BT4222 Final Report

Title: Stock Price Forecasting with Omni Channel Features

Time Series Group 5

Name	Matriculation Number
Tan Si Xuan	A0257152H
Chua Ler Han	A0258561X
Simon See Yongsheng	A0249451B
Ian Wong Han Li	A0255278U
Tay Xinyu, Zandra	A0239429U
Lee Jie Long, Bryan	A0233537E

Abstract

The stock market, an expansive public market for trading shares of publicly listed companies, is profoundly influenced by a confluence of factors including political, economic conditions, social sentiments, and financial performance of companies. This complexity presents significant challenges for investors aiming to make informed decisions based on stock price forecasting, which if inaccurate, can lead to substantial financial losses. To address this challenge, our project developed a machine learning model designed to predict future closing prices of stocks, thereby facilitating more informed investment decisions through insights into potential market movements.

Business Issue and Machine Learning Problem Alignment

- Business Issue: Given the multifaceted influences on stock prices, there is a crucial need for tools that can analyze and predict market trends more accurately to minimize investment risks.
- Machine Learning Problem: We address this issue by utilizing ML models to predict the future closing prices of stocks, employing omni-channel features that reflect a broad spectrum of influencing factors.

Datasets and Models

- We integrated multiple datasets including historical stock prices, macroeconomic indicators, and sentiment analysis data, ensuring a rich and comprehensive dataset for analysis.
- The machine learning model was adapted from established predictive models, incorporating advanced algorithms for time series forecasting and sentiment analysis.

Achievements and Highlights

- We explored a mix of traditional algorithms and deep learning models. While traditional models did not perform as expected, our deep learning models showed promising performance.
- Notably, our Google TFT model demonstrated promising results despite the limited existing research on its application in stock prediction. It performed comparably to GRU, a widely utilized model in stock prediction, achieving RMSE of 11.1362 and R-squared of 0.9293 in an 80-10-10 train-validation-test split ratio, and RMSE of 9.2242 and R-squared of 0.5677 during prediction of COVID pandemic, accurately predicting stock close price.

1. Proposal Review

Proposed Contribution	Achievements	Satisfactory level
Exploring different ML models for stock prediction.	This includes both traditional and deep learning algorithms, allowing us to assess the strengths and weaknesses of each model in the context of stock price forecasting.	Satisfactory
Incorporating omnichannel features for stock prediction.	This includes traditional stock metrics, economic indicators, news and report sentiments, enhancing our predictive modeling. Missing NaN values were addressed based on domain-specific considerations. Based on the frequency and use case of the features, NaNs were filled using context-appropriate strategies like backfilling, forward filling, linear interpolation or creating placeholder values, ensuring the completeness and reliability of our dataset.	Satisfactory
Employing GRU; claimed to be promising for stock price prediction by research.	We have employed GRU as one of our models, achieving promising performance metrics with an RMSE of 11.1362 and an R-squared of 0.9293.	Satisfactory
Utilisitising ensemble methods (bagging, boosting, stacking).	We employed 2 types of ensemble models – bagging with RFR, and boosting with XGBoost. Due to their poor performance we did not extend our use of ensemble techniques to stacking.	Satisfactory
Explore the effectiveness of machine learning models in predicting stock performance during unpredictable scenarios (black swan events).	We achieved promising performance in predicting stock prices during the COVID-19 pandemic (a black swan event) using our Google TFT model, with an RMSE of 9.2242 and an R-squared value of 0.5677. This performance surpassed that of all other models, including our LSTM and GRU deep learning models, which had RMSE values of 42.2334 and 12.8423 and R-squared values of -8.2983 and 0.1444 respectively.	Exceeds Expectations

