Comparison of conventional deep learning models and Generative Adversarial Network for stock price prediction

QF634 - Applied Quantitative Research Methods

Group member

1.) Balina Kirti Kumar
2.) Kendrick Winata
3.) Najwa Jia Hui Rujok
4.) Saran Somboonsiripok

Academic year 2023 - 2024
MSc in Quantitative Finance
Singapore Management University

1

Objective

- Generative Adversarial Network (GAN) is a new machine learning framework introduced in 2014. After it was introduced, there have been only few papers that applied it to stock price prediction
- The main objective of this project is to compare the efficacy of the GAN framework with conventional deep learning models such as Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) for the purpose of stock price prediction

Outline

1.) Theoretical Background

- **Deep learning models**
  - Long short-term memory model (LSTM)
  - Gated Recurrent Unit (GRU)
  - Convolutional neural networks (CNN)
- **Generative Adversarial Network (GAN)**

2.) Datasets and Features

- **Datasets**
  - AAPL
  - EBAY
  - SBUX
- **Features**
  - Basic features
  - Fundamental Indicators
  - Technical Indicators

3.) Methodology

- **Data preprocessing**
  - Normalization
  - Reshaping data
  - Train test split
- **LSTM architecture**
- **GRU architecture**
- **GAN architecture**

4.) Experimental Result

- **Performance measurement**
- **Experimental Result**
  - AAPL
  - EBAY
  - SBUX

2

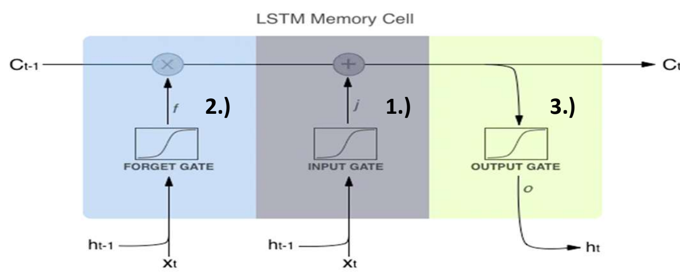
SMU

1.) Theoretical Background

Long short-term memory networks (LSTM)

- LSTM is a more advanced version of Recurrent Neural Network (RNN) that solves the vanishing gradients, exploding gradients, and long-term dependency problems in RNN. RNNs are not able to preserve the context for long range sequences.
- The basic components of LSTM include an input gate, an output gate, and a forget gate. The purpose of these gates is to control the flow of information within the LSTM model.

2.) Forget gate decides on how much of the long-term state should be erased.



Computed based on current input and previous internal state.

1.) Input gate decides on how the internal state will be updated.

3.) Output gate gate controls which part of the long-term state should be **read and output** at this timestep.

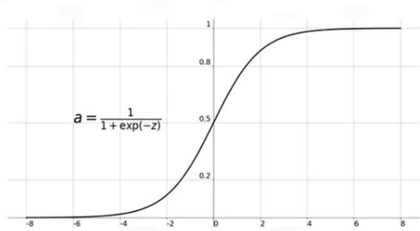
3

SMU

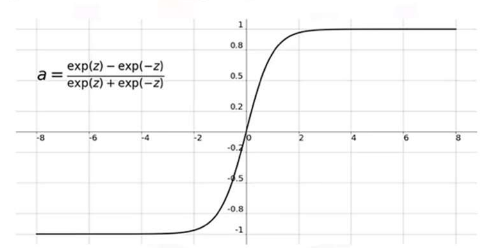
1.) Theoretical Background

Long short-term memory networks (LSTM)

Sigmoid Function



Hyperbolic Tangent Function



- Gates in the LSTM model make use of sigmoid activation functions, which produces an output of 0, 1 or a value between.
- An output of 0 means that the gate is blocking everything, while an output of 1 means that the gate allows everything to pass through it.

- Tanh activation function is also used in the LSTM Model, but it is mostly used for hidden layers.
- Tanh is similar to sigmoid but its range is from -1 to 1 and it results in zero centered data which helps data normalization and leads to faster data convergence.

4

4

1.) Theoretical Background

Long short-term memory networks (LSTM)

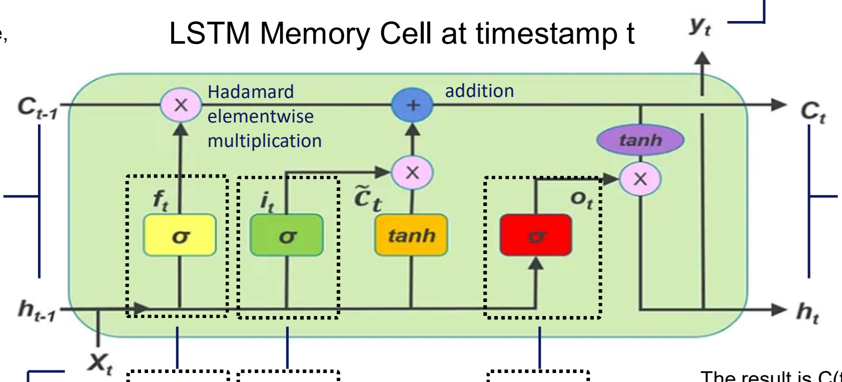
Output

The long-term state C(t-1) goes through a forget gate, dropping some memories, then it adds memories selected by the input gate.

Output of LSTM cell of the previous timestep (t-1)

LSTM cells manage two state vectors, h and c. h is the short-term state and c is the long-term state.

LSTM Memory Cell at timestamp t



Generated as inputs for LSTM cell of the next timestep (t+1)

Input

Forget gate

Input gate

Output gate

The result is C(t). The result is also copied and passed through the tanh function, filtered by the output gate. This produces the short-term state h(t), which is also the output y(t).

5

1.) Theoretical Background

Long short-term memory networks (LSTM)

LSTM Summary

- LSTM cell can learn to recognize an important input (input gate)
- Store it in the long-term state (state vector c)
- Learn to preserve it for as long as it is needed (forget gate)
- Learn to extract it whenever it is needed

- Thus, LSTM models have been successful at capturing long-term patterns in time series

6

6

1.) Theoretical Background

Gated Recurrent Unit (GRU)

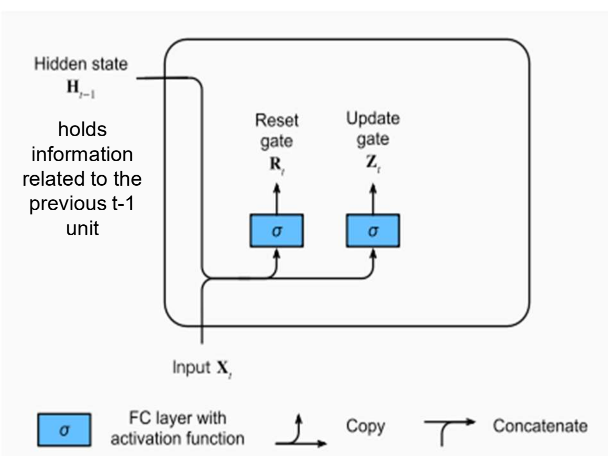
- GRU, similar to LSTM, is able to selectively remember and forget information over time. Thus, GRU is also suitable for the modelling of sequential data.

