Microcap Stock Predictions with GRU Neural Network

Abstract *- Accurate stock price projections are still essential for making well-informed investment decisions in the ever-changing financial world of today. This study uses a sophisticated GRU (Gated Recurrent Unit) model to improve stock price predictions for closing prices of the current, next, and previous day for five selected Microcap stocks. Even though we incorporate multiple features, such as news sentiment scores, our analysis takes a more insightful approach to understanding their impact, rather than establishing a direct connection between sentiment and price movements. By carefully preprocessing the data, training the model, and assessing its performance, we examine the GRU model's predictive ability and work to improve its ability to predict stock prices. Our research clarifies the complex interactions between the various factors and changes in stock prices, improving current techniques and promoting a better comprehension of market behaviour.*

Keywords - Stock price projections, GRU model, Closing prices, News sentiment scores, Data preprocessing, Model training, Performance assessment, Predictive ability, Market behavior.

1. Introduction

Understanding the complex relationship between different factors and stock movements is still essential for investors to succeed in the current volatile financial landscape. Using a diverse set of features including daily returns, volatility, RSI (Relative Strength Index), SMA (Simple Moving Average), Sentiment scores, and stochastic oscillator, the GRU (Gated Recurrent Unit) model is utilized to predict the closing prices for the five microcap (small cap) stocks of the following, current, and previous days. Our emphasis now is on using features, such as news sentiment, to improve the accuracy of stock price forecasts for the close of the price of today, the close of the price of tomorrow, and the close of price the day before, as compared to emphasizing direct sentiment-based predictions. Using the GRU (Gated Recurrent Unit) model, a powerful tool that combines financial data analysis and machine learning, we aim to investigate how adding features to data could improve stock market forecasts. Our task is to determine to underlying market trends and offer investors insightful information by carefully processing data, training models, and conducting extensive evaluations.

1.1. Literature Review

Jingyi Shen and M. Omar Shafiq [1] predicts stock price movements using an LSTM (Long Short-Term Memory) neural network. PCA (Principal Component Analysis) is used to reduce the complexity of the input data, which is then prepared for input into the LSTM layer. The LSTM structure has two layers: an input LSTM layer and an output layer. The output layer predicts whether the stock price trend will increase or decrease. LSTM models excel at handling sequential data, identifying hidden patterns over time. However, a major drawback is the high computational resources needed to train these deep learning models, especially those dealing with extensive time-series data. We use the Gated Recurrent Unit (GRU) model to predict stock prices. GRU is a simpler version of LSTM, which makes it faster to train and less computationally expensive. Despite its simplicity, GRU models have performed similarly to LSTM models in predicting sequences, making them a good option for forecasting stock prices.

Khalid Alkhatib, Hassan Najadat, Ismail Hmeidi, Mohammed K. Ali Shatnawi [2] they used a k-Nearest Neighbor (kNN) classifier to forecast stock market closing prices. The kNN algorithm is easy to understand and set up, which makes it a good choice for many classification tasks. In this case, stock prediction is framed as a classification task based on similarity. Historical stock data and test data are represented as vectors in a space with N dimensions based on stock features. They then calculate the similarity between the test data and points in the historical data using the Euclidean distance or another similarity metric. The KNN is easy to understand and use, it can adjust to different data types and relationships and can be updated efficiently with new data without retraining. However, it can be slow to process large datasets and may have difficulty predicting with high-dimensional or noisy data. Unlike the kNN method, our study uses the Gated Recurrent Unit (GRU) model, a different machine learning technique, to predict stock market closing prices. One benefit of GRU over kNN is that it can capture patterns in sequential data over time, making it ideal for predicting time series like stock prices. Furthermore, compared to kNN, GRU models are better equipped to handle high-dimensional data and less susceptible to noisy inputs.

2. Methodology

This research began by finding five microcap stocks on Yahoo Finance. Their historical financial data was then collected from the Yahoo Finance. Relevant news data was also gathered from Yahoo Finance. A dataset combining the financial and news data was then created. Next, a GRU (Gated Recurrent Unit) model was built using Python to predict the closing stock price for the current day, previous day, and following day. The GRU model's performance was then compared to baseline models like the k-Nearest Neighbor (KNN) regression and a Decision tree regression model. This comparison assessed the GRU model's ability to predict stock prices.

In this section, we thoroughly describe the methods used in this research.

