Optimized LSTM Networks," *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, 2019, pp. 721-728, doi: 10.1109/ETFA.2019.8869094.

[23] S. Singh and M. Bhat, "Transformer-based approach for Ethereum Price Prediction Using Crosscurrency correlation and Sentiment Analysis," 2024. *Online*. Available: arXiv:2401.08077 [cs.LG].

[24] H. Han, Z. Liu, M. Barrios Barrios, et al., "Time series forecasting model for non-stationary series pattern extraction using deep learning and GARCH modelling," *J Cloud Comp*, vol. 13, no. 2, 2024. doi: 10.1186/s13677-023-00576-7.

[25] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PLoS ONE*, vol. 12, no. 7, pp. e0180944, 2017. doi: 10.1371/journal.pone.0180944.

[26] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient," 2017. Online. Available: arXiv:1609.05473 [cs.LG].

[27] E. Brophy, Z. Wang, Q. She, and T. Ward, "Generative Adversarial Networks in Time Series: A Systematic Literature Review," *ACM Comput. Surv.*, vol. 55, no. 10, articleno. 199, pp. 1-31, Oct. 2023. doi: 10.1145/3559540.

[28] M. Wiese, R. Knobloch, R. Korn, and P. Kretschmer, "Quant GANs: deep generation of financial time series," *Quantitative Finance*, vol. 20, no. 9, pp. 1419-1440, Apr. 2020. doi: 10.1080/14697688.2020.1730426.

[29] O. Mogren, "C-RNN-GAN: Continuous recurrent neural networks with adversarial training," 2016. Online. Available: arXiv:1611.09904 [cs.AI]

[30] J. Yoon, D. Jarrett, and M. van der Schaar, "Time-Series Generative Adversarial Networks," in *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, vol. 32, Curran Associates, Inc., 2019.

[31] S. Liao, H. Ni, L. Szpruch, M. Wiese, M. Sabate-Vidales, and B. Xiao, "Conditional Sig-Wasserstein GANs for Time Series Generation," 2023. Online. Available: arXiv:2006.05421 [cs.LG]

[32] L. Wang, W. Zhang, and X. He, "Continuous Patient-Centric Sequence Generation via Sequentially Coupled Adversarial Learning," in *Continuous Patient-Centric Sequence Generation via Sequentially Coupled Adversarial Learning*, 2019, pp. 36-52. doi: 10.1007/978-3-030-18579-4_3.

[33] D. Hazra and Y.-C. Byun, "SynSigGAN: Generative Adversarial Networks for Synthetic Biomedical Signal Generation," *Biology*, vol. 9, no. 12, p. 441, 2020. doi: 10.3390/biology9120441.