Parameters	LSTMs	GRUs
Structure	More complex	Simpler than LSTM
Training	Can be more difficult	Easier than LSTM
Performance	Good for complex tasks	Can be intermediate between simple and complex tasks
Hidden state	Multiple (memory cell)	Single — In GRU, both state vectors (c and h) are merged into a single vector h
Gates	Input, output, forget	Update, reset
Ability to retain long-term dependencies	Strong	Intermediate between RNNs and LSTMs

7

1.) Theoretical Background

Gated Recurrent Unit (GRU)



Reset gate for time step t:

$$r_t = \sigma(w_{xr}x_t + w_{hr}h_{t-1} + b_r)$$

- The **reset gate** decides how much of the previous hidden state to forget

Update gate for time step t: bias parameters

$$z_t = \sigma(w_{xz}x_t + w_{hz}h_{t-1} + b_z)$$

- The **update gate** decides how much of the candidate activation vector should be incorporated into the new hidden state

- The reset gates capture **short-term** dependencies in sequences, while the update gates capture **long-term** dependencies in sequences

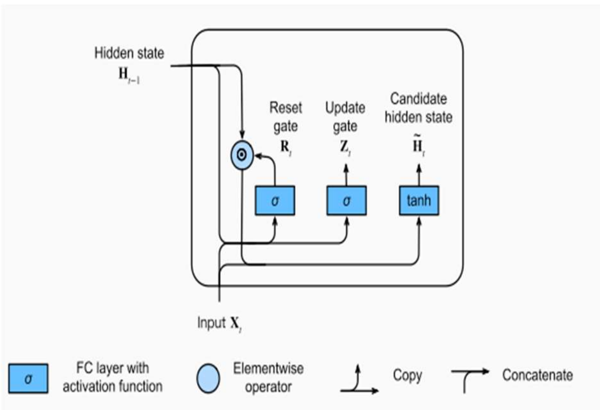
8

8

SMU

1.) Theoretical Background

Gated Recurrent Unit (GRU)



Candidate hidden state for timestep t

bias parameter

$$\widetilde{H_t} = \tanh(x_t w_{xh} + (R_t \odot H_{t-1}) w_{hh} + b_h)$$

Hadamard (elementwise) product operator

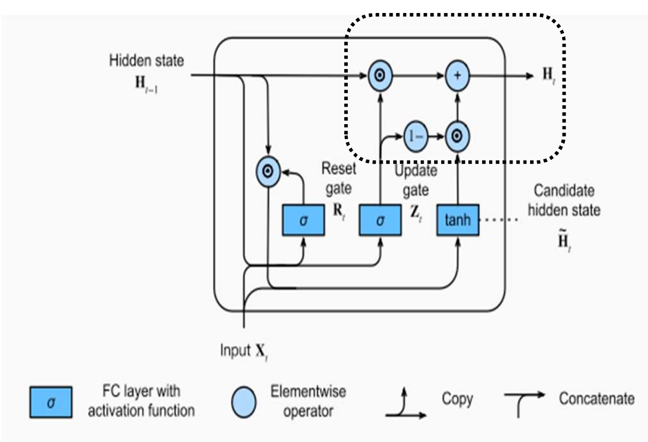
- The candidate hidden state combines information from the input and the previous hidden state
- GRU computes a **candidate hidden state** at each time step, which is then used to update the **hidden state** for the following time step

9

SMU

1.) Theoretical Background

Gated Recurrent Unit (GRU)



Hidden step for timestamp t

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \widetilde{H}_t$$

Hadamard (elementwise) product operator

- In the hidden step, the update gate Z_t determines how much of the new hidden state H_t matches the previous hidden state H_{t-1}, and the new candidate state $\widetilde{H_t}$.
- When the value of the output gate Z_t is 1, we retain the hidden step of the previous timestamp H_{t-1}, and we ignore the input x_t. This would be equivalent to excluding information related to timestep t.
- On the other hand, when the value of the output gate Z_t is 0, the hidden state of timestamp t would approach the candidate hidden state $\widetilde{H_t}$.

10

10

SMU

1.) Theoretical Background

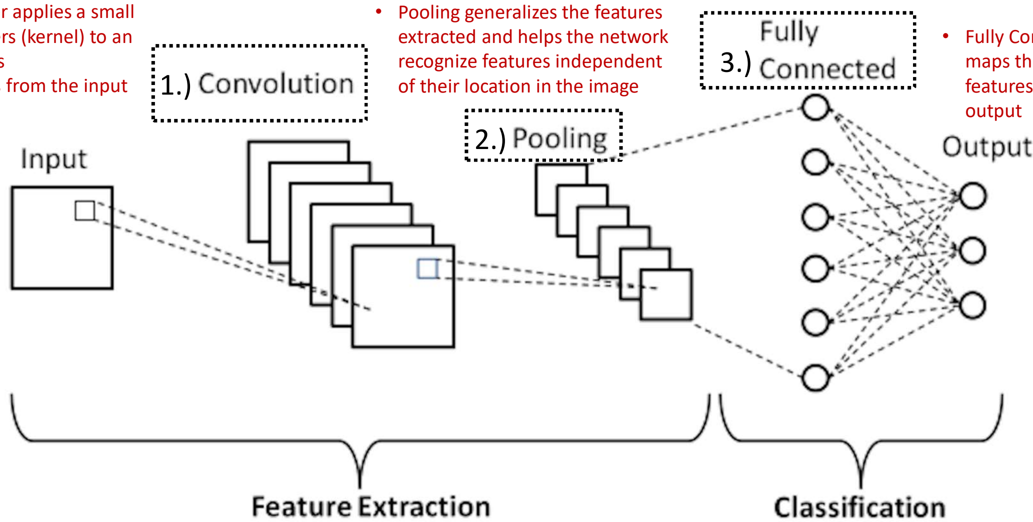
Convolutional Neural Network (CNN)

Deep learning model used to process data, such as images, that have grid patterns