Chi Chen, Lei Xue, Wanqi Xing [3] demonstrated that the GRU (Gated Recurrent Unit) is a more simplified alternative to the LSTM (Long Short-Term Memory) model, combining LSTM's three gates into two. This makes the GRU more efficient and requires less computational power, leading to faster training. The GRU's ability to grasp and store long-term patterns in time-series data makes it ideal for analysing sequential data. It also requires less memory, allowing it to handle large datasets more effectively. Therefore, in this study, we selected the basic GRU model as our main model for its efficiency, effectiveness in handling long-term dependencies, and reduced memory needs.

2.1 GRU (Gated Recurrent Unit) Model

The GRU is a type of recurrent neural network designed to capture dependencies in sequences of data, like stock prices over time. It addresses some issues found in traditional RNNs, such as the vanishing gradient problem, by using gating units. It has two gating units the update gate and the reset gate. These gates help mitigate the vanishing gradient issue and outperform conventional RNNs. The update gate helps to control how very much historical data is carried forward, while the reset gate sets historical data discarded. Together, these gates control the flow of information. With this mechanism, GRUs can manage memory efficiently, which makes them ideal for activities that involve the modelling of long-term dependencies.

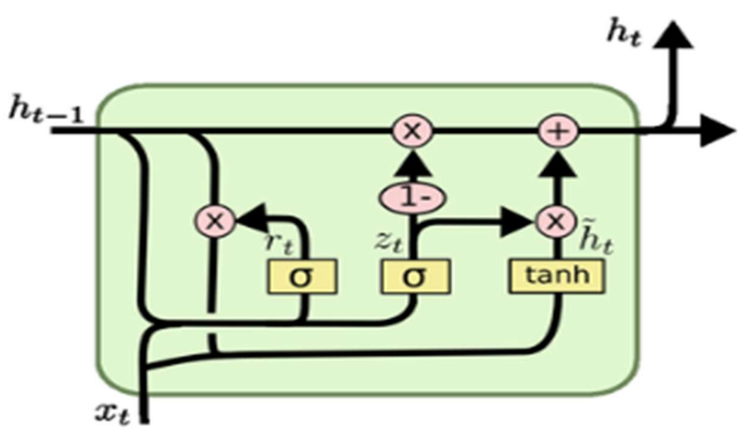

Fig 1 – GRU Model (Image taken from [5])

2.2 GRU Equations and Operations

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Equations taken from research paper [5]

As per the GRU model equations shown above the update gate (z_t) and the reset gate (r_t) are the two main gating units used by the Gated Recurrent Unit (GRU) framework to process sequential data. By using a sigmoid function to assess the importance of the historical data, the update gate oversees deciding how much of it should be kept for the future. It accomplishes this by examining both the current input (x_t) and the previous hidden state (h_{t-1}). Similar solutions are used by the reset gate to determine how much of the historical data should be erased. To maintain output within a reasonable range, the tanh function modifies the previous hidden state and combines it with the current input to create the candidate hidden state $(\widetilde{ht})$which is a possible new hidden state. After that, combining the previous hidden (h_{t-1}) state yields the actual new hidden state (h_t). The GRU can effectively handle and utilize historical data for sequential data processing because of this advanced gating mechanism, which solves issues that conventional RNNs frequently encounter. Below figure gives us clear idea of GRU cell.

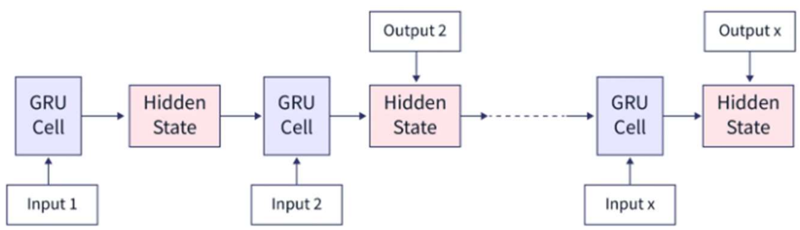

Fig 2 – Understanding the GRU cell (figure taken from research paper [9])

