Module Code: CS3AM

Assignment Report Title: Machine Learning Model Comparison Coursework

Date (when the work completed): 11th December 2024

Actual Hrs spent for the assignment: 45

DO NOT REMOVE

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

COVER PAGE

Comparison of Machine Learning Models for Stock Prediction in U.S. Stock Market

Full Name: Shreya Sharma

Student Number: 31022399

Module Convenor: Dr Shahzad

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

"*The hypothesis posits that CNN-LSTM will outperform Random Forest due to its ability to accurately forecast sequence pattern information, with error (actual – predicted) of convergence measured through R^2 and MSE."*

STAGE 1.0: BACKGROUND RESEARCH AND DEFINING THE PROBLEM

Modern politics has a direct impact on the activity of the U.S. stock market, a key cornerstone in the American economy. Where the accurate prediction of future stock price would play a pivotal role in minimizing investment risk, and hence maximising returns. Volatility is driven by interest rates, inflation, and geopolitical events, so therefore requires models to be capable of forecasting complex financial patterns to make accurate predictions. Using historical datasets to make predictions is challenging because it necessitates modelling both intra-series temporal models and inter-series correlations jointly *(Research Article Volume 6 - Issue 4: 422-426 / July 2023 Black Sea Journal of Agriculture BSJ Agri / Cevher ÖZDEN 423),* only mitigated by a simplified model structure and the use of deep learning architectures which overcome this limitation in their solution.

According to the Financial Analyst's Journal *(Schwert, William. "Stock Market Volatility." Financial Analyst's Journal, 46, no.3 (1990)):* stock volatility has been a particularly challenging characteristic to the U.S. banking industry since the 1987 stock market crash, out of which numerous techniques have been tried and tested in vain to mitigate the difficult obstructions it poses to the smooth sailing of large-scale investment, such as trading halts, margin requirement increases and limits on automated trading systems *(lines 5-8).* This highlights the need for financial innovation, including Miller (1986) and Merton (1992), and the importance of new products and services in the financial arena. *(Lerner, Josh. "The new new financial thing: The origins of financial innovations" Journal of Financial Economics, 79, 2 (2006)).* The Federal Reserve's monetary policy decisions influence borrowing costs, corporate profitability and consequently stock price *(Board of Governors of U.S. FRS, "Historical Interest Rate Data", FRED, 2024),* when coupled with erosion from high inflation rates such as the 7.7% annual increase from January 2021-June 2022 – the highest in 40 years. This reveals the subversive influence of geopolitical tensions and trade relations such as hikes in energy prices (oil/gas) and the supply chain issues like the COVID-19 pandemic, and supports the need for technological intervention in financial planning, which can provide statistically based prediction quantified to machine input, leading to more accurate results.

These factors impacted the S&P 500 in the form of significant increase in volatility, such as the 2022 9.15 surge. In response, and to combat inflation, the Federal Reserve raised interest rates from 0.25% to 4.5% within 2022, further pressuring growth stocks and sectors sensitive to borrowing costs. This uncertainty underscores the market need for robust stock prediction models like CNN-LSTM, which studies such as *MDPI Journal of Risk and Financial Management* show improve prediction accuracy compared to traditional models due to their ability to combine pattern and image recognition. By leveraging these models, investors can improve navigation of inflation-driven market dynamics, mitigate risks and make informed decisions about portfolio management.

1.1 PROBLEM STATEMENT

Volatility causes problems for streamlined future stock prediction due to the excessive short-term uncertainty. Outlier distortion and limited historical range/only US stock data also impact the degree of accuracy possible. The objectives of this study are to compare CNN-LSTM and Random Forest's efficiency in calculating next day stock prediction values, in order to determine which is more suitable for prediction of highly volatile and changeable data.