- Convolution layer applies a small grid of parameters (kernel) to an array of numbers
- Extracts features from the input image

- Pooling generalizes the features extracted and helps the network recognize features independent of their location in the image

- Fully Connected layer maps the extracted features to the final output



11

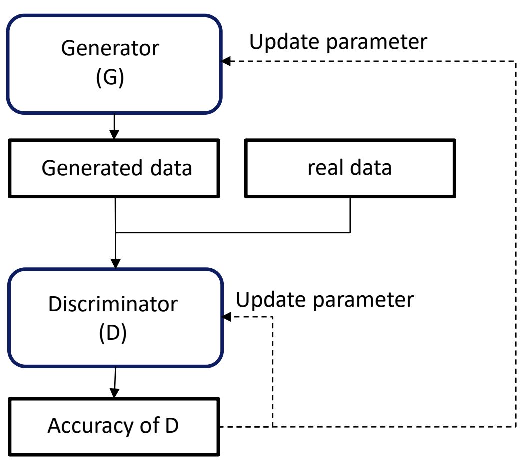
11

SMU

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | **Architecture of GAN** | **Training process in GAN** | **Advantages / Disadvantages**



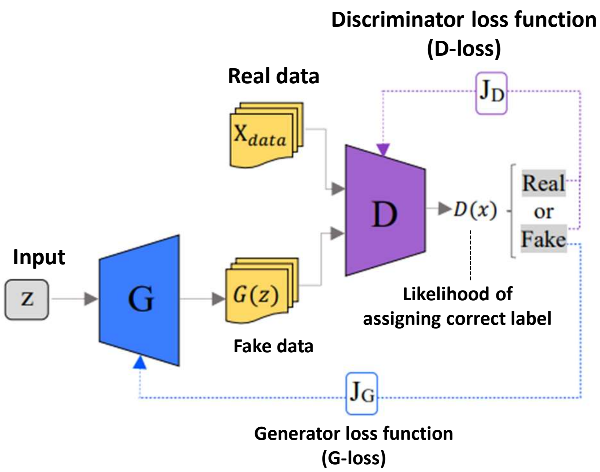
- GAN is a deep learning framework developed by Ian Goodfellow in 2014.
- It consists of two separate neural networks that compete against each other in a game-like scenario. The first network, known as **the generator network**, tries to create fake data that looks real.
- The second network, known as **the discriminator network** that tries to distinguish between data generated by the generator (fake data) and real data.
- Both networks are trained until they reach a point where they cannot improve because the discriminator is unable to differentiate between the real and the fake data anymore. At that point we say that the generator has arrived at an optimal point in generating data.

12

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | **Architecture of GAN** | Training process in GAN | Advantages / Disadvantages



- The generator and the discriminator can be neural networks, convolutional neural networks (CNN), and recurrent neural networks (RNN).
- The choice of a neural network depends on the specific task and data. For instance, CNN is commonly chosen for both generator and discriminator for image-related tasks.

Log - likelihood

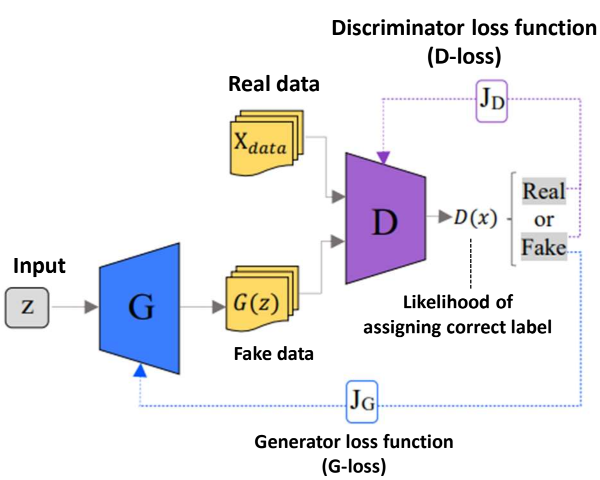
- The discriminator takes x or G(z) as input and returns a binary judgment as to whether the data is real (return 1) or fake (return 0).
- We define the log-likelihood that the discriminator will accurately categorize real data as $\boldsymbol{logD(x_i)}$ and the log-likelihood that the discriminator would correctly categorize generated samples as fake as $\mathbf{log}(\mathbf{1}-\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z_i})))$
- The generator aims to minimize $\mathbf{log}(\mathbf{1}-\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z_i})))$ while the discriminator aims to maximize both $\boldsymbol{logD(x_i)}$ and $\mathbf{log}(\mathbf{1}-\boldsymbol{D}(\boldsymbol{G}(\boldsymbol{z_i})))$

13

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | **Architecture of GAN** | Training process in GAN | Advantages / Disadvantages



- From the definition of $log\big(D(x)\big)$ and $log(D(G(z_i)))$, GAN can be modeled as a two-player minimax game with simultaneous training of both generator and discriminator network. Minimax GAN Loss is regarded as an optimization strategy in two-player games whereby each player reduces their losses or increases the costs of the other player. Minimax refers to minimizing the loss in the generator and maximizing the loss in the discriminator.
- The discriminator seeks to maximize the probability of assigning proper labels to the data. On the contrary, the generator seeks to generate a series of samples that can fool discriminator.
- The equation explaining minimax game is given as:

$$\min_{G}\max_{D}\,(G,D)=E_{x\sim p_{data}(x)}\big[log\big(D(x)\big)\big]+E_{z\sim p_z(z)}[log(1-D(G_{zi}))]$$

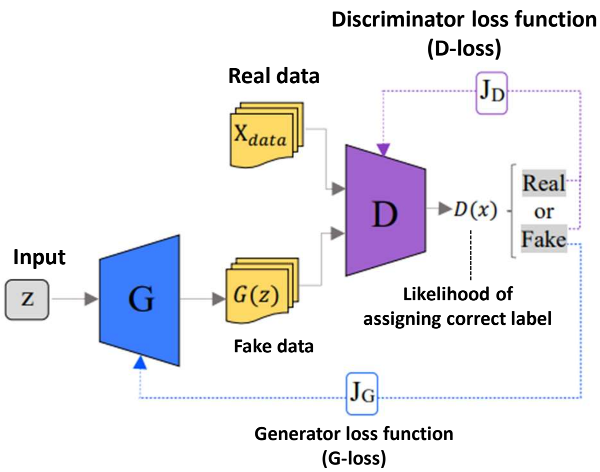
14

SMU

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | **Architecture of GAN** | Training process in GAN | Advantages / Disadvantages



- However, according to the original paper of GAN, the minimax equation may not provide sufficient gradient for the generator to learn well. Early in learning, when the generator is poor, the discriminator can reject samples with high confidence because they are clearly different from the training data.

