Neural Network-based Risk Analysis and Prediction for Short Selling using Financial Market Indicators

S M Rayeed
Rensselaer Polytechnic Institute
Troy, NY, USA
rayees@rpi.edu

Professor Lirong Xia
Rensselaer Polytechnic Institute
Troy, NY, USA
xial@rpi.edu

ABSTRACT

This paper explores the application of neural networks in analyzing the short-selling tragedy of financial market. Leveraging some of the key financial indicators including MACD (Moving Average Convergence Divergence) and RSI (Relative Strength Index), our study employs Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models to predict stock market trends. The aim is to identify optimal time-frames for strategic short selling. Through extensive experimentation, the results showcase the effectiveness of utilizing deep learning techniques along with traditional financial indicators for enhanced risk assessment and informed decision-making in the dynamic world of trading, especially short selling. Our analysis demonstrates the remarkable accuracy of the GRU model, which has a low Mean Squared Error (MSE) of 0.00081. The LSTM model, on the other hand, has a marginally higher MSE of 0.00657 despite being very effective in identifying the market trends. Both models, however, perform poorly in case of detecting abrupt and usual changes in stock prices.

KEYWORDS

Short Selling, Financial Forecasting, Long-short term memory, Gated Recurrent Units

1 INTRODUCTION

In the dynamic world of stock market, mastering the skill of short selling has become a fascinating yet challenging task for traders and investors. In contrast to the traditional strategy of buying low and selling high, short selling involves betting against a stock's rise—a money-making technique fraught with invaluable complications and risks. The strategy is to borrow shares and sell with the hope of repurchasing them at a reduced cost, where the difference becomes the trader's profit . This process involves a risk of facing infinite loss, which arises from the unpredictable nature of markets. As stocks can skyrocket unexpectedly, those who shorted face potential losses that exceed their initial investment. This inherent risk factor necessitates advanced tools and methodologies for effective navigation, especially considering how erratic and unpredictable stock markets can be.

In this study, we have integrated financial market indicators with deep neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to assess the risk in short-selling trading strategy and predicting an optimal time of such investment. Our work provides a detailed investigation of the potential benefits of combining state-of-the-art deep learning methods with conventional financial market indicators, such as Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI). With combination of financial knowledge and neural networks, our models perform very well in identifying the market trends, while GRU and LSTM performing with a minuscule Mean Squared Error (MSE) of 0.00081 and 0.00657, respectively. But one of the major limitations of these models are they cannot perform up to the mark while there is abrupt spikes or drops in stock prices. We understand that anticipating sudden changes in stock prices is difficult, but still this is an important point that needs to be improved.

2 RELATED WORKS

Several research works have explored the domain of stock market prediction using various machine learning (ML) methodologies. In [8], ML algorithms like Radial Basis Function (RBF) and Support Vector Machine (SVM) were employed to predict the Karachi Stock Exchange (KSE). Deep neural networks integrated with Principal Component Analysis (PCA) were utilized in [7] to analyze the performance of Google stock data from NASDAQ. In [6], the authors used Support Vector Machine (SVM) for predicting stock prices across large and small capitalization. LSTM models were applied in [4] for tasks such as estimating survival probabilities over time and default prediction using time-series accounting data. The dataset comprised 8262 public companies listed on the American stock market between 1999 and 2018. In [2], the authors proposed a modified deep reinforcement learning approach for automated stock trading, with performance evaluation on the U.S. stock market's Dow Jones Industrial Average (DJIA) index constituents. In [1], Support Vector Regression (SVR) was used to predict the direction of price changes for a small set of DJIA stocks, compared against logistic regression analysis predictions. Some of the works distinctively focused on short-selling include [9], which explored short-selling analysis using a dataset of 2687 stocks from different industry sectors traded on the NASDAQ and NYSE, employing a logistic regression-support vector machine ensemble model. [5] utilized an LSTM algorithm for forecasting price movements on 5 stocks, incorporating only 2 indicators, MACD and RSI. Notable works in financial market indicators include [3], where 93 market indicators were reviewed. The authors emphasized on predicting stock market returns based on these indicators and highlighted the risk of data snooping.