2.3 How Model Predicts Stock Prices

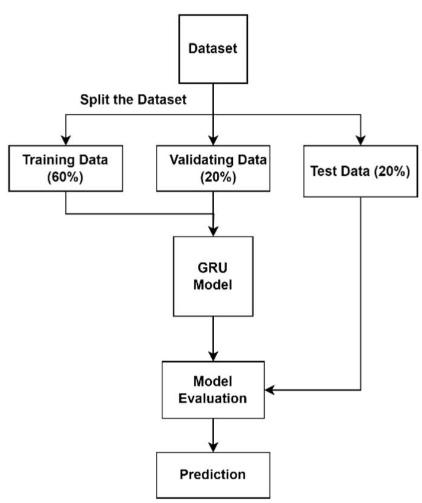

Fig 3 – Flowchart of methodology

As the block diagram shows after the dataset is cleaned and merged with features it is split into train (60%), validation (20%), and test set (20%). The training and validation data is used to train the GRU Model. In our stock prices prediction, key columns like 'Open', 'High', 'Low', 'Close', 'Volume', 'Daily Returns', 'Volatility', 'RSI', '%K', '%D', and 'Sentiment Score' are processed by the GRU layers using specialized gates and operations. These layers are skilled at keeping important details while eliminating irrelevant ones. The GRU layer's update gate is essential for maintaining information integrity across several time steps, which helps to comprehend long-term dependencies in stock price fluctuations. Simultaneously, the reset gate allows the model to determine whether a reset is required based on how relevant the previous hidden state was to the current prediction. The GRU dynamically updates its hidden state at each time step by combining data from current inputs, such as opening price, volume, and other technical indicators, with information from the stock prices of the previous day. This procedure is essential for forecasting the closing price of the following day. Likewise, the model uses a similar methodology to predict the close price of the previous day and the current price. The GRU model adjusts its hidden state to capture the finer details of the current market conditions by incorporating features and historical data, making accurate predictions for both the current and past close prices possible. The implementation of a unified methodology guarantees uniformity and efficacy in a range of prediction tasks related to the stock market.

2.4 Comparison with Null Model

The simple null baseline model is used to predict the target variable's mean or median for each instance in the test set. This provides a simple baseline against which the effectiveness of more complex models can be evaluated. In this research we using the regression model as the baseline model because a more thorough assessment can be obtained by contrasting the GRU model with regression models, considering the model's performance in comparison to other methods as well as any potential benefits it may have in terms of identifying temporal dependencies and non-linear patterns in the data. Compared to just comparing the model with a null baseline model, which might not adequately capture the challenges of stock price prediction, this method provides deeper insights into the model's capabilities.

Decision Tree Regression

Decision tree regression is a supervised learning algorithm that creates a regression model in the shape of a tree structure. By dividing the feature space into regions recursively and estimating the target variable using the mean of the target values in each region, the tree is built. For this, the scikit-learn library's DecisionTreeRegressor is used in the python code that is implemented. The training and validating data is used to train the model, and the test data is used to generate predictions. Calculations are made to determine the model's performance using evaluation metrics like Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R2) score.

KNN

The K-Nearest Neighbors (KNN) Regression algorithm is a non-parametric technique that generates predictions by averaging the target values of the K closest data points in the feature space. The Python code implements KNN Regression using a scikit-learn KNeighborsRegressor. The model is trained on the training and validating data and assessed using the test data and MSE, MAE, and R2 score, much like Decision Tree Regression.

2.5 Data Collection

We obtained two main datasets: news headlines relating to microcap stocks and historical data on microcap stocks. We gathered information on five different microcap stocks (SAVE, CLNE, LAZR, AMWL, and GEO) from January 1, 2019, to January 1, 2024, using the Yahoo Finance API. Over the same time period, news headlines for these stocks were thoroughly gathered from the Yahoo Finance RSS feed. To prepare for additional analysis, the gathered data was cleaned, arranged, and saved in CSV files.

2.6 Data Cleaning and Standardization

We cleaned and standardized both datasets to ensure data integrity and consistency. Python code has been used to remove duplicate news entries, and both dataset's dates were formatted uniformly. This procedure made it much easier to ensure the data were consistent and reliable, which was used to analyse later.

2.7 Feature Engineering

To find valuable information from the raw data, feature engineering techniques were used. For historical stock data, the following metrics were computed: daily returns, volatility, %K Stochastic Oscillator, Simple Moving Average (SMA), and Relative Strength Index (RSI). VADER, a sentiment analysis tool utilized for text analysis, was used to calculate sentiment scores for news headlines as per [4]. These characteristics enhanced our models' accuracy.

2.8 Data Merging

To build a single dataset, we merged the historical stock data and news headline datasets based on date and ticker columns. We were able to effectively analyse the relationship between changes in the stock market and news events due to this combined dataset. To separate the effect of news on stock prices, dates lacking relevant news articles were given a default "Missing Sentiment" score of zero. To facilitate additional research and analysis, the entire dataset was saved as a CSV file and then it was used for the prediction of the stock prices.