2. Data Description

Overarching feature	Description	Statistics / categories	Other information	Source
Open, High, Low, Close	Daily stock price[1] in USD.	'Close' statistics: Mean: 64.65 Std: 82.10 Min: 0.97 Max: 305.29	Nans were handled by forward filling & linear interpolation.	yfinance package
Volume	Daily Volume[1] of the number of shares transacted.	Mean: 7.88e+07 Std: 4.82e+07 Min: 1.33e+07 Max: 6.56e+08	Nans were handled by forward filling & linear interpolation.	yfinance package
Relative Strength Index (RSI)	(Engineered feature) Derived daily from exponential weighted moving average[1] of upward & downward price changes over 14 days, then normalizing. Used to identify overbought or oversold conditions in an asset.	Mean: 54.52 Std: 13.29 Min: 16.35 Max: 89.09	Nans were handled by forward filling & linear interpolation.	yfinance package
Volatility	(Engineered feature) Daily volatility[1] derived from standard deviation of the Close prices over a 14-day rolling window, reflecting the degree of variation in price movements.	Mean: 1.57 Std: 2.20 Min: 0.02 Max: 11.76	Nans were handled by forward filling & linear interpolation.	yfinance package
Dividends	Daily payments[1] made to shareholders, representing a portion of the company's earnings, in USD. Statistics are all 0 due to how tech companies typically do not issue dividends, and how the metric is aggregated by taking median.	Mean: 0 Std: 0 Min: 0 Max: 0	Nans were handled by forward filling & linear interpolation.	yfinance package
Stock Splits	Daily stock splits[1] where a company divides its existing shares into multiple shares to boost the liquidity. Represented as a ratio. Statistics are all 0 due to how tech companies typically do not issue	Mean: 0 Std: 0 Min: 0 Max: 0	Nans were handled by forward filling & linear interpolation.	yfinance package

[1] aggregated by taking the median of 4 different stocks (NFLX, AAPL, MSFT, AMZN)

	stock splits, and how the metric is aggregated by taking median.			
20 Day Simple Moving Average	Daily arithmetic average of a stock's close prices[1] over the last 20 days, to smooth price data & identify trends.	Mean: 63.11 Std: 81.00 Min: 0.96 Max: 302.27	Nans were handled by forward filling & linear interpolation.	yfinance package
Rate of Change (ROC)	Momentum indicator[1] measuring % change in price between the current price and the price a certain number of periods ago, calculated daily.	Mean: 1.33 Std: 5.48 Min: 28.95 Max: 35.31	Nans were handled by forward filling & linear interpolation.	yfinance package
T10Y2Y	Daily 10-year treasury constant maturity minus 2-year treasury constant maturity in percent. Treasury data obtained from the US Treasury Department.	Mean: 1.12 Std: 0.98 Min: -1.08 Max: 2.91	Nans were handled by forward filling and backward filling.	Federal Reserve Economic Data (fredapi)
CCSA	Weekly number of people who have made continued claims in the US, seasonally adjusted. Continued claims is the number of people who have already filed an initial claim and who have experienced a week of unemployment and then filed a continued claim to claim benefits for that week of unemployment.	Mean: 3157847 Std: 2370549 Min: 1339000 Max: 23130000	Nans were handled by forward filling and backward filling.	Federal Reserve Economic Data (fredapi)
DGORDER	Monthly manufacturers' new orders for durable goods in USD million, seasonally adjusted.	Mean: 222365.38 Std: 31871.61 Min: 146323 Max: 301262	Nans were handled by forward filling and backward filling.	Federal Reserve Economic Data (fredapi)
Ticker sentiment scores	Includes: ● NFLX_Sentiment Score ● AAPL_Sentiment Score ● MSFT_Sentiment Score ● AMZN_Sentiment Score Scores ranges from -1.0 to 1.0	AAPL: Mean: 0.649780 Std: 0.112548 Min: -1.000000 Max 1.000000	Nans were handled by imputing the median score of the respective ticker sentiment scores.	Polygon.ai API
Ticker sentiment labels	Includes: ● NFLX_Sentiment Label ● AAPL_Sentiment Label ● MSFT_Sentiment Label	AMZN: Positive: 6677 Non-positive: 77	Nans were handled based on the imputed median score for the respective tickers.	Polygon.ai API