- So instead of minimizing the likelihood of the discriminator correctly determining fake data, the objective of the generator has changed to maximizing the likelihood of the discriminator classifying fake data as real ($\boldsymbol{logD(G(z_i))}$)

$$J_G = \overset{\text{minimize}}{G - loss} = -\frac{1}{m}\sum_{i=1}^{m} \overset{\text{maximize}}{logD(G(z_i))}$$

- And Dloss is defined as

$$J_D = \overset{\text{minimize}}{D - loss} = -\frac{1}{m}\sum_{i=1}^{m} \overset{\text{maximize}}{\log D(x_i)} - \frac{1}{m}\sum_{i=1}^{m} \overset{\text{maximize}}{\log(1 - D(G(z_i))}$$

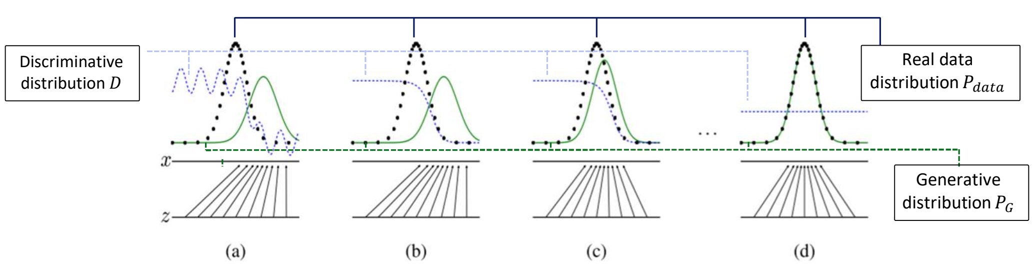
15

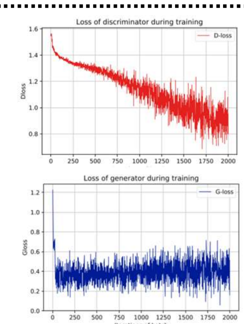
SMU

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | Architecture of GAN | **Training process in GAN** | Advantages / Disadvantages





Steps of training

1.) The generator generates the fake data. The discriminator tries to distinguish between fake and real data. The first Gloss and Dloss are calculated (figure a)

2.) The discriminator then is updated by Dloss through backpropagation and becomes better in distinguishing according to the last generated (fake) data (figure b)

3.) The generator is then updated by Gloss through backpropagation and produces the fake data that has distribution closer to the - Real data distribution (figure c).

4.) After several steps of training, the generator and discriminator will reach a point where the distribution of fake data equals to the distribution of real data and the discriminator is unable to differentiate between the two distributions (D(X) = 1/2)

16

1.) Theoretical Background

Generative Adversarial Networks (GAN)

What is GAN? | **Architecture of GAN** | **Training process in GAN** | **Advantages / Disadvantages**

Advantages

1.) Realistic data

Because GAN is made up of 2-neural networks: a generator and a discriminator. As the two networks compete with each other, the generator becomes better at creating realistic data.

2.) Unsupervised learning

GANs can be trained without labeled data, making them suitable for unsupervised learning tasks, where labeled data is scarce or difficult to obtain.

Disadvantages

1.) Difficult to train

This is because the two networks in a GAN (the generator and the discriminator) are constantly competing against others, which can make training unstable and slow.

2.) Mode collapse

GANs can be vulnerable to mode collapse, which is when the generator only produces a limited number of output instead of the variety that is desired.

17

2.) Datasets and features

- Datasets

- **Target variable :** Closing price of Apple Inc. (AAPL), eBay Inc. (EBAY) and Starbucks Corporation (SBUX)
- **Time period :** January 1, 2013 to December 31, 2022 (10-year period)
- **Data source :** Yfinance open library (Python)

- Types of feature

Structure-type variable commonly used as features in stock price prediction

Basic features

- Historical information including Open, High, Low, Close and Volume

Technical Indicators

- Information that is subtracted from historical price by mathematical equations

Fundamental Indicators

- Macroeconomic factors and microeconomic factors

18

2.) Datasets and features

Features used in the analysis :

- **Basic features :** Previous studies suggested that closing price is the most significant feature for forecasting closing price
  Data source: Yfinance open library (Python)

- **Technical Indicators :** 5 technical indicators were calculated from closing price and used as features to capture information regarding price movement.

1.) Simple Moving Average (SMA21) : $= \frac{P_t + P_{t-1} + P_{t-2} + \ldots + P_{t-20}}{21}$

2.) Relative Strength Index (RSI) : $= 100 - (100/1 + RS)$ where : $RS = \frac{Average\ gain\ (14\ days)}{Average\ loss\ (14\ days)}$

3.) Moving Average Convergence Divergence (MACD) : $= EMA\ (12\ days) - EMA\ (26\ days)$

4.) Upper Bollinger Band : $= SMA21 + (2\ x\ standard\ deviation)$

5.) Lower Bollinger Band : $= SMA21 - (2\ x\ standard\ deviation)$

19

2.) Datasets and features

Features used in the analysis :

- **Fundamental Indicators :**
  - **Microeconomic factors :** 4 financial metrics from quarterly financial statement that demonstrate the profit-making capability and the financial health of the company were included as features.

Data Source: Wharton Research Data Service (WRDS)

1.) $\frac{\textbf{Earning per share (TTM)}}{\textbf{Closing price}}$

2.) $\frac{\textbf{Net profit margin (TTM)}}{\textbf{Closing price}}$

3.) $\frac{\textbf{Book value per share}}{\textbf{Closing price}}$

4.) $\frac{\textbf{Debt-to-Equity ratio}}{\textbf{Closing price}}$

- All metrics were divided by closing price to make them move on a daily basis.
- The report date was utilized instead of end-of-quarter date to reflect the date when the public becomes aware of the updated fundamental data, ensuring that the analysis incorporates the most current and publicly available information

20

2.) Datasets and features

Features used in the analysis :

- **Fundamental Indicators :**
  - **Macroeconomic factors :** We included 5 macroeconomic factors that are commonly used as economic indicators.
    Data source: Yfinance open library (Python)

1.) Crude oil price : When the global economy is robust and growing, demand for oil tends to increase as industries expand and consumer activities rise.

2.) Gold price : Investors tend to flock to gold as a store of value and a hedge against inflation or economic downturns. Therefore, an increase in the demand for gold and a rise in its price may indicate concerns about the stability of financial markets and the overall economy.