While existing works have individually explored financial indicators, short-selling analysis, and neural-network-based stock analysis, there is a notable gap in research that combines these elements. The incorporation of these diverse methodologies is crucial for a comprehensive understanding of stock market dynamics, offering a holistic approach that holds the potential to enhance prediction accuracy and provide more robust insights into the complexities of financial markets.

3 FINANCIAL MARKET INDICATORS

Analyzing market dynamics and predicting market trends require a nuanced approach through precise mathematical formulations. In such cases, indicators serve as interpreters by converting data into actionable insights and offer these insights into trends, volatility, and trading activity. For example, Moving Averages smooth out short-term market fluctuations, revealing underlying trends; while the RSI provides insights into the magnitude of recent price changes. Another example is the market volatility, or the degree of variation in stock prices. This is evaluated through indicators like Standard Deviation, Bollinger Bands, and other volatility indicators. High standard deviation or expansive Bollinger Bands signal heightened volatility, contrasting with low values indicative of a more stable market. Altogether, these indicators offer a snapshot of the market's recent behavior, stock strength, volatility, and trading volume, thereby helping investors and traders to make informed decisions, and gauging potential opportunities or risks. **Table 5** contains a complete list of financial indicators used in this work.

4 DATA COLLECTION AND PROCESSING

This section describes the process of collecting the stock data and process it in such a way that could be fitted into our models. The process goes through several phases, including data collection, financial indicators calculation, data cleaning, normalization, transformation into time series data, and finally dividing into training, testing, and validation sets.

4.1 Data Collection

In the first step, we meticulously curated historical data, which served as the fundamental resource of our study. From the S&P500, we have chosen the top 20 companies based on their weight for data gathering. The selected stock data was collected in a comprehensive process, leveraging the robust capabilities of Python's yfinance library and carefully verifying that no values are missing. We collected the stock data from January 1, 2020 till December 31, 2022 for all the companies. The initial dataset has 5 price components - high, low, open, close, and adjusted close.

4.2 Financial Indicators Calculation

In the next step, using the stock data and the equations from Table 5, we calculated the financial indicators. In total, we had 25 financial indicators, mainly serving four purposes: identifying market trend, assessing the volatility, strength, and trading volume of a stock. Figure Figure 1 gives a graphical demonstration of some of the indicators. After calculating the financial indicators, we merged the values to our data. However, one noteworthy thing about this merge is, it creates some some null values in the data. For example, a 7-day moving average (ma7) cannot have anything before day 7, therefore has its first six values set to NaN; which is completely fair and justified, but at the same time the dataset should also not have null values in an ideal scenario. So our next task was to process the data and then normalize it.

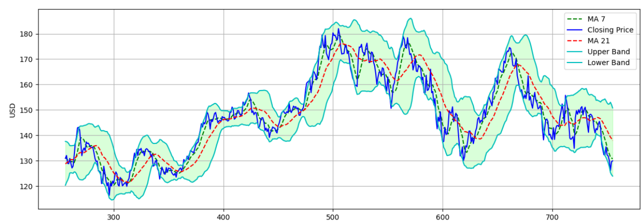

Figure 1: Bollinger Bands & Moving Averages : Tesla('TSLA')

4.3 Data Cleaning and Normalization

To get the data ready for analysis, we started by cleaning it and making sure it's all set for predictions. One issue we tackled was missing values (NaN) that was just discussed above. To fill these gaps, we used a mix of forward fill and backward fill, because this combination is designed to work well for time-series data, where each data point is linked to what came before and what comes after. After that, we dropped the columns associated with stock prices except the closing price (i.e. 'open,' 'high,' 'low,' and 'adjusted close' were dropped), because these prices are usually quite similar (or somewhat close) to the closing price we are trying to predict, and keeping them in the data might make the data biased and lead to overfitting. Finally to avoid the data being biased towards any indicators, we normalized the data using the MinMaxScaler tool. This helps create a more balanced dataset.