	● AMZN_Sentiment Label Categories are Positive, Neutral, Negative before binning; and Positive, Non-positive after binning in feature engineering.			
RF metrics	The total count of sentences and total sentiment score for positive, negative, and all sentences derived from the Risk Factor (RF) section. Includes ● RF_positive_count ● RF_negative_count ● RF_count ● RF_total_positive_score ● RF_total_negative_score ● RF_avg_sentiment_score	**RF counts (Mean):** positive: 8.420 negative: 87.708 total: 220.817 **RF_avg_sentiment_score:** Mean: -0.340 Std: -0.340 Min: -0.643 Max: -0.224	Nans were handled by forward filling and backward filling. Mean values were used for tickers that shared the same reporting period.	SEC EDGAR API
MDA metrics	The total count of sentences and total sentiment score for positive, negative, and all sentences derived from the MD&A section. Includes ● MDA_positive_count ● MDA_negative_count ● MDA_count ● MDA_total_positive_score ● MDA_total_negative_score ● MDA_avg_sentiment_score	**MDA counts (Mean):** positive: 41.292 negative: 32.630 total: 272.572 **MDA_avg_sentiment_score:** Mean: 0.0404 Std: 0.0738 Min: -0.2021 Max: 0.206	Nans were handled by forward filling and backward filling. Mean values were used for tickers that shared the same reporting period.	SEC EDGAR API
Date	Date corresponding to the information in each row.	Range: 2003-01-01 to 2024-12-31	Split into: year, month and day for time series analysis.	yfinance package

Table 1. Data Description

3.1 Models

Traditional models XGBoost and Random Forest Regression (RFR) performed poorly on our dataset (Table 3). Both models utilized hyperparameter tuning with time series cross validation, and lagged features of close prices at 1, 7, 14, and 30 days ago.

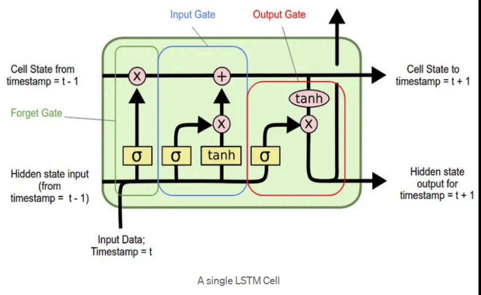
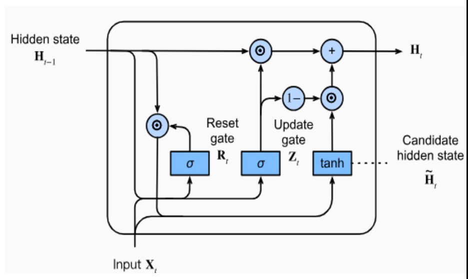
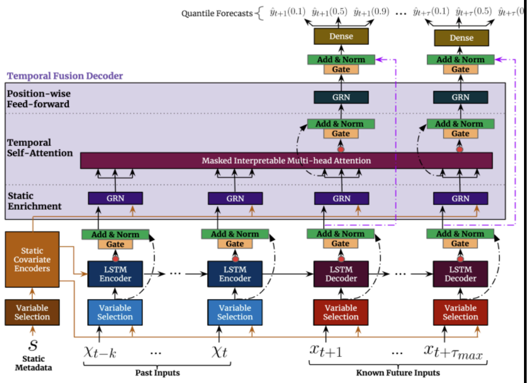
LSTM (J, 2021)	**GRU** (Dive Into Deep Learning, n.d.)	**Google TFT** (Lim et al., 2021)
 Input Gate Output Gate Cell State from timestamp = t - 1 Cell State to timestamp = t + 1 Forget Gate tanh σ Hidden state input (from timestamp = t - 1) Hidden state output for timestamp = t + 1 Input Data; Timestamp = t A single LSTM Cell	 Hidden state H_{t-1} H_t Reset gate R_t Update gate Z_t 1− σ tanh Candidate hidden state $\tilde{H}_t$ Input X_t	 Quantile Forecasts Dense Add & Norm Gate Temporal Fusion Decoder Position-wise Feed-forward GRN Temporal Self-Attention Masked Interpretable Multi-head Attention Static Enrichment Static Covariate Encoders LSTM Encoder LSTM Decoder Variable Selection Static Metadata Past Inputs Known Future Inputs

Table 2: Deep Learning Model's Architecture

The performance of our deep learning models improved. Long Short-Term Memory (LSTM) was used because of its accurate predictions for stock prices. Moghar & Hamiche (2020) had done univariate forecasting with stock open prices using an 80-20 train-test split, reporting a low MAE and RMSE loss.