3.) S&P500 index
4.) NASDAQ100 index : An upward movement in stock index values are generally associated with economic expansion and investor confidence, while a downward movement may signal concerns about economic growth or stability.

5.) Federal funds rate : The federal funds rate is a key interest rate that influences borrowing costs throughout the economy, serving as an indicator of the fed's monetary policy stance and it also provides insights about the overall economic condition.

21

2.) Datasets and features

Features used in the analysis :

No.	Feature type	Feature	Data source
1.)	Basic feature	Closing price	yfinance library
2.)	Technical indicators	SMA (21 day)	Calculated from closing price
3.)		RSI (14-day window length)	
4.)		MACD (12 short, 26 long)	
5.)		Upper Bollinger band	
6.)		Lower Bollinger band	
7.)	Fundamental indicators	Earnings per share / closing price	WRDS
8.)		Book value per share / closing price	
9.)		Debt-to-equity ratio / closing price	
10.)		Net profit margin / closing price	
11.)		Federal interest rate	yfinance library
12.)		Gold price	
13.)		Crude oil price	
14.)		S&P500 index	
15.)		Nasdaq100 index	

22

3.) Methodology

Data Preprocessing

Import Data → Specify x and y variables → Normalize Data → Reshape Data → Split into Train and Test Data

Import Data

Features		
Historical Closing Price	Earnings per share / quarter	Book Value
Debt-to-equity Ratio	Net Profit Margin	Gold
Crude Oil	S&P 500	NASDAQ-100
Federal interest rate	MACD	RSI
SMA 21	Bollinger Upper Band	Bollinger Lower Band

23

3.) Methodology

Data Preprocessing

Import Data → Specify x and y variables → Normalize Data → Reshape Data → Split into Train and Test Data

x and y variables

x-variable		
Historical Closing Price	Earnings per share / quarter	Book Value
Debt-to-equity Ratio	Net Profit Margin	Gold
Crude Oil	S&P 500	NASDAQ-100
Federal Reserve System	MACD	RSI
SMA 21	Bollinger Upper Band	Bollinger Lower Band

y-variable
Historical Closing Price

- The x-variable consists of the closing price of the stock and 14 features
- The y-variable consists of the closing price only.

24

3.) Methodology

Data Preprocessing

Import Data → Specify x and y variables → Normalize Data → Reshape Data → Split into Train and Test Data

Normalization

- Normalization was applied to both the 16 input features and the targeted variable
- This was to ensure that all input features are on a consistent scale
- Also, it would promote faster convergence and mitigating issues related to vanishing or exploding gradients
- And enhance the model's generalization to unseen data.
- We had used the MinMax Scaler as it is known to effectively manage outlier data. Thus, ensuring a robust scaling transformation.

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad \text{and} \quad Y_{scaled} = \frac{Y - Y_{min}}{Y_{max} - Y_{min}}$$

25

3.) Methodology

Data Preprocessing

Import Data → Specify x and y variables → Normalize Data → Reshape Data → Split into Train and Test Data

Data Reshaping

- A predictive model was developed using a sequence of three-day historical data (T-1, T-2, T-3) to forecast the stock price on the subsequent day (T).
- During the reshaping, the 3-day features' data was formatted as input, and the stock price of the following day served as the output for model training and testing.
- The reshaping of data resulted in a total of 2503 observations for both X (features) and y (targeted variable).

Train & Test Split

- 70% : Train and 30%: Test
- Training Set consisted of historical price data spanning from the start of 2013 to the end of 2019, a total of 1,752 observations
- Testing Set consisted of data from the beginning of 2020 to the end of 2022, a total of 751 observations.

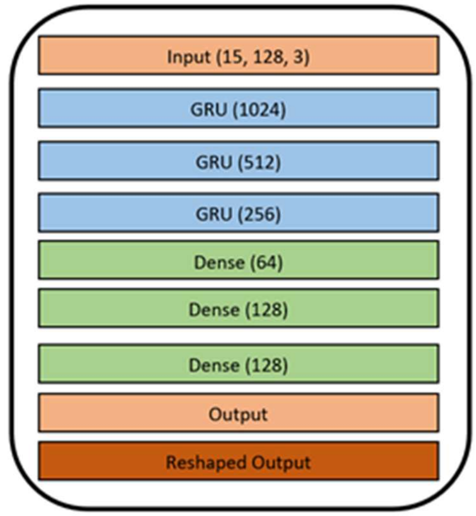
26

3.) Methodology

Generator

Explanation

- Used GRU as the Generator
- The generator processes 3D input data comprising of
  - ❑ Features
  - ❑ batch size with input-step
- Produces output with batch size and output-step dimensions.
- For optimal generator performance, we applied three layers of GRU, where the number of neurons are in descending manner, 1024, 512 and 256.
- Subsequently, three Dense layers were incorporated, and output was produced.



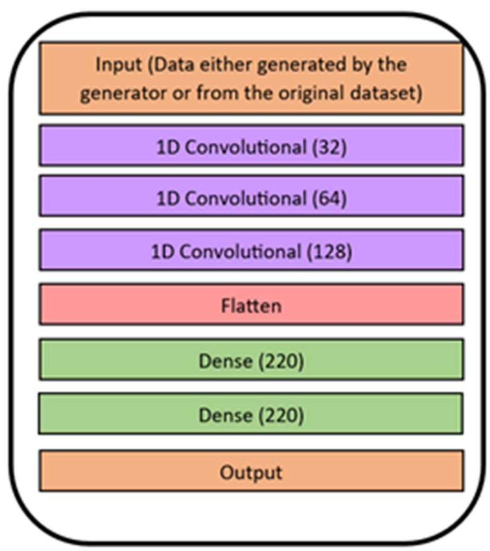
27

3.) Methodology

Discriminator

Explanation

- Used CNN as discriminator
- With the primary objective of distinguishing between genuine and fabricated input data.
- The input data would either be data generated from the generator or from the original dataset.
- Consists of three distinct 1D Convolution layers, each containing 32, 64, and 128 neurons, respectively.
- Additionally, there are two Dense layers incorporated at the end, with neuron counts of 220 each.
- Throughout these layers, the activation function employed is the Leaky Rectified Linear Unit (ReLU), except for the output layer.
- In this case, for GAN, the output layer uses the Sigmoid activation function to yield a scalar output of either 0 (indicating fake data) or 1 (indicating real data).