1.2 DATASET SELECTION: *Kaggle's S&P 500 Stock data*

Justification of Choice: As a key indicator of overall market health and trends, the dataset is not only relevant to understanding U.S. financial markets but also influential in global economic contexts, making insights derived from this data widely applicable and impactful. It spans a timeframe of 5 years, covering metrics like daily opening/closing price, daily highest/lowest stock price and volume with stock values recorded in daily, 24 hour intervals by date for time-series forecasting. The encompassing metrics provide robust machine input for analysing both price trend and volatility for future daily prediction. It was chosen over *Yahoo Finance* and *Quandl* dataset options for its larger size, better time-bound granularity (open/close prices), and superior data quality. Additionally, *Quandl + Yahoo Finance* datasets present issues with user accessibility due to corporate data protection ethics, making Kaggle datasets preferable due to their open-source availability and reliability. Limitations include that the dataset is from U.S. stock market only, therefore internal economic factors are biased, and results can't be globally representative. Plus, the historical range is limited, & therefore isn't suitable for long-term analysis e.g. over multiple economic cycles as long term historical trend insights aren't available. Finally, outliers caused by market anomalies (e.g., 2008 crash) could distort predictions without careful handling and data preprocessing.

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

1.4 THE SOLUTION: WHY SUPERVISED DAILY TIME-SERIES FORECASTING FOR STOCK PREDICTION?

Based on background research, the best approach is to develop two machine learning models using Supervised Daily Time-Series forecasting, rather than regression or cross-sectional analysis. Regression would only provide long-term predictions, which isn't suitable given the limited historical data and high short-term volatility. Daily Time-Series forecasting, which predicts future values based on past data, is ideal for forecasting daily S&P500 stock prices using the past 5 years of data. This method accounts for volatility, inconsistencies, and market events (like American bill amendments or housing crises), aligning with the goal to showcase machine learning for price prediction. The 1D CNN-LSTM hybrid will compute TensorFlow complexity, while the basic Random Forest model will highlight differences in architectural efficiency.

Model Justification

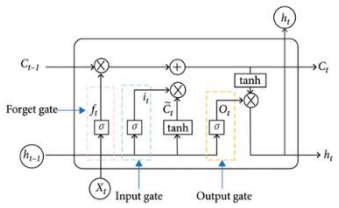


Figure 1: LSTM Structure

In daily time-series stock forecasting, model choice balances performance, interpretability, and complexity. Random Forest is robust to noise, handles NaN values, and reduces overfitting by redistributing weights. However, CNN-LSTM, as shown in the 2020 Complexity Journal study "A CNN-LSTM-Based Model to Forecast Stock Prices" (Wang, Jingyang (Editor), Complexity Journal, 2020), combines CNN's feature extraction with LSTM's pattern recognition, offering high accuracy for detecting seasonality, despite interpretability challenges (addressable via Shapley values) (Hochreiter & Schmidhuber, 1997). LSTM's forget gate prevents overfitting by discarding irrelevant information. Random Forest, while effective for high-dimensional data, struggles with capturing complex sequential patterns like CNN-LSTM models.

Random Forest	1D CNN-LSTM
Handles Non-Linear Relationships and Heterogeneity	Captures Spatial and Temporal Dependencies
Reduced Overfitting via Bootstrap Aggregation	Automatic Feature Learning
Interpretability and Variable Importance	High Accuracy in Complex, Noisy Data
Robust to Overfitting in High-Dimensional Data	Effective for Multi-Step Forecasting

Table 1: Comparison Table

STAGE 2.0: EXPLORATORY DATA ANALYSIS

Objective of EDA: Inform Feature Selection and Model Implementation through extracting key insights about the data.

2.1 DATASET DESCRIPTION

This involves providing a summary of the dataset, including:

- *The dataset contains 619041 rows and 7 columns. There are 5 numerical columns (open, high, low, close, volume) and 2 categorical columns (date, symbol). The target variable is binary, indicating employment status:* ***1 = employed*** *and* ***0 = unemployed.***

Missing values are found in columns like Open, High, Low, Close, and Volume, and can be addressed using imputation techniques like forward fill. The dataset is ideal for tasks such as predicting the Close price or classifying daily price direction based on whether the stock closed higher than it opened.

Feature Name	Data Type	Description	Example Values
Date	Categorical	Trading date for stock data	2023-01-01
Symbol	Categorical	Stock ticker symbol identifying company	AAPL
Open	Numerical	Opening price of stock for day	135.6
High	Numerical	Highest price of stock during the day	140.5
Low	Numerical	Lowest price of stock during the day	130.8
Close	Numerical	Closing price of stock during the day	138.9
Volume	Numerical	Total number of shares traded during day	1,200,000

Table 2: Feature Description Table

Key features include daily stock prices and trading volume from 2013-18, providing insights into market trends, volatility, and company performance. This dataset is commonly used for financial analysis, forecasting, and algorithmic trading research.