4.4 Data Split and Conversion into Time-series

With our data being ready, we moved on to split it for training and testing. We followed the usual split ratio of 80% for training and 20% for testing. But one thing to mention is we did not shuffle the data around. Since this is a continuous time-series data, the order is very important for our models learn from the historical sequence of stock prices, catching on to trends and patterns. It is like learning from the past to make better predictions in the future. We divided the data into training and testing sets maintaining the chronological order. Next, we converted our data into a dynamic time series with a 14-day window, which equips our models to not just see the bigger picture but also discern subtle fluctuations and changes over these shorter periods, enhancing their ability to make precise predictions.

5 PREDICTION

This section analyzes the reason of choosing GRU ad LSTM over traditional Recurrent Neural Networks (RNN), and also describes the process of designing the models' architectures, tweaking the parameters and fine-tuning the models for the stock prediction.

5.1 Why GRU and LSTM?

When it comes to handling sequences of data over time, RNNs are a common choice. They're good at learning from what happened before, but they face a challenge called the vanishing gradient problem, where the gradients become so extremely small that they impede the effective updating of weights and thereby losing important information. This challenge makes it hard for RNNs to understand faraway connections in a long sequence of data — just like our dataset. To tackle this, we have used GRU and LSTM, which are two specialized RNNs, designed to mitigate the problem.

GRU: GRU skillfully controls information flow through its Update Gate, finding a balance between incorporating new insights and preserving valuable data. By preventing gradients from disappearing during backpropagation, this tactical control solves the challenges associated with vanishing gradient. GRU's Reset Gate improves memory performance, which helps with efficient long-term learning and protects against the vanishing gradient problem.

LSTM: In LSTMs, the Cell State serves as a dynamic memory bank that manages information across time steps. To tackle the vanishing gradient problem, the Forget Gate prioritizes crucial data, discarding less important information. Simultaneously, the Input Gate regulates the influx of new data into the memory bank. The Output Gate acts as a discerning overseer, identifying and safeguarding meaningful patterns in time-series data.

5.2 Designing Model Architecture

For the GRU model architecture, we've crafted a robust structure with three stacked layers. Each layer plays a specific role, consisting of 32, 64, and 32 units respectively. This thoughtful design aims to capture intricate patterns within the time-series data. To prevent overfitting and enhance generalization, we've included a dropout layer with a rate of 0.10. The output layer, a single unit, is responsible for predicting closing prices—the ultimate goal of our financial forecasting model. In terms of evaluating our model's performance, we employ the mean squared error as our chosen loss function. The optimizer at play is Adam, with a default learning rate of 0.001. The model is summarized in Table 1.

Table 1: GRU Model Architecture

Layer	Output Shape	Param #
GRU	(None, 14, 32)	5,760
GRU	(None, 14, 64)	18,816
GRU	(None, 32)	9,408
Dropout	(None, 32)	0
Dense	(None, 1)	33

In the LSTM model, we have applied similar structures as GRU, with three layers of 32, 64, and 32 units, followed by a dropout layer with a rate of 0.20 and a single unit output layer. The optimization and loss function also remained the same for this model. Summary of LSTM model architecture is shown below in Table 2.

Table 2: LSTM Model Architecture

Layer	Output Shape	Param #
LSTM	(None, 14, 32)	7,552
LSTM	(None, 14, 64)	24,832
LSTM	(None, 32)	12,416
Dropout	(None, 32)	0
Dense	(None, 1)	33

5.3 Prediction results and Observations

This section discusses about the outcomes of running the designed models on the stock data. After completing the model design, a total of 50 epochs were executed for each model, with a batch size of 32. The execution time for the GRU model was 50.55 seconds, and for LSTM, it took even less, approximately 45.83 seconds. Both models exhibited commendable performance metrics, including Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2); as shown in Table 3 for Tesla 'TSLA':

Table 3: Model Performance Metrics: Tesla

Metric	GRU	LSTM
Mean Squared Error (MSE)	0.00081	0.00657
Mean Absolute Error (MAE)	0.02309	0.07205
R-squared (R2)	0.95499	0.63370

For a better visual representation of the models' learning patterns, training and validation loss were plotted in Figure 2.