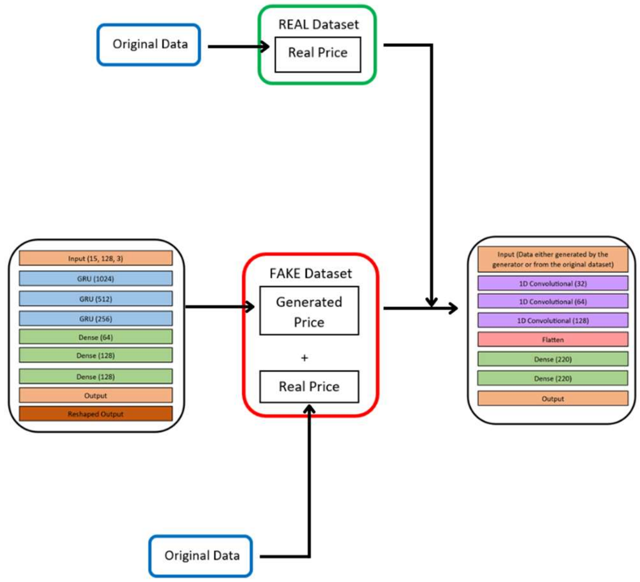


28

3.) Methodology

GAN Model Structure

- To form our GAN Model, we incorporated and merged the architectural designs of both our generator and discriminator
- Within the framework of our GAN structure, we employed the cross-entropy method to compute losses for both the generator and discriminator.



29

3.) Methodology

GAN Model Structure

Cross Entropy

- Cross-entropy is a mathematical idea that finds extensive applications in a range of domains, such as information theory, machine learning, and statistics.
- is a particularly common choice as a loss function in machine learning models
- especially when dealing with classification tasks.
- serves as a metric to gauge the dissimilarity between the predicted probability distribution and the actual probability distribution of outcomes.

Types of Cross Entropy

- 2 types of cross-entropy which are:
  - ❑ Binary cross-entropy, and
  - ❑ Categorical cross-entropy
- In our model, we have used binary cross-entropy (log loss) as we were dealing with 2 possible outcomes. (e.g. 0 or 1).

Binary Cross Entropy

- Binary cross entropy measures the dissimilarity between the predicted probability distribution and the true binary labels of a dataset.
- compares each of the predicted probabilities to actual class output which can be either 0 or 1.
- then calculates the score that penalizes the probabilities based on the distance from the expected value, which means how close or far from the actual value

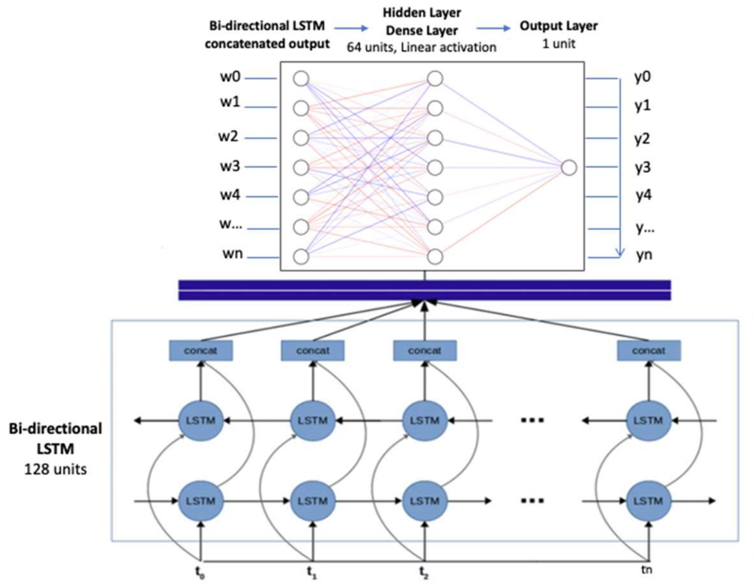
30

SMU SINGAPORE MANAGEMENT UNIVERSITY

3.) Methodology

LSTM Model Structure

- Bidirectional LSTM with 128 units captures patterns in both forward and backward directions. Each LSTM layer has its own parameter
- The output of the Bi-LSTM layer comprises of the hidden states from both the forward and backward LSTM layers at each time step combined through concatenation.
- First dense layer (64 units) acts as an intermediary representation and final dense layer (1 unit) maps this representation to the output space.
- Training Details:
  - ✓ Adam optimizer with a learning rate of 0.001 is used.
  - ✓ Mean Squared Error (MSE) is the chosen loss function.
  - ✓ Training executed for 50 epochs with a batch size of 64.



31

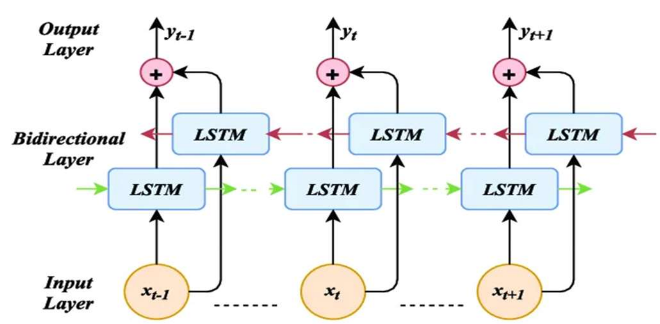
SMU SINGAPORE MANAGEMENT UNIVERSITY

3.) Methodology

Bi-directional Long short-term memory networks (Bi-LSTM)

- Bi-LSTM consists of two LSTM layers and uses bidirectional processing to process sequential data simultaneously in two directions, both forward and backward.
- Both the forward and backward LSTM, based on the current input and the previous hidden state and memory cell, will update its memory cell and compute its hidden state.
- Eventually, the hidden states of each LSTM layer are combined at each time step, when both forward and backward processes are complete.
- The benefit of Bi-LSTM is that it captures not only the context that comes before a specific time step but also the context that follows. It allows to capture richer dependencies in the input sequence

Backward processing: the input sequence is fed into the backward LSTM layer from the last to the first step.



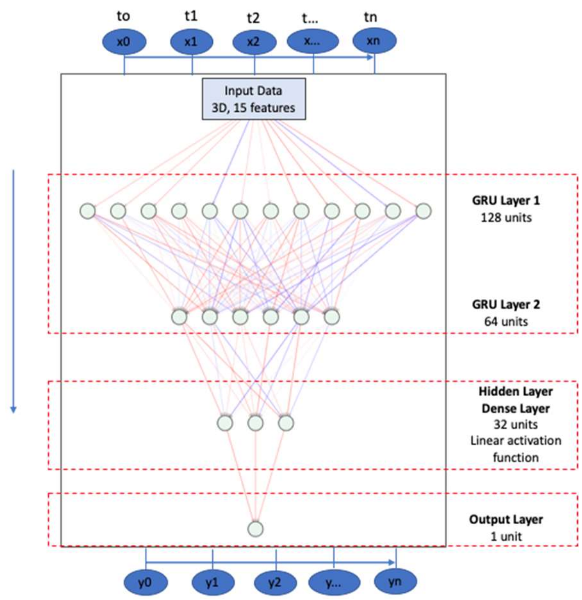
Forward processing: Input sequence is fed into the forward LSTM layer from the first to the last step.