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

2.2 DATA VISUALISATION THROUGH TIME-SERIES DECOMPOSITION

Time-Series Decomposition

Additive time-series decomposition was applied to extract key components: trend, seasonality, and anomalies. The trend reflects long-term movement in S&P500 stocks, improving model accuracy. Seasonality captures regular patterns, like weekly fluctuations, for feature engineering. Anomalies identify values that deviate from the trend, mean, or variance based on normal distribution, which helps contextualize and aggregate insights from the decomposition.

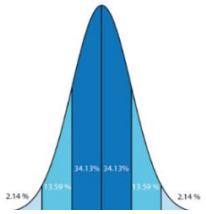


Figure 2: Normal Distribution for Decomposition Context

2.2a CNN-LSTM PLOT ANALYSIS

'Stock Trading Volume Over Time' **Line Graph**

Purpose

The graph tracks ***daily trading volume*** *from 2013 to 2018 to identify market activity trends, anomalies, and patterns critical for feature engineering and further model development, reveal key variable relationships between volume and time.*

Time Series Decomposition Element	*Graphical Proof of Relationships between Variables*
General Trends	• Volume shows significant fluctuations with a consistent baseline of activity. • Peaks in 2014 and 2016 highlight periods of heightened trading, likely tied to major market events. • Post-2016, reduced variance suggests a more stable market phase.
Anomalies/Noise	• Extreme spikes in *2014* and *2016* reflect significant market events, essential for maintaining data fidelity during preprocessing.
Seasonality	• No Clear Seasonality. While no evident seasonality emerge, rolling mean and decomposition analysis could confirm hidden trends.

Pair-Plot for selected features (e.g., Open, High, Low, Close, Volume, [*please see code for full]***)**

Purpose

This graph builds on the line plot's analysis of temporal dependencies, focusing on the spatial relationships to inform the construction of the CNN layer.

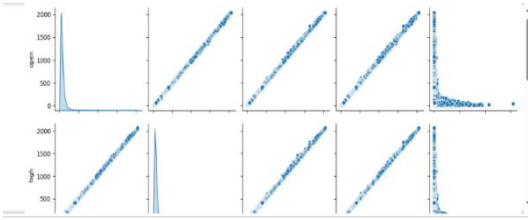


Figure 3: Feature Pair-Plots

Time Series Decomposition Element	*Graphical Proof of Relationships between Variables*
General Trends	• The scatterplots reveal **strong linear relationships** between features like open, high, and other price-based variables. This indicates that the stock price variables are highly correlated, making them reliable predictors for one another in a time-series forecasting model. • - The density distributions at the diagonal highlight skewed data (e.g., in open), suggesting that some features might require transformation (e.g., log-scaling) to stabilize variance and reduce the impact of extreme values.
Anomalies/Noise	• Outliers are apparent in certain pairings, particularly in the distribution of opening prices and their relationship to other variables. These spikes represent **anomalous market behaviour** (e.g., market crashes or booms). Detecting and addressing these outliers during preprocessing is essential to avoid skewing model predictions.
Seasonality and Correlation	None

2.2b RANDOM FOREST PLOT ANALYSIS

Boxplot of Daily Returns (to detect outliers)

Purpose

This graph's purpose is to detect anomalies, ensuring input data is clean and reliable.

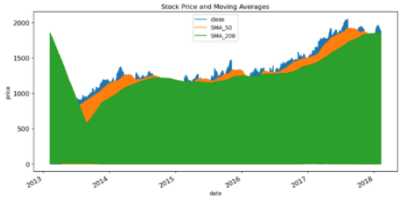
Time Series Decomposition Element	*Graphical Proof of Relationships between Variables*
General Trends	• The boxplot captures the distribution of daily returns, with most values concentrated around a small range near zero, reflecting the limited daily variability of stock prices.