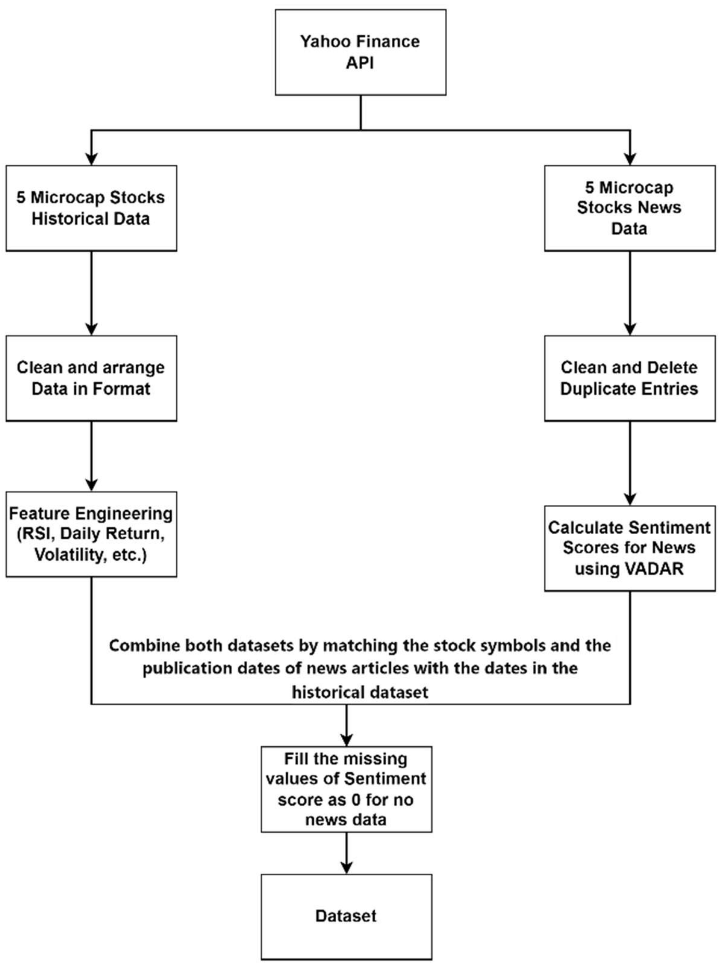

Fig – 4 Flowchart for Dataset creation

3. Results

The GRU model's ability to predict closing prices for the following day, the current day, and the day before demonstrates how well it captures the dynamics of stock prices over time. The predictions offer insightful information about short-term stock price movements through careful data analysis and model training, assisting investors in making wise financial decisions.

GRU Model's Next Day Close Price Prediction Output

Stocks Ticker	MSE	MAE	R-square
SAVE	0.853	0.655	0.984
CLNE	0.110	0.189	0.989
LAZR	0.443	0.450	0.992
AMWL	0.446	0.454	0.995
GEO	0.083	0.186	0.985

Table 1- Next day close price prediction output

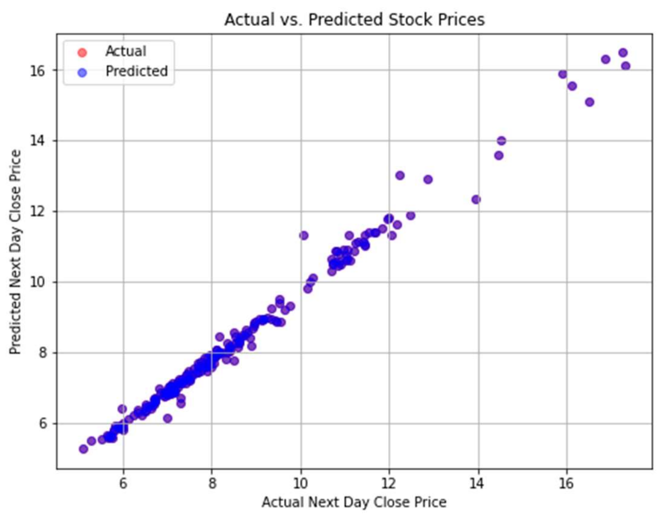

Fig 5 – Scatter plot of actual vs predicted next day close price

The GRU model's predictive accuracy for the closing price of five microcap stocks the next day is demonstrated by the results, where R-squared values are consistently above 0.98, indicating that a significant portion of the variance in stock prices is explained by the model. The results for AMWL are especially remarkable; they show an R-squared of 0.995, indicating nearly perfect predictive ability within the parameters of the model. The model's performance is represented by a scatter plot, where the closeness of the "Predicted" and "Actual" values, particularly near the line of unity, further supports the model's efficacy in predicting future stock prices.