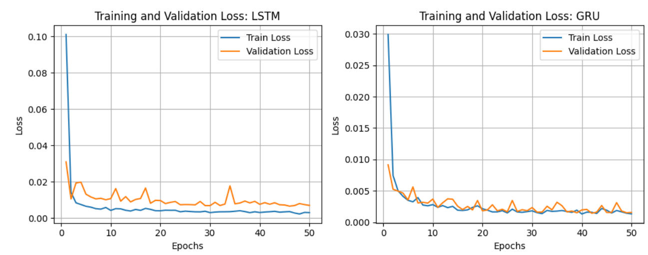

Figure 2: Train & Validation Loss: LSTM(L), GRU(R) for Tesla

Subsequently, the predicted values by both GRU and LSTM were graphically compared against the actual values as displayed in Figure 3. Both fugures show the performaces of Tesla stock only.

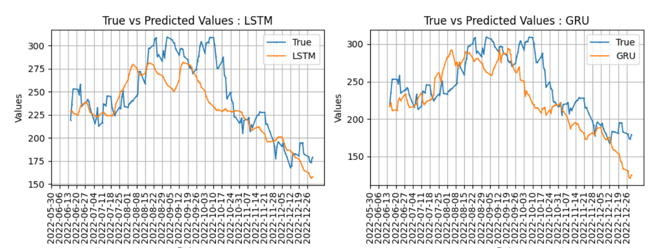

Figure 3: True vs Predicted Price: LSTM(L), GRU(R) for Tesla

Observations: Both models performed very well in identifying the broader market trends on a longer timeframe, capturing gradual rises and falls with commendable accuracy. But the models showed poor results in promptly recognizing and adapting to abrupt changes. For a better graphical representation, a graph showing the actual and predicted results for Amazon's stock prices is taken. In Figure 4, we clearly see that the models showcased their ability to accurately predict the overall downward trend from October 2022 to December 2022.

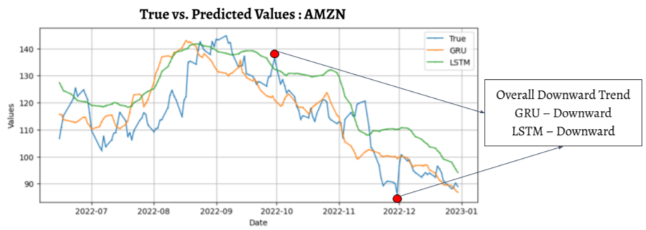

Figure 4: Model Strength : Overall Trend Identification

And, as shown in Figure 5, both models fails to identify sudden spikes or drop in prices (there was a sharp decline followed by a sudden spike in late November 2022).

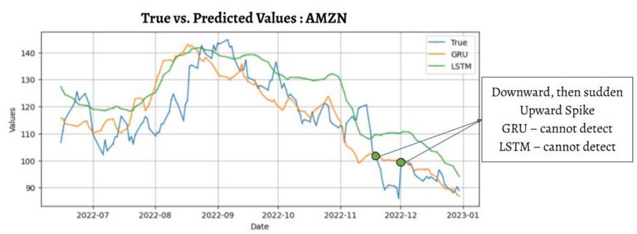

Figure 5: Model Weakness : Sudden Spike/Drop Detection

One plausible explanation for this limitation is that sometimes stock data changes unpredictably without following a discernible trend or pattern, making it challenging to anticipate. The market indicators used are designed to identify patterns, but their performance diminishes when faced with unexpected changes.