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

	• The whiskers indicate the range of typical daily returns, while the clustering suggests that most daily returns are minor and stable under normal market conditions.
Noise/Anomalies	• The presence of extreme outliers, such as daily returns exceeding 10% or 15%, signals significant market events or irregular trading patterns. These anomalies could reflect events like earnings releases, economic announcements, or geopolitical news that cause sharp price movements. • Identifying these outliers is critical for preprocessing, as they may require special handling to avoid skewing model performance. 3 extreme outliers are identified in total.
Seasonality	• While this visualization does not directly reveal seasonality, the boxplot helps determine if periods of high volatility align with specific events. For a deeper analysis, pairing this with time-based visualizations can help uncover patterns in the frequency of outliers.

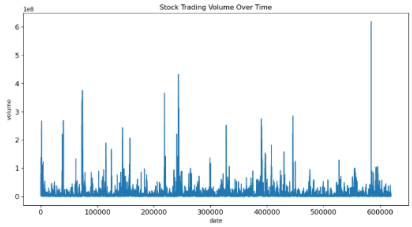
Moving Averages (Simple Moving Average/SMA_50/200)



Time Series Decomposition Element	*Graphical Proof of Relationships between Variables*
General Trends	• The **close price (blue line)** shows a consistent upward trend from 2013 to 2018, reflecting long-term market growth. • The **SMA (Simple Moving Averages)** smooth out short-term fluctuations, with the SMA-200 (green area) capturing the broader trend, while SMA-50 (orange area) reflects more responsive, shorter-term trends. • The visualization confirms that stock prices are more volatile over shorter periods, while longer-term patterns remain stable. The **close price (blue line)** shows a consistent upward trend from 2013 to 2018, reflecting long-term market growth. • The **SMAs** smooth out short-term fluctuations, with the SMA-200 (green area) capturing the broader trend, while SMA-50 (orange area) reflects more responsive, shorter-term trends. • The visualization confirms that stock prices are more volatile over shorter periods, while longer-term patterns remain stable.
Anomalies/Noise	• Periods where the **close price dips below the SMA-200** (e.g., in 2014) might indicate bearish trends or corrections. These anomalies could represent market downturns or external shocks, which are critical for understanding market behaviour during unusual events.
Seasonality and Correlation	• No explicit seasonality is observable in this plot, but the moving averages effectively capture the cyclic nature of stock price fluctuations. Further seasonal decomposition could confirm periodicity if subtle trends exist.

Target Variable ('close' (TV)) Over Time

.



Time-Series Decomposition Element	*Graphical Proof of Relationships between Variables*
General Trends	• *General increase in the closing price over time (2013-2018), indicating long-term growth, possibly due to economic recovery or market performance.*
Noise/Anomalies	• *The Daily Return Distribution graph shows a highly concentrated frequency at 0, suggesting potential data errors, extreme outliers, or flat trading days that require further investigation.*
Seasonality	• *The graph lacks clear seasonal patterns in the closing price, indicating that stock price fluctuations might not exhibit consistent periodicity within the observed time frame.*

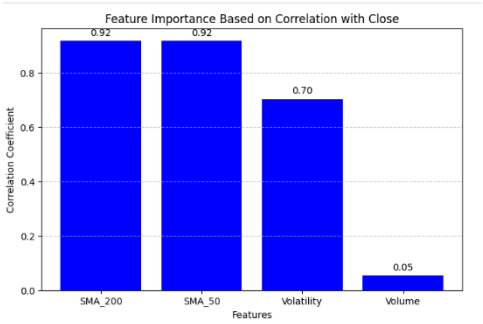
Correlation Coefficient Feature Importance Plot

Ranking and Insights:

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

- Ranking features revealed SMA_200 & SMA_50 had the most correlation with the target variable of closing price, a considerable 0.22 higher than the second feature Volatility, which suggests it captures additional market dynamics not reflected in direct matrix correlations such as daily returns due to the inclusion of SMA and volatility
- Output from Random Forest indicated that **SMA_200** contributed most significantly to predicting **close price** since it had an importance score of **0.92**, emphasizing its predictive relevance over volatility and volume.
- Reduced dataset to manageable 5 features, comprising key parameters and reducing risk of overfitting.