32

3.) Methodology

GRU Model Structure

- Two GRU layers, 1st layer (128 units), 2nd layer (64 units)
- The model, utilizing two GRU layers to captures both local and global patterns in sequential data
- First dense layer (32 units) captures intermediary representations.
- Final dense layer (1 unit) maps this representation to the output space.
- Training Details:
  - ✓ Adam optimizer with a learning rate of 0.0001 is used.
  - ✓ Mean Squared Error (MSE) is the chosen loss function.
  - ✓ Training executed for 50 epochs with a batch size of 64.



33

4.) Experimental results

Performance measurements :

Root Mean Squared Error (RMSE)
- Measures the square root of average squared differences between predicted and actual values.
- Lower RMSE indicates better model performance.

Mean Absolute Error (MAE)
- Calculates average absolute differences between predicted and actual values.
- Lower MAE values signify better performance.

Mean Absolute Percentage Error (MAPE)
- Evaluates errors in terms of relative size of predicted values.
- Lower MAPE values indicate more accurate predictions.

Mean Squared Logarithmic Error (MSLE)
- Useful for target variables with exponential growth patterns.
- Lower MSLE values indicate better accuracy on a logarithmic scale.

R-squared (R2)
- Assesses how much of the dependent variable's variance is predictable.
- Higher R2 indicates a better fit of the model to the data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(\widehat{Y}_i - Y_i\right)^2}{n}}$$

$$MAE = \frac{\sum_{i=1}^{n}\left|\widehat{Y}_i - Y_i\right|}{n}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\widehat{Y}_i - Y_i}{Y_i}\right| \times 100$$

$$MSLE = \frac{\sum_{i=1}^{n}\left(\log\left(\widehat{Y}_i + 1\right) - \log(Y_i + 1)\right)^2}{n}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}$$

34

SMU SINGAPORE MANAGEMENT UNIVERSITY

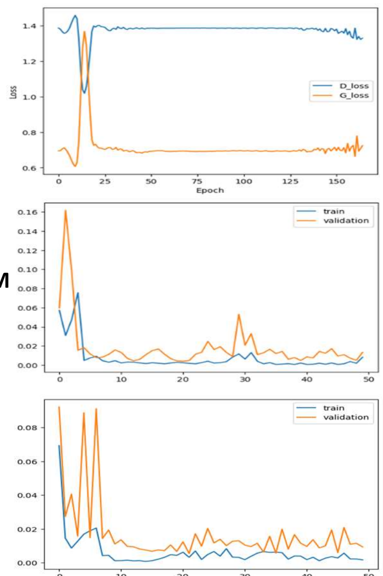
4.) Experimental results

Loss function curve

GAN

LSTM

GRU



Observation of Loss Functions:
- LSTM and GRU: Observable patterns with decreasing and stabilized losses.
- GANs: Distinct, counterintuitive behaviour.

Dynamics in GANs:
- Unlike LSTM and GRU, D-loss plateaus at a relatively high point.
- Competitive interplay between generator (G-loss) and discriminator (D-loss).

Adversarial Dynamics in GANs:
- Improvements in one component leads to higher losses in the other.
- Both discriminator and generator losses converge to stable values.

Equilibrium and Training Success:
- Generator learns to produce realistic samples.
- Discriminator unable to effectively distinguish between real and generated data.

Factors for Successful Training:
- Hyperparameter Adjustment.
- Choosing the right Network Architecture.
- Continuous Monitoring of Loss Functions.

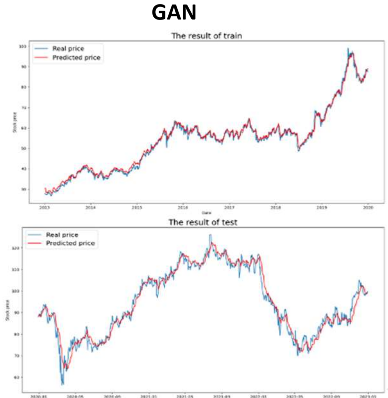
35

SMU SINGAPORE MANAGEMENT UNIVERSITY

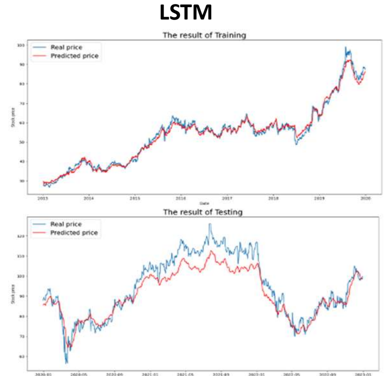
4.) Experimental results

Stock Price Prediction - SBUX

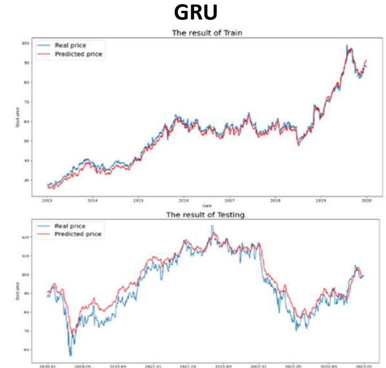
GAN



LSTM



GRU



SBUX	GAN		LSTM		GRU	
	training	testing	training	testing	training	testing
RMSE	1.27	3.01	1.7	5.758	1.83	4.825
MAE	0.95	2.2	1.324	4.557	1.59	3.866
MAPE	0.0194	0.0249	0.0249	0.0458	0.033	0.0455
MSLE	0.00063	0.001	0.000916	0.00436	0.001577	0.00325
R^2	0.993	0.962	0.987	0.86	0.985	0.902