Gated Recurrent Unit (GRU) is a popular RNN alternative in modeling sequential data that was utilized. GRU yielded a lower MSE than LSTM by 43.2% in a multivariate time series forecasting of the S&P 500 index research by Li & Pan (2021). GRU is also more computationally efficient and less prone to overfitting on smaller datasets than LSTM (Srivatsavaya, 2023), a fit for our project's small data size.

Google Temporal Fusion Transformer (TFT) (Lim et al., 2021) is a multi-horizon time series forecasting model that captures long-term dependencies of temporal features. Google TFT separates static and temporal features. Then, the model distinguishes and weighs features based on their importance with the Variable Selection Network (VSN). The input is then encoded and passed through an Interpretable Multi-head Attention that identifies long-range dependencies throughout the input data's different time steps, before being decoded for predicted output. Google TFT was selected because of its good performance in long-term forecasting for financial instruments (He et al., 2023).

3.2 Performance

Due to the large number of features relative to the dataset size, we observed negative R^2 in both the training and validation datasets, meaning underfitting has occurred.

	XGBoost (Baseline model)	**XGBoost**	**Random Forest**	**Random Forest (with Lagged Variables)**
Val RMSE	0.2296	0.2348	0.2255	0.2007
Val R^2	-0.7212	-2.6445	-2.3633	-1.664

Table 3. Model performance for XGBoost and Random Forest models

We utilized hyperparameter tuning to determine the optimal parameters for each model. We then tested the models with different split ratios, and a train-test split against the COVID-19 pandemic for January to June 2020 (Table 4), to determine accuracy during various events. Each specific row in input data has 10 previous timesteps, representing the rows' most recent 10 days of stock data. Each model predicts 3 future timesteps of stock close price for each row and measures the average RMSE and average R^2 score across the three timesteps. Google TFT overall performed the best for different split ratios and COVID-19.

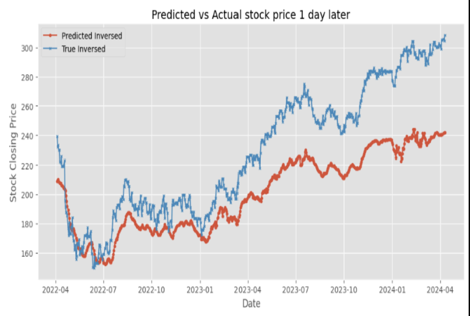
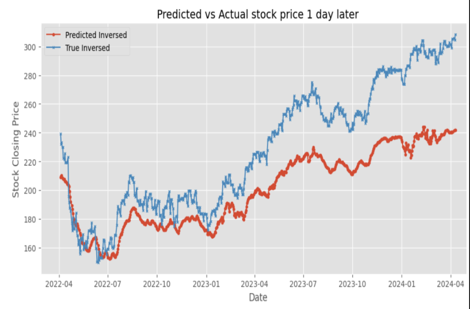
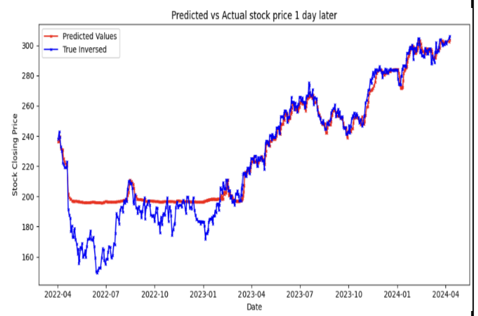
	LSTM				**GRU**				**Google TFT**			
Split ratio	**70-15-15**	**80-10-10**	**90-5-5**	**COVID**	**70-15-15**	**80-10-10**	**90-5-5**	**COVID**	**70-15-15**	**80-10-10**	**90-5-5**	**COVID**
Val RMSE	29.3603	26.2498	13.6051	-	13.0890	17.4180	9.6090	-	6.1598	12.5484	8.9283	-
Val R^2	0.3964	0.3936	0.4310	-	0.8801	0.7327	0.7158	-	0.9710	0.8292	0.7552	-
Test RMSE	72.3988	33.9336	7.4870	34.0893	25.2373	19.7392	10.8479	20.7108	12.1403	11.1362	8.9578	9.2242
Test R^2	-2.8389	0.4028	0.8827	-5.0612	0.5343	0.7980	0.7533	-1.2448	0.8829	0.9293	0.8246	0.5677
Graph	 Test set, predicted versus actual stock price one day later, 80-10-10 Split				 Test set, predicted versus actual stock price one day later, 80-10-10 Split				 Test set, predicted versus actual stock price one day later, 80-10-10 Split			