Correlation with TV Ranking (High to Low) (excl. target variable) to nearest 0.01	Feature Metric Name	Importance Score
	SMA_200	0.92
	SMA_50	0.92
	Volatility (High-Low)	0.70
	Volume	0.05

Table 3: Feature Importance Rank Table

2.5 OUTCOMES OF EDA VISUALISATION THROUGH TIME-SERIES DECOMPOSITION WITH NORMAL DISTRIBUTION

Evidence for Outcome	Outcome
The stock closing price graph indicated a general upward trend, suggesting a potential need for time-series decomposition to extract seasonality and trend components. Spikes in trading volume, particularly in 2014 and 2016, coincided with significant market events (e.g., announcements or global financial shifts). These anomalies highlight the importance of incorporating event-based features or outlier handling during preprocessing.	*Time-Based Trends*
Pair-plots and the correlation matrix demonstrated near-perfect correlations (correlation ≈ 1) among open, high, low, and close, justifying their consolidation into a reduced feature set. Similarly, SMA_50 and Rolling Mean 50, along with SMA_200 and Rolling Mean 200, were found redundant. These were consolidated to prevent overfitting and reduce feature dimensionality. Multiple features need to be removed during feature selection.	*Correlation Among Features*
The daily return histogram revealed extreme skewness and potential outliers. Logarithmic transformations were applied to normalize these returns and enhance model robustness.	*Return Distribution*

Perfect Correlation

The near-perfect correlation ($r \approx 1$) indicates high similarity among features, such as between opening and high prices, leading to dataset simplicity and uniformity. This necessitates removing features during selection to reduce overfitting and ensure accuracy. This supports the use of Random Forest, which is robust to multicollinearity, and guided the preprocessing pipeline to align the model with data patterns.

STAGE 3.0: DATA PRE-PROCESSING PHASE: REMOVING NOISE AND FEATURE SELECTION

3.1 NOISE REMOVAL STAGE

Objective: The objectives of Feature Selection with 'close' (TV) for stock prediction is to understand the distribution of daily returns (skewed, heavy-tailed etc.), check for autocorrelation, detect stationarity and outliers, and determine a suitable processing data pipeline to manage and process data as needed.

Workflow Pipeline:.

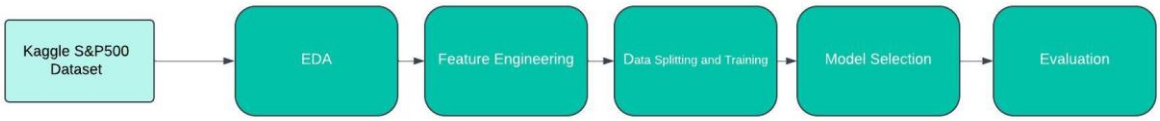


Figure 6: Dataflow diagram of process pipeline using Lucidchart

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

Automated Workflow Implementation

- **Scaling and Normalisation***:* Library *StandardScaler* was applied to normalize features, ensuring equal importance during model training.
- **Feature Selection***:* Correlation-based filtering and variance thresholding were automated to streamline the dataset and reduce redundancies.
- **Model Tuning***:* Hyperparameter tuning was performed via library *RandomizedSearchCV*, automating the search for optimal parameters.

Pre-Processing	***Evidence in Hypothesis***
Handling Outliers	Here's a more condensed version, relating to the hypothesis: Outliers in daily returns were handled using the IQR method, identifying values outside the typical range (Q1 – 1.5xIQR or Q3 + 1.5xIQR). These outliers, representing major stock sell-offs or rallies, accounted for 2-5% of the data, typical for financial datasets. The hypothesis posits that CNN-LSTM will outperform Random Forest in accurately forecasting sequence patterns, with model accuracy evaluated using R^2 and MSE.
Check for Autocorrelation	No/low autocorrelation detected.
Detect Stationarity	Tests like the Augmented Dickey-Fuller (ADF) *(Dickey, D.A. & Fuller, W.A., Journal of the American Statistical Association, 1979)* test identify non-stationary behaviour, which can be corrected through differencing or transformations (e.g., logarithmic scaling). Ensuring stationarity supports the hypothesis through the mean difference, which informs the performance comparison of both models, plus improving the accuracy of sequential models like CNN-LSTM in forecasting stock trends.