Among the two models, as suggested by the performance metrics, GRU performs marginally better. For a comprehensive evaluation, the process was repeated for the top 20 companies listed in the S&P500[1]. Figure 6 demonstrates a graphical representation of the models' performance for top nine companies among those 20.

5.4 Prediction for Future

After running the models on across the time-frame of the stock data (from 2020 to 2022), we used the models for predicting stock market values. And here are the results I got for Tesla stock prices for the next 7 business days shown in Table 4:

Table 4: Prediction for Next 7 Days

Date	True	GRU	LSTM
2023-01-03	108.099998	126.377035	135.053304
2023-01-04	113.639999	127.326618	135.464113
2023-01-05	110.339996	128.295097	135.822320
2023-01-06	113.059998	129.253288	136.128511
2023-01-09	119.769997	130.191691	136.388681
2023-01-10	118.849998	131.108669	136.611007
2023-01-11	123.220001	132.004609	136.804119

[1]Source: S&P500

Upon initial observation, it might appear that both models are overestimating values by a significant margin. However, deeper analysis revealed a sharp decline from Dec 30, 2022, to Jan 03, 2023 (from $123.18 in Dec 30 to $108.09 in Jan 03); which neither LSTM nor GRU could identify. This scenario highlights the limitation of our models' in predicting future stock prices, which is 'the case of abrupt changes'. In a nutshell, our models showcased their strengths and limitations in identifying market trends and responding to sudden unanticipated fluctuations, respectively.

6 CONCLUSION AND FUTURE WORK

In stock market trend prediction, our research utilizing financial market indicators and designing neural-networks has demonstrated promising outcomes. The models, carefully built and adjusted, have proven their ability to predict changes in stock prices. The GRU model, especially, is reliable for long-term investors, giving them a solid foundation for smart decision-making. But as mentioned above, the models might not be as reliable if the stock market unexpectedly shifts. And the market is always shifting due to unfavorable news, world politics, political developments, and economic ups and downs. Because of this, we need a more detailed approach.

Looking Beyond Numbers: Understanding the limits of financial indicators, our next step is exploring new possibilities. We plan to make the models stronger by adding sentiment analysis, through news articles and social media, especially Twitter, to understand the general economic feeling and the specific feelings around individual companies. The goal is to focus on the hidden, intangible factors that affect the market, and to make the models more resilient, understanding unexpected changes in the market. To conclude, our current achievements show the potential of AI-driven models in understanding stock markets. The future holds the promise of discovering new aspects in predictive analytics, making these models essential allies in the ever-changing financial world.

REFERENCES

[1] David E Allen, Robert J Powell, and Abhay K Singh. 2012. Machine learning and short positions in stock trading strategies. In *Handbook of Short Selling*. Elsevier, 467–478.

[2] Eeshaan Asodekar, Arpan Nookala, Sayali Ayre, and Anant V Nimkar. 2022. Deep reinforcement learning for automated stock trading: Inclusion of short selling. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 187–197.

[3] Jiali Fang, Yafeng Qin, and Ben Jacobsen. 2014. Technical market indicators: An overview. *Journal of behavioral and experimental finance* 4 (2014), 25–56.

[4] Gianfranco Lombardo, Mattia Pellegrino, George Adosoglou, Stefano Cagnoni, Panos M Pardalos, and Agostino Poggi. 2022. Machine Learning for Bankruptcy Prediction in the American Stock Market: Dataset and Benchmarks. *Future Internet* 14, 8 (2022), 244.

[5] Bhaskar V Patil, Deepali M Gala, and Mr Sanjay A Jadhav. 2023. BUY TODAY SELL TOMORROW [BTST] A SHORT SELLING TECHNIQUE USING PREDICTION ALGORITHM. *The Online Journal of Distance Education and e-Learning* 11, 1 (2023).

[6] V Kranthi Sai Reddy. 2018. Stock market prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)* 5, 10 (2018), 1033–1035.

[7] Ritika Singh and Shashi Srivastava. 2017. Stock prediction using deep learning. *Multimedia Tools and Applications* 76 (2017), 18569–18584.