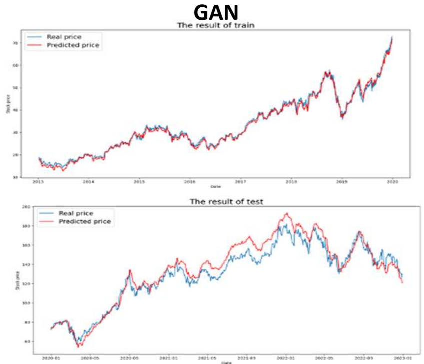
36

SMU SINGAPORE MANAGEMENT UNIVERSITY

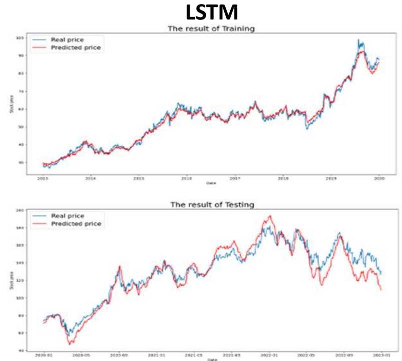
4.) Experimental results

Stock Price Prediction - APPL

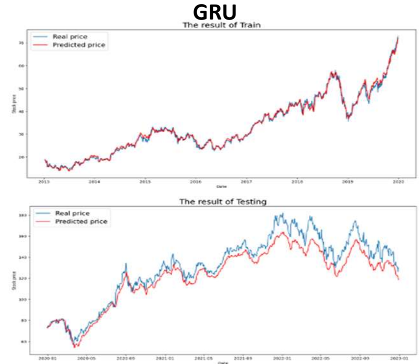
GAN



LSTM



GRU



AAPL	GAN		LSTM		GRU	
	training	testing	training	testing	training	testing
RMSE	1.079	8.3975	3.811	9.401	0.8843	9.96
MAE	0.847	6.796	3.484	7.309	0.6446	8.314
MAPE	0.0286	0.05	0.1348	0.0574	0.02	0.06
MSLE	0.00135	0.00351	0.02992	0.00608	0.00064	0.005
R^2	0.992	0.925	0.9096	0.906	0.9951	0.895

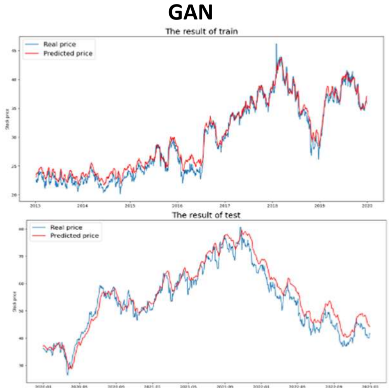
37

SMU SINGAPORE MANAGEMENT UNIVERSITY

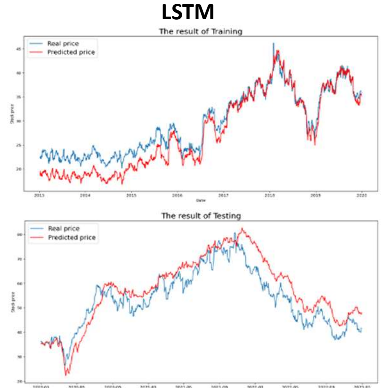
4.) Experimental results

Stock Price Prediction - EBAY

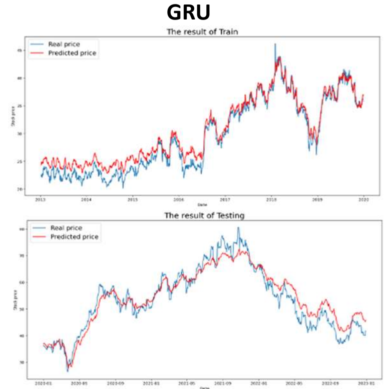
GAN



LSTM



GRU



EBAY	GAN		LSTM		GRU	
	training	testing	training	testing	training	testing
RMSE	1.079	3.198	2.575	6.095	1.491	3.3557
MAE	0.873	2.63	2.1072	5.25	1.2518	2.7523
MAPE	0.0327	0.0518	0.0844	0.1022	0.0493	0.0546
MSLE	0.00152	0.00364	0.0126	0.0125	0.00325	0.0042
R^2	0.973	0.927	0.846	0.734	0.948	0.9192

38

4.) Experimental results

- GAN consistently achieves higher testing accuracy compared to other models. Notable superiority in capturing time series representations.
- GRU demonstrates superior training performance in specific scenarios compared to GAN and LSTM.
- GRU has comparable training results with GAN, but GAN excels in testing accuracy.
- LSTM and GRU lagss behind and GAN in overall accuracy.
- GAN's testing accuracy is consistently higher across various datasets.
- GAN architecture excels in capturing time series representations.
- GAN architecture indicates superior accuracy compared to traditional LSTM and GRU models.

SBUX	**GAN**		**LSTM**		**GRU**	
	training	testing	training	testing	training	testing
RMSE	1.27	3.01	1.7	5.758	1.83	4.825
MAE	0.95	2.2	1.324	4.557	1.59	3.866
MAPE	0.0194	0.0249	0.0249	0.0458	0.033	0.0455
MSLE	0.00063	0.001	0.000916	0.00436	0.001577	0.00325
R^2	0.993	0.962	0.987	0.86	0.985	0.902

AAPL	**GAN**		**LSTM**		**GRU**	
	training	testing	training	testing	training	testing
RMSE	1.079	8.3975	3.811	9.401	0.8843	9.96
MAE	0.847	6.796	3.484	7.309	0.6446	8.314
MAPE	0.0286	0.05	0.1348	0.0574	0.02	0.06
MSLE	0.00135	0.00351	0.02992	0.00608	0.00064	0.005
R^2	0.992	0.925	0.9096	0.906	0.9951	0.895

EBAY	**GAN**		**LSTM**		**GRU**	
	training	testing	training	testing	training	testing
RMSE	1.079	3.198	2.575	6.095	1.491	3.3557
MAE	0.873	2.63	2.1072	5.25	1.2518	2.7523
MAPE	0.0327	0.0518	0.0844	0.1022	0.0493	0.0546
MSLE	0.00152	0.00364	0.0126	0.0125	0.00325	0.0042
R^2	0.973	0.927	0.846	0.734	0.948	0.9192

39

5.) Conclusion & Future Studies

GAN Outperforms in Accuracy:

- GAN network excels in capturing time series representations. It demonstrates a remarkable ability to capture representations within time series data.
- Demonstrates superior accuracy compared to traditional LSTM and GRU models.

Complex Architecture:

- GAN framework has a complex architecture.
- Extensive trial and error needed during hyperparameter tuning.

Future Studies:

- Future studies can explore more effective approaches in determining the model combination and hyperparameters used.
- Aim to streamline the hyperparameter tuning process to optimize GAN framework performance.
- Include feature selection process by using algorithms such as DecisionTreeClassifier and ExtraTreeClassifier

Significant Potential:

- Despite challenges, GANs offer significant potential for advancing predictive modeling in financial contexts.
- Ongoing research and refinement can unlock even greater capabilities.

40