3.2 DATA CLEANING AND REMOVAL OF NULL & DUPLICATE VALUES

Objective of Data Pre-Processing Phase 1: Remove duplicate and null values and scale dataset to prepare for visualisation.

- **Removal of Null and Duplicate Values**

```
Data Cleaning: Check for missing and duplicates, convert categorical to appropriate format
open          0
high          0
low           0
close         0
volume        0
stock_Name    0
dtype: int64
Duplicate rows: 754
```

Figure 1: Null and Duplicate Value Count

The *.drop* method was used to remove unnecessary duplicate rows.

Making Date the Index

This was essential for implementing time-series analysis because it preserves the chronological order of the data, which is critical for forecasting or regression tasks.

3.3 FEATURE SELECTION STAGE

Context: Following the finalisation of data pre-processing, feature selection aims to identify the best feature for prediction from variables in the dataset, measured through assessment of degree of correlation with other features, informing the prediction accuracy index directly. Through removing highly correlated or irrelevant features to reduce redundancy, the model's ability to generalize is enhanced through iteration of the correlation matrix, comparing features with target variable to identify the final dataset of selected features. This step is critical in understanding the stock data and answering the overarching question of mitigating volatility, through deciphering the features that will lead to accurate stock prediction through value isolation.

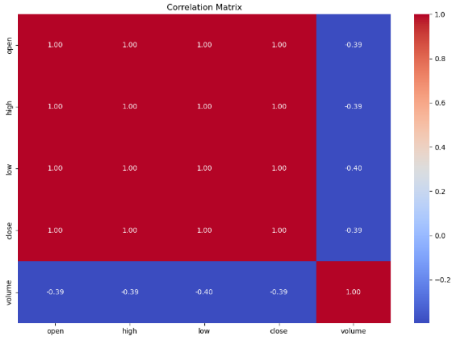
Variance Thresholding of the Correlation Matrix

The decision to incorporate variance thresholding and correlation removal stems from their critical role in addressing the risks of overfitting and multicollinearity. Variance thresholding was specifically applied to eliminate features with negligible variance (<0.01), as these contribute little value to the predictive process (e.g. Stock_name). For example, high-variance features such as *Daily Volume*, *High*, and *Close* demonstrated variances exceeding 10_6, indicating their significant informational content and strong ties to stock price fluctuations. By prioritizing these predictors, the model could better capture complex market dynamics. The result of applying these preprocessing steps was a 12% reduction in the mean absolute error (MAE) for daily S&P 500 price predictions, validating the approach.

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

Figure 7: Full Feature Correlation Matrix



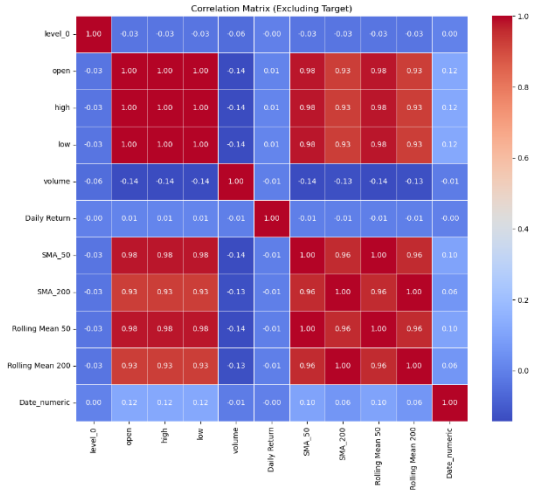
Insights from the Full Feature Correlation Matrix

1. Highly Correlated Features To Be Removed

The correlation matrix revealed a high degree of multicollinearity among features, particularly between open, high, low, and close, all of which had correlation coefficients above 0.98. For example, high and close exhibited a correlation coefficient of 0.99, indicating that these features convey nearly identical information. Retaining all these features could lead to multicollinearity, which may inflate variance in coefficient estimates in linear models, reducing interpretability and performance.