Table 4. Performance of deep learning models for different train-val-test split ratios and COVID

4. Contribution & Justification

Contribution		**Low**	**Medium**	**High**
(Dataset) Use valuable and high-quality new datasets, including integrating existing datasets, crawling or retrieving data, etc.	Effort			✅
	Effectiveness		✅	
Low: Use existing datasets with very limited features and data points (i.e., a few thousands) Medium – High: newly collected datasets or integrated datasets from multiple sources considering the richness of the information, the amount of data points, and the creativity in finding surprisingly useful resources.				
Creativity in feature engineering	Effort			✅
	Effectiveness		✅	
Low: generate new features based on descriptive statistics (e.g., mean, variance, etc.) Medium – High: generate new features by applying relevant theories, domain knowledge, representative learning, social network analysis, or other machine learning or econometrics methods. Adopting methods proposed by high-quality research papers is also encouraged.				
Design or adaptation of new ML methods/architecture or the integration of existing methods with a balance of resource and cost	Effort			✅
	Effectiveness			✅

Low: ensemble learning by straightforwardly integrating multiple ML models. Medium – High: Design innovative architecture or pipeline or adapt ML methods, which have changed the way how learning and prediction will be conducted. Adopting methods proposed by high-quality research papers is also encouraged.				
Creativity in understanding or further explaining the prediction results and performance, in identifying and resolving the gap or confusion between ML outcome and business decision making	Effort			✅
	Effectiveness			✅
Low: describe your results and performance without extra analysis or without further insights Medium – High: Use data analytics methods (econometrics, network analysis, etc. what you have learned from the other courses) and additional ML methods to clarify, distinguish, idenTIfy the bias of, or evaluate your ML output towards effective business decision making.				
Any aspects that are distinct from the above...	Effort			
	Effectiveness			
Objective evidence is needed to sufficiently justify your self-evaluation.				

Table 5: Self-evaluation table

Dataset Contribution

High Effort Justification

Complexity of Data Acquisition:

- Managing Multiple Diverse Sources: Utilizing the yfinance API provides a solid foundation by accessing historical stock price data. Augmenting this with macroeconomic indicators, news content, and sentiment analysis from other multiple third-party APIs (fredapi, polygon.ai, SEC EDGAR) significantly increases the complexity due to varying data formats, access protocols, and rate limits. Managing multiple APIs, each with its unique challenges and requirements, requires substantial technical expertise and effort, ensuring all data is timely and accurately retrieved. This ensures a rich dataset for machine learning models to derive patterns.

Data Integration Challenges:

- Data Alignment: The integration process involves meticulous alignment of data points that differ in data granularity. For instance, aligning quarterly financial reports with daily stock prices and monthly economic indicators presents significant technical challenges. A lot of care was taken for the sentiment features in the data integration, especially important for the Temporal Models. Lagging of news and quarterly sentiments was crucial to make sure that there was no reverse causality. Increasing the complexity by multiple folds.
- Data Consistency and Quality: Ensuring data consistency across multiple sources demands extensive and multiple iterations of data cleaning, deduplication, and normalization processes. This is critical for maintaining data integrity and reliability, which are essential for accurate analysis.

Medium Effectiveness Justification

Predictive Limitations:

Despite the comprehensive nature of the dataset, we rate the effectiveness of our dataset as medium. We observe from our studies that only some of our deep learning models perform relatively well. We attribute it to several factors that temper its impact, despite the dataset's depth and innovative composition.

- Weak Feature Correlations:

    - Not all features incorporated into the dataset, such as specific macroeconomic indicators or sentiment scores, demonstrate strong correlations with the target variable—stock price movements. This lack of significant correlations can detract from the dataset's overall predictive power, as these features do not contribute effectively to the model’s accuracy.
- Inherent nature of Stock Prices:
    - Stock prices are temporal and non-stationary in nature. They are subjected to complex volatility which changes rapidly due to external events (some of which may be intangible i.e behavioral economics), making it challenging for most machine learning models to accurately model and predict movements. There could be many more external factors that contribute to the daily movements that we cannot possibly capture in our dataset.