GRU Model's Current Close Price Prediction Output

Stocks Ticker	MSE	MAE	R-square
SAVE	0.271	0.332	0.995
CLNE	0.067	0.143	0.993
LAZR	0.231	0.277	0.996
AMWL	0.212	0.274	0.997
GEO	0.015	0.081	0.997

Table 2 - Current day close price prediction output

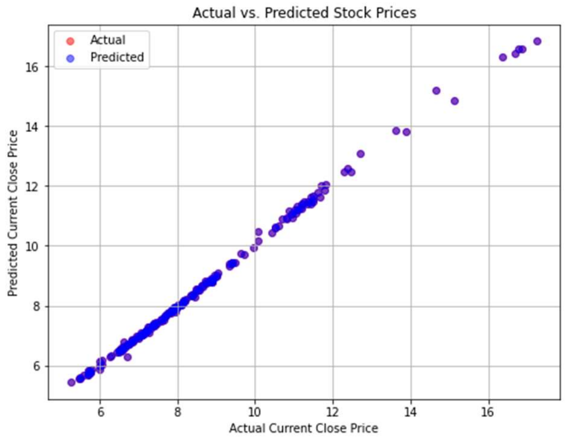

Fig 6 – Scatter plot of actual vs predicted current day close price

The table and scatter plot that above shows how well the GRU model forecasts current close prices for a variety of stocks. The model is noteworthy for achieving remarkably high R-squared values, it reaches 0.997 for AMWL and GEO, indicating that almost all variance in the actual prices is captured by the model's predictions. GEO has the lowest Mean Squared Error (MSE) and Mean Absolute Error (MAE), highlighting the accuracy of the model. The model's accuracy is demonstrated by the scatter plot, where values that are predicted and actual prices closely match, especially for values that are clustered around the line of perfect prediction.

GRU Model's Previous Day Close Price Prediction Output

Stocks Ticker	MSE	MAE	R-square
SAVE	0.133	0.246	0.997
CLNE	0.039	0.105	0.996
LAZR	0.134	0.267	0.997
AMWL	0.151	0.201	0.998
GEO	0.019	0.10	0.996

Table 3 - Previous day close price prediction output

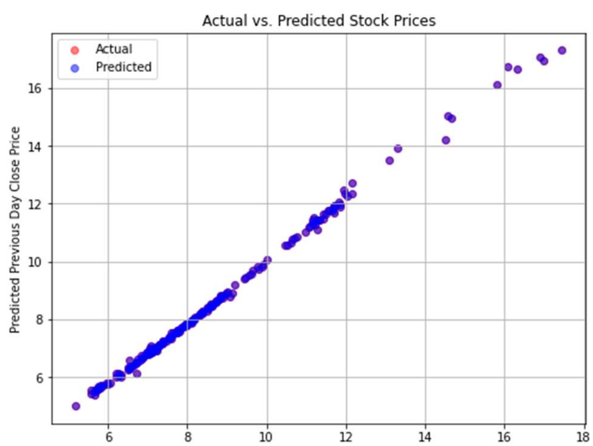

Fig 7 – Scatter plot of actual vs predicted previous day close price

R-squared values for all stocks are close to unity, indicating that the GRU model has shown remarkable accuracy in predicting the closing prices of the previous day. The very low MSE and MAE across various tickers, which indicate a high degree of fit, imply that the model has successfully captured the temporal dynamics of the stock prices. The model's accuracy is validated by the scatter plot of the actual versus predicted prices, which shows that the model is reliable in hindsight price predictions because the predicted values closely match the actual figures, forming a dense and nearly linear cluster.

Comparison Table of GRU vs Decision Tree and KNN

Model	MSE (NC)	MAE (NC)	R-Square (NC)	MSE (CC)	MAE (CC)	R-Square (CC)	MSE (PC)	MAE (PC)	R-Square (PC)
SAVE									
GRU	0.853	0.655	0.984	0.271	0.332	0.995	0.133	0.246	0.997
Decision Tree	0.775	0.589	0.985	0.276	0.319	0.995	0.578	0.478	0.989
KNN	1.241	0.793	0.977	1.245	0.818	0.997	1.097	0.778	0.979
CLNE									
GRU	0.110	0.189	0.989	0.067	0.143	0.993	0.151	0.201	0.998
Decision Tree	0.187	0.257	0.982	0.044	0.119	0.995	0.154	0.171	0.984
KNN	0.346	0.404	0.967	0.403	0.421	0.963	0.294	0.398	0.970
LAZR									
GRU	0.443	0.450	0.992	0.231	0.277	0.996	0.134	0.267	0.997
Decision Tree	0.947	0.545	0.984	0.382	0.299	0.993	0.310	0.296	0.994
KNN	2.008	0.996	0.966	2.261	1.069	0.9628	1.919	0.984	0.968
AMWL									
GRU	0.446	0.454	0.995	0.212	0.274	0.997	0.151	0.201	0.998
Decision Tree	0.621	0.389	0.993	0.108	0.181	0.998	0.233	0.225	0.997
KNN	1.697	0.924	0.981	1.865	0.977	0.978	1.816	0.931	0.980
GEO									
GRU	0.083	0.186	0.985	0.015	0.081	0.997	0.019	0.10	0.996
Decision Tree	0.194	0.255	0.965	0.038	0.114	0.993	0.052	0.131	0.991
KNN	0.161	0.289	0.971	0.149	0.283	0.973	0.153	0.270	0.973