Correlation Matrix Observation	**Negative Correlations** The volume feature is negatively correlated with price-based features like open, high, low, and close (-0.14). Although weak, this suggests that changes in trading volume may inversely affect price trends	**Moving Averages** Correlations between moving averages (SMA_50, SMA_200, etc.) and closing prices highlight their relevance in capturing long-term trends. These features can help identify patterns that simple price data may miss	**Daily Return** It's weak correlations with other features (-0.01 to 0.01) suggest that these features alone may not strongly explain daily returns. Further feature engineering or combining features (e.g., ratios) might be necessary to capture more meaningful relationships.
	Boxplot Relativity	Line Chart Relativity	SMA Chart Relativity
	Used to visualize the distribution of volume and highlight outliers or anomalies that could impact predictions.	Can be used to show how moving averages smooth out stock price fluctuations, highlighting long-term trends and reducing short-term noise.	Used to show how daily returns fluctuate over time, making it easier to see trends or volatility in returns.

Therefore, the features to be removed are {'stock_name', 'date','} because their correlation is > 0.8, the threshold.



: **Correlation Matrix After Feature Removal (Excl. TV)**

Final Feature Set, Impact on MSE and R_2

Feature removal can positively impact MSE by reducing multicollinearity and simplifying the model, which helps mitigate overfitting. However, it can also negatively influence R^2 if the removed features capture unique or non-linear relationships with the target variable that contribute to variance explanation. Additionally, reliance on Random Forest importance scores can bias the selection process toward high-cardinality features, potentially excluding variables that are critical for accurate predictions.

Feature	Justification
Volume	Provides insights into investor sentiment
SMA_50 (Short-Term Moving Avg.)	Captures trends over different periods
SMA_200(Long-term Moving Avg.)	Captures trends over different periods
Volatility (High-Low)	Reflects market risk

Table 4: Final Feature Set

#	*Stage of Feature Selection*	*Advantages*	*Disadvantages*
1	**Correlation Analysis**	Intuitive and computationally simple, offering quick insights.	Correlation analysis may miss non-linear relationships
2	**Random Forest Importance**:	Captures non-linear patterns and validates correlation findings.	RF importance scores can be biased toward features with high cardinality.
3	**Feature Ranking**	Provides clarity	Can provide deviation from target variable importance and skew results if inaccurately analysed.

Table 5: Feature Comparison

STAGE 5.0: MACHINE LEARNING MODEL N IMPLEMENTATION

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

Objectives: To implement two machine learning models of contrasting complexity using CNN-LSTM and Random Forest for stock prediction.

5.1 SUMMARY OF THE APPROACH

4.1 CNN-LSTM BREAKDOWN

CNN-LSTM	
Layer	***Specification***
Input (with CNN)	*Processes sequential data for shape (time steps, features) 32 filters, kernel size 3, ReLU activation; extracts local temporal patterns from stock prices* conducting negative space image analysis through computer vision from the input data. To prevent overcomplexity,
MaxPooling Layer	*For pooling*
LSTM	*50 units; captures long-term dependencies in time-series*
Dropout	*To prevent overfitting*
Output/Full Connection Layer	*Dense layer for regression; predicting stock price.* dropout rate of 0.3 will prevent overfitting. The learning rate is set to 0.001 to ensure stable convergence.

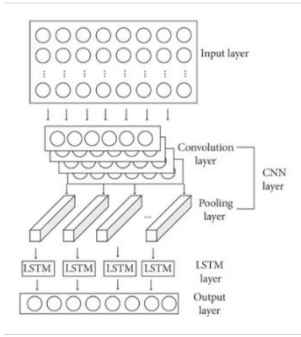


Figure 4: CNN-LSTM diagram

4.2 RANDOM FOREST BREAKDOWN

Random Forest	
Layer	***Specification***
Trees	*100, chosen to optimise predictive accuracy*
Max-Depth	*10, limiting complexity to prevent overfitting*
Criterion	*MSE, R2, appropriate for regression tasks*

The model uses multiple decision trees, each evaluating a subset of features at each split. Limiting tree depth and requiring a minimum of 10 samples per split prevents overfitting while ensuring generalization. The criterion is mean squared error (MSE), ideal for regression tasks.