Design/Adaptation of new Machine Learning Models

High Effort Justification

Diverse experimentation of Machine Learning models:

- In our project we experimented with a wide variety of Machine learning models from simple to complex; Classical Machine Learning models (XGBoost, Random Forest Regression) and Deep Learning Models (LSTM, GRU, and Google's Temporal Fusion Transformer.) This allows us to gain a greater understanding of the nature of financial prediction as well as learn different machine learning model’s application, methodology, and attributes.

Advanced Adaptation of Deep Learning Models:

- Implementing novel models such as Google's Temporal Fusion Transformer (TFT) requires an understanding of both the underlying theory, practical feasibility and implementation challenges.

Resource Allocation:

- Significant computational resources over long periods of time were necessary for training complex models like LSTM, GRU, or TFT, especially when working with large datasets typical in financial contexts.

High Effectiveness Justification

Improved Predictive Accuracy:

- Our project proved that classical machine learning models fail to do well. We have also proven that LSTM, GRU, and Google TFT models are effective for stock price predictions due to their ability to capture long-term dependencies and non-linear patterns within volatile financial data.

Handling Complex Data Structures:

- These advanced deep learning models are particularly suited to the multifaceted nature of stock data, which includes a mix of static and time-varying features, making them more effective than simpler models that might not fully capture these complexities.

Creativity in understanding prediction results and performance, in identifying and resolving the gap or confusion between ML outcome and business decision-making

High Effort Justification

- We determined underfitting by observing the R^2 for both the training, validation and testing set. Since both the training and validation R^2 were negative for traditional ML models, that means the models were underfitted because the model fits very poorly to the data.
- We also checked for the fit of the deep learning models in different proportions of the train-validation-test split.
- There is no gap between ML outcome and business decision-making, as the project predicts quantitative stock price for the next 1-3 days.

High Effectiveness Justification

- Negative R^2 informs us that traditional ML models are a poor fit for our project,
- For deep learning models, it also serves as an indicator of the optimal train-validation-test split occurring, which is where the R^2 had a positive value with a decently high magnitude (>0.5).

5. References

Dive Into Deep Learning. (n.d.). *9.1. Gated Recurrent Units (GRU) — Dive into Deep Learning 0.16.6 documentation*. D2l.ai. Retrieved April 19, 2024, from https://d2l.ai/chapter_recurrent-modern/gru.html

Goehry, Benjamin, et al. "RANDOM FORESTS FOR TIME SERIES." RANDOM FORESTS FOR TIME SERIES, vol. 0, no. 0, 2021, p. 21. Random Forests for Time Series, https://hal.science/hal-03129751/file/Block_bootstrap_for_random_forests.pdf, Accessed 28 February 2024.

He, K., Zheng, L., Yang, Q., Wu, C., Yu, Y., & Zou, Y. (2023). Crude oil price prediction using temporal fusion transformer model. *Procedia Computer Science*, *221*, 927–932. ScienceDirect. https://doi.org/10.1016/j.procs.2023.08.070

J, R. T. J. (2021, September 10). *LSTMs Explained: A Complete, Technically Accurate, Conceptual Guide with Keras*. Analytics Vidhya. https://medium.com/analytics-vidhya/lstms-explained-a-complete-technically-accurate-conceptua l-guide-with-keras-2a650327e8f2

Li, Y., & Pan, Y. (2021). A novel ensemble deep learning model for stock prediction based on stock prices and news. *International Journal of Data Science and Analytics*, *13*, 139–149. Springer Link. https://doi.org/10.1007/s41060-021-00279-9

Lim, B., Arık, S. Ö., Loeff, N., & Pfister, T. (2021). Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, *37*(4), 1748–1764. https://doi.org/10.1016/j.ijforecast.2021.03.012

Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science*, *170*, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Qu, H., & Kazakov, D. (2019, March 19). Detecting Causal Links between Financial News and Stocks. White Rose Research Online. Retrieved March 3, 2024, from https://eprints.whiterose.ac.uk/143829/

Srivatsavaya, P. (2023, July 24). *LSTM vs GRU*. Medium. https://medium.com/@prudhviraju.srivatsavaya/lstm-vs-gru-c1209b8ecb5a#:~:text=LSTM%3A% 20LSTM%20typically%20has%20more