Table 4 – Comparison of GRU model with Base line model (Decision rree regrssion and KNN)

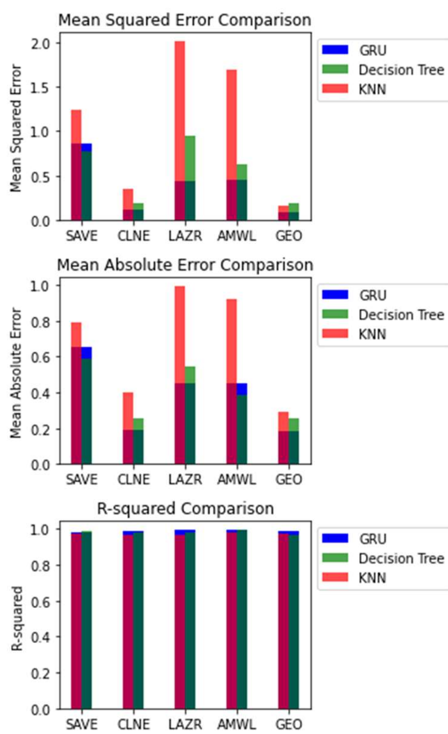

Fig 8 – Comparison plot of GRU vs Base line model (Decision tree regression and KNN)

The detailed comparison demonstrates how well the GRU model predicts stock prices compared to Decision Tree and KNN algorithms. The GRU model shows exceptional predictive accuracy, especially for the stocks LAZR and AMWL, with R-squared values of 0.997 and 0.998, respectively, indicating a nearly perfect explanation of the price variance. The GRU model's consistency and forecasting reliability are further highlighted by bar charts that show the Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared metrics across various models and stocks. Notably, the GRU model has higher R-squared values and lower error rates than its counterparts.

K Fold Cross Validation Result

Ticker	MSE	MAE	R-Square
Next day close price			
SAVE	0.701	0.564	0.986
CLNE	0.107	0.195	0.987
LAZR	0.586	0.459	0.989
AMWL	0.643	0.483	0.992
GEO	0.106	0.209	0.982
Current day close price			
SAVE	0.319	0.367	0.993
CLNE	0.043	0.163	0.995
LAZR	0.287	0.324	0.995
AMWL	0.483	0.442	0.994
GEO	0.03	0.144	0.993
Previous day close price			
SAVE	0.354	0.421	0.993
CLNE	0.053	0.152	0.993
LAZR	0.220	0.282	0.996
AMWL	0.317	0.327	0.996
GEO	0.023	0.099	0.996

Table 5 – K fold cross validation results

The K-Fold Cross Validation results for close price predictions for the next day, current day, and previous day demonstrate a consistently high R-Square for all stocks, indicating the model's strong predictive capacity. MSE and MAE, however, vary widely for some stocks, such as CLNE, have significantly lower error rates, indicating better model performance on those stocks. This variation in errors may be a sign of how sensitive the model is to the distinct volatility and trading patterns associated with various stocks. The model's validity in stock price forecasting is generally confirmed by the results, though the variation in error metrics might indicate the need for feature engineering or stock-specific model tuning.