[8] Mehak Usmani, Syed Hasan Adil, Kamran Raza, and Syed Saad Azhar Ali. 2016. Stock market prediction using machine learning techniques. In *2016 3rd international conference on computer and information sciences (ICCOINS)*. IEEE, 322–327.

[9] Timothy Vervaet. [n.d.]. EMPIRICAL EVALUATION OF SHORT SELLING STRATEGIES. ([n. d.]).

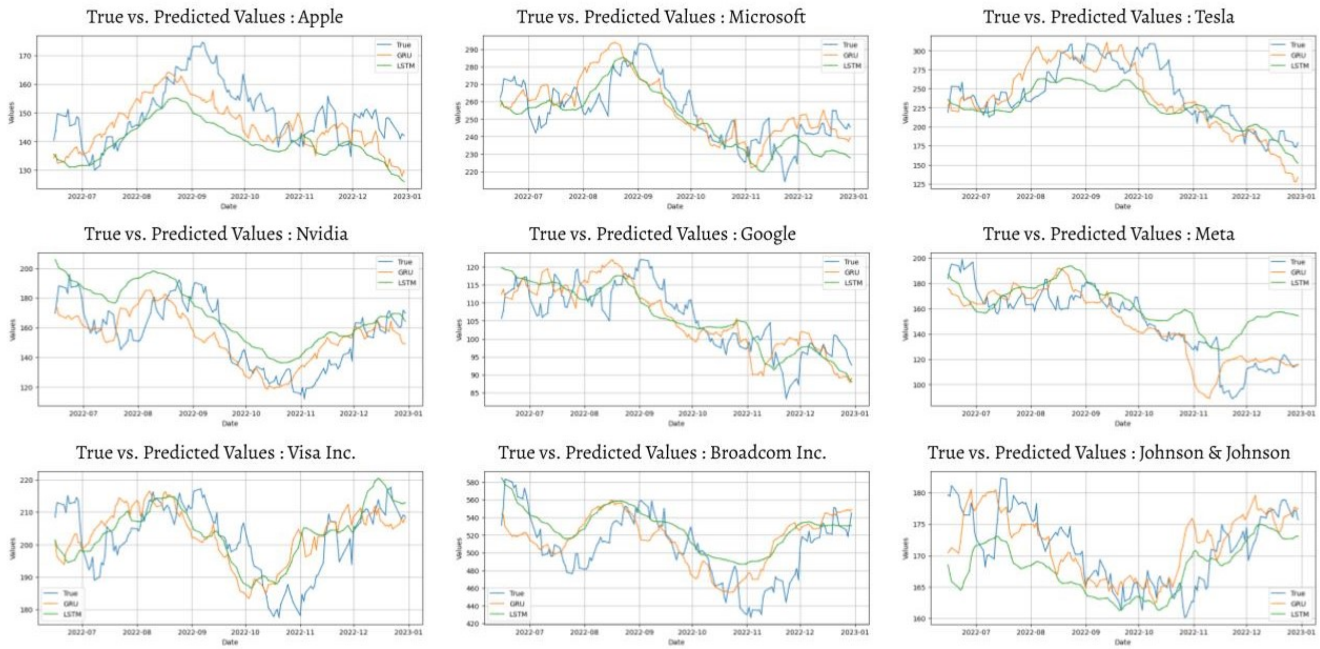

Figure 6: True vs Predicted Price: Top Companies

Table 5: Equations for Financial Market Indicators used

Indicator	Equation	Description
'volume'	$V = \sum_{i=1}^{n} \text{Volume}_i$	Trading activity sum.
'ma7'	$MA = \frac{1}{n} \sum_{i=1}^{n} \text{Price}_i$	7-day Moving Average.
'ma21'	$MA = \frac{1}{n} \sum_{i=1}^{n} \text{Price}_i$	21-day Moving Average.
'26ema'	$EMA_{26} = \frac{2}{27} \times (\text{Price} - EMA_{\text{prev}}) + EMA_{\text{prev}}$	26-day Exponential Moving Average.
'12ema'	$EMA_{12} = \frac{2}{13} \times (\text{Price} - EMA_{\text{prev}}) + EMA_{\text{prev}}$	12-day Exponential Moving Average.
'MACD'	$MACD = EMA_{12} - EMA_{26}$	Moving Average Convergence Divergence.
'20sd'	$\sigma_{20} = \sqrt{\frac{\sum_{i=1}^{20} (X_i - \bar{X})^2}{20}}$	20-day Standard Deviation.
'upper_band'	$UpperBand = MA + k \times \sigma$	Upper Bollinger Band.
'lower_band'	$LowerBand = MA - k \times \sigma$	Lower Bollinger Band.
'ema'	$EMA = \frac{2}{\text{span}+1} \times (\text{Price} - EMA_{\text{prev}}) + EMA_{\text{prev}}$	Exponential Moving Average.
'momentum'	$Momentum = \text{Price}_n - \text{Price}_{n-1}$	Rate of price change.
'rsi_30'	$RSI_{30} = 100 - \frac{100}{1+RS}$	Relative Strength Index.
'cci_30'	$CCI_{30} = \frac{\text{Typical Price} - SMA_{30}}{0.015 \times \text{Mean Deviation}_{30}}$	Commodity Channel Index.