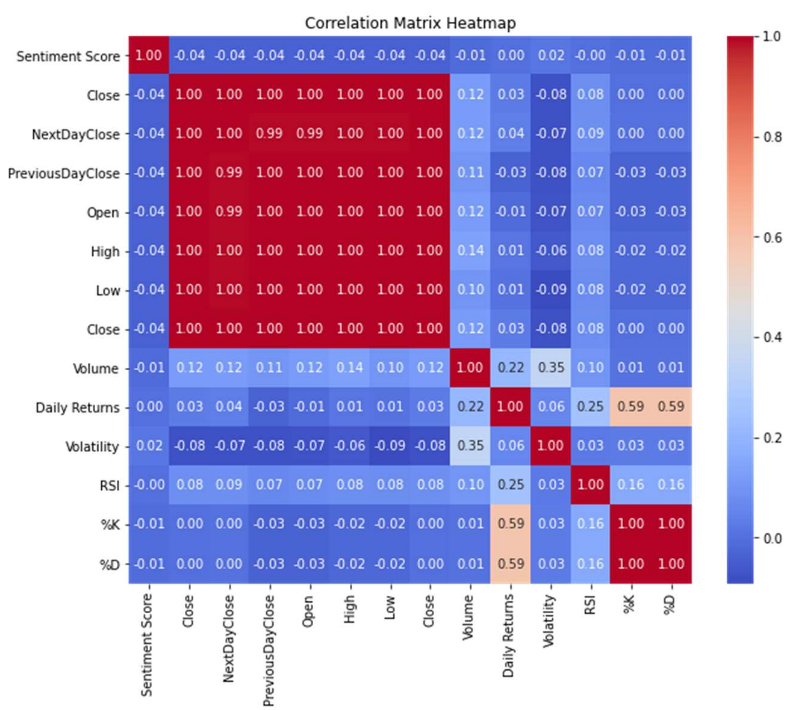

Fig 9 – Correlation Matrix

As we can see from the nearly zero correlation coefficients with "Open," "High," "Low," and "Close" prices, sentiment scores have a negligible linear relationship with daily stock prices when comparing the correlation matrix for various stock-related metrics against sentiment scores. This suggests that other external factors may have an impact on sentiment as sentiment scores do not significantly correspond with daily price movements in the market. However, as predicted, there is a significant intercorrelation between the price metrics, which illustrates how the opening, closing, high, and low prices within trading periods are interdependent. Although it is not a strong predictor, the trading volume shows a slight positive correlation with stock prices, suggesting some relationship between volume and price changes. It is interesting to note that there is a strong positive correlation between "Daily Returns" and "Volatility," indicating that larger price movements are typically accompanied by higher volatility.

Even though they have a strong correlation with "Daily Returns" and a high intercorrelation, technical indicators like "%K" and "%D" (stochastic oscillator) show little correlation with sentiment scores. This may suggest that these technical indicators function somewhat apart from the overall sentiment of the market as recorded in the dataset. When creating predictive models and conducting financial analysis, it is important to distinguish between variables that offer unique information and those that might be deemed redundant, and this is something that the correlation matrix helps with.

The lack of news may be a major contributing factor to the poor correlation observed between sentiment scores and microcap stock prices. Microcap stocks usually get less media attention than large-cap stocks, where news coverage is plentiful and can have a significant impact on prices. Sentiment scores, which frequently consider the volume and content of news, might not fully capture the range of market sentiment affecting these stocks because of the decreased news flow. As a result, news's impact on microcap stocks is probably more muted, as seen by the correlation matrix, and this could mean using distinct analytical techniques for these financial market assets.

4. Discussion

The GRU model has limitations even though it shows impressive predictive accuracy for stock prices, with high R-squared values demonstrating its efficacy in capturing time-series patterns. Despite its strength, its intricate neural network architecture can be computationally demanding, requiring considerable resources for both training and inference. Moreover, the "black box" nature of the GRU may present interpretability issues, making it challenging for stakeholders to understand and have faith in the model's decision-making process. To guarantee that performance on training data transfers well to unknown data, rigorous tuning and validation are also necessary due to the possibility of overfitting with GRU models. These elements emphasize how crucial it is to strike a balance between operational viability and performance, as well as the capacity to decipher and verify model outputs, especially in the closely regulated and thoroughly investigated field of finance.

Comparison Table of GRU vs Linear Regression and Random Forest Regression

Model	MSE (NC)	MAE (NC)	R-Square (NC)	MSE (CC)	MAE (CC)	R-Square (CC)	MSE (PC)	MAE (PC)	R-Square (PC)
SAVE									
GRU	0.853	0.655	0.984	0.271	0.332	0.995	0.133	0.246	0.997
Linear Regression	0.456	0.473	0.991	0.047	0.140	0.999	0.093	0.182	0.998
Random Forest	0.503	0.433	0.990	0.159	0.222	0.997	0.394	0.373	0.992
CLNE									
GRU	0.110	0.189	0.989	0.067	0.143	0.993	0.151	0.201	0.998
Linear Regression	0.056	0.165	0.994	0.157	0.074	0.998	0.061	0.089	0.993
Random Forest	0.104	0.183	0.990	0.385	0.099	0.996	0.144	0.136	0.985
LAZR									
GRU	0.443	0.450	0.992	0.231	0.277	0.996	0.134	0.267	0.997

Linear Regression	0.331	0.415	0.994	0.092	0.176	0.998	0.153	0.210	0.997
Random Forest	0.491	0.397	0.991	0.199	0.205	0.996	0.1643	0.227	0.997
AMWL									
GRU	0.446	0.454	0.995	0.212	0.274	0.997	0.151	0.201	0.998
Linear Regression	0.301	0.329	0.996	0.054	0.120	0.999	0.062	0.167	0.999
Random Forest	0.400	0.315	0.995	0.056	0.128	0.999	0.081	0.145	0.999
GEO									
GRU	0.083	0.186	0.985	0.015	0.081	0.997	0.019	0.10	0.996
Linear Regression	0.076	0.175	0.986	0.006	0.050	0.998	0.008	0.054	0.998
Random Forest	0.085	0.163	0.984	0.020	0.079	0.996	0.025	0.093	0.995

Table 6 – Comparison between GRU vs Linear Regression and Random Forest Regression

The GRU models' performance is evaluated using MSE, MAE, and R-squared as evaluation metrics, in comparison to Linear Regression and Random Forest Regression for a range of stocks. Because they are so good at capturing temporal dependencies, GRU models are well-known for their strength in time-series analysis. This is evident in their consistently strong performance, which is indicated by high R-squared values, which indicate significant variance explanation across all stocks. As an explanation of its drawbacks in capturing the complex, non-linear relationships found in financial time-series data, Linear Regression, while providing a straightforward modelling approach, yields higher MSE and MAE values when compared to GRU models. But because it is easy to understand and straightforward, it is a helpful tool. GRUs, being more complex, require more computational resources and are less interpretable than Linear Regression

5. Future Scope

Future for this work might centre on expanding its scope to include a wider range of market indicators, such as macroeconomic variables, sector indices, and worldwide financial trends, to improve the GRU model's predictive accuracy. Investigating hybrid models that combine the interpretability of Random Forest or XGBoost with the temporal strengths of GRU would also be beneficial. More complex market insights might also be obtained by utilizing alternative data sources like real-time trading data, news sentiment extraction, and social media sentiment analysis. Using the model in real-time trading systems with real-time market data and evaluating its performance in an operational environment is another way to investigate the concept. Lastly, to facilitate its use, efforts could be made to enhance the model's interpretability and computational efficiency.

6. Conclusion

The study has shown that GRU neural networks can predict stock prices with a high degree of accuracy. This is supported by strong R-squared values for a variety of microcap industries. GRU's advantages in capturing intricate, nonlinear relationships in time-series data have been revealed by comparing its performance to that of more conventional models such as Random Forest and Linear Regression. The GRU model is a prominent instrument in financial modelling, even though it has interpretability and computational demands that present challenges. These findings set the stage for future research to improve these predictive models even more, making them more useful for financial analysis and decision-making in the real world and opening the door to more advanced algorithmic trading techniques.

7. References

[1] Shen, J. and Shafiq, M.O. (2020) “Short-term stock market price trend prediction using a comprehensive deep learning system,” *Journal of Big Data*, 7(1), p. 66. Available at: https://doi.org/10.1186/s40537-020-00333-6.

[2] Alkhatib, K. *et al.* (2013) *Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm*, *International Journal of Business*. Available at: https://www.ijbhtnet.com/journals/Vol_3_No_3_March_2013/4.pdf.

[3] Chen, C., Xue, L. and Xing, W. (2023) “Research on Improved GRU-Based Stock Price Prediction Method,” *Applied Sciences*, 13(15), p. 8813. Available at: https://doi.org/10.3390/app13158813.

[4] Kabbani, T. and Usta, F.E. (2022) “Predicting The Stock Trend Using News Sentiment Analysis and Technical Indicators in Spark.” Available at: https://www.researchgate.net/publication/358233294_Predicting_The_Stock_Trend_Using_News_Sentiment_Analysis_and_Technical_Indicators_in_Spark

[5] Agarap, A.F. (2017) “A Neural Network Architecture Combining Gated Recurrent Unit (GRU) and Support Vector Machine (SVM) for Intrusion Detection in Network Traffic Data.” Available at: https://doi.org/10.1145/3195106.3195117.

[6] By Cathrine Jeeva (2023) *Gated Recurrent Unit (GRU)*, *Scalar Topics*. Available at https://www.scaler.com/topics/deep-learning/gru-network