'dx_30'	$DX_{30} = \frac{\text{DI}_{30+} - \text{DI}_{30-}}{\text{DI}_{30+} + \text{DI}_{30-}}$	Directional Movement Index.
'close_30'	$Close_{30} = \sum_{i=1}^{30} \text{Closing Price}_i$	Closing price (last 30).
'close_60'	$Close_{60} = \sum_{i=1}^{60} \text{Closing Price}_i$	Closing price (last 60).
'Log_Returns'	$\text{Log Returns} = \log\left(\frac{\text{Price}_n}{\text{Price}_{n-1}}\right)$	Logarithmic returns.
'Vola10d'	$Vola_{10d} = \sqrt{\frac{\sum_{i=1}^{10} (\text{Log Returns}_i - \text{Log }\bar{\text{R}}\text{eturns})^2}{10}}$	Volatility (10-day).
'Vola30d'	$Vola_{30d} = \sqrt{\frac{\sum_{i=1}^{30} (\text{Log Returns}_i - \text{Log }\bar{\text{R}}\text{eturns})^2}{30}}$	Volatility (30-day).
'Vola60d'	$Vola_{60d} = \sqrt{\frac{\sum_{i=1}^{60} (\text{Log Returns}_i - \text{Log }\bar{\text{R}}\text{eturns})^2}{60}}$	Volatility (60-day).
'Vol_Pct_Change'	$\text{Vol \% Change} = \frac{\text{Volume}_{\text{current}} - \text{Volume}_{\text{previous}}}{\text{Volume}_{\text{previous}}} \times 100$	Volume percentage change.
'Volu10d_ff'	$\text{Volu}_{10d\_ff} = \sum_{i=1}^{10} (\text{Volume Force}_i - \text{Volum}\bar{\text{e}}\text{ Force})$	Volume Force Index (10-day).
'Volu30d_ff'	$\text{Volu}_{30d\_ff} = \sum_{i=1}^{30} (\text{Volume Force}_i - \text{Volum}\bar{\text{e}}\text{ Force})$	Volume Force Index (30-day).
'Volu60d_ff'	$\text{Volu}_{60d\_ff} = \sum_{i=1}^{60} (\text{Volume Force}_i - \text{Volum}\bar{\text{e}}\text{ Force})$	Volume Force Index (60-day).
'atr'	$\text{ATR} = \frac{\sum_{i=1}^{n} \max(\text{High}_i - \text{Low}_i, \lvert\text{High}_i - \text{Close}_{i-1}\rvert, \lvert\text{Low}_i - \text{Close}_{i-1}\rvert)}{n}$	Average True Range.