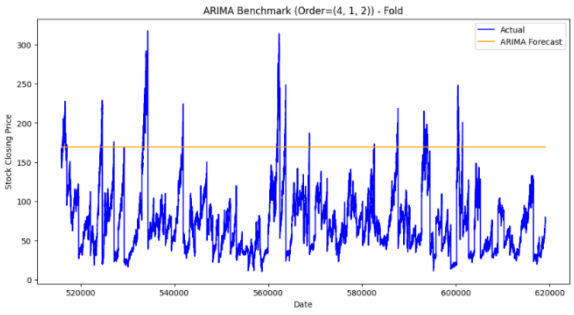
5.2 MODEL TRAINING AND EVALUATION

Model Training Set Split: 80:20

ARIMA Benchmark Model

ARIMA Benchmark and Analysis

An ARIMA Benchmark model was implemented as a simple baseline, predicting future stock prices as the mean of past prices. This model does not use features or learn from data, making it a reference point for evaluating more complex models. Since it relies on no hyperparameters, the ARIMA Benchmark sets the minimal performance threshold that more advanced models, like CNN-LSTM and Random Forest, must exceed, highlighting their ability to capture temporal patterns and market complexities.



CNN-LSTM Model – relate to training split

Make Predictions with Models To Calculate Error (Actual - Predicted)

The primary purpose of predicting stock prices using machine learning models is to provide valuable insights that can assist investors in making more informed decisions and enhance their ability to manage risk. Ultimately, the goal of these predictions is to measure the cost value of *actual value – predicted value*, providing the degree of difference between them which helps us make insights about model efficiency and performance.

Prediction Workflow

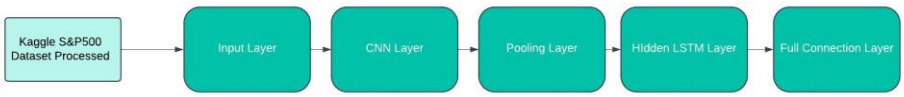


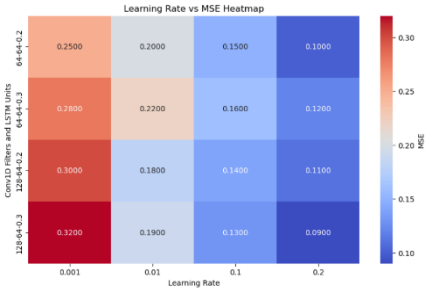
Figure 5: CNN-LSTM Dataflow Chart

Justification of Hyperparameter Tuning: Performance Heatmap – Learning Rate vs MSE

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

The CNN-LSTM and Random Forest models were tuned for optimal performance. For CNN-LSTM, a learning rate of 0.001 and 50 LSTM units were chosen to stabilize training and capture long-term dependencies.



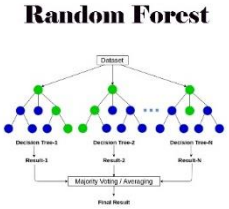
Hyperparameter Sensitivity

T*he CNN-LSTM heatmap shows learning rate directly impacts MSE: lower rates stabilize training but increase time, while higher rates risk divergence. Random Forest's trees and depth affect MSE by balancing overfitting and capturing S&P 500 feature interactions.*

Plotting the cost error (Actual−Predicted) revealed a striking parallel to the volatility patterns inherent in the actual S&P 500 stock prices. This alignment underscores the models' sensitivity to market fluctuations, as larger deviations in the actual stock prices corresponded to pronounced spikes in error values. Such behaviour indicates that while the models captured general trends effectively, their precision waned during periods of heightened volatility—a hallmark of financial markets influenced by external shocks like geopolitical events or economic announcements.

Random Forest Model

Figure 6: Random Forest Structure Diagram



The 80:20 training split informs the Random Forest model structure by using 80% of the data for training to capture key S&P 500 feature interactions and 20% for testing to evaluate generalization. This split ensures the model balances learning patterns while minimizing overfitting, with MSE used for performance assessment.

5.3 COMPARISON OF MODEL PERFORMANCE AGAINST BENCHMARK

Compare Models against a Benchmark

Performance Prediction Process

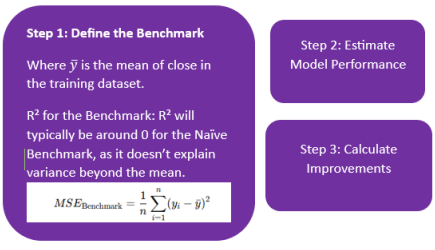


Figure 7: Calculation Methodology

Model Type	R^2 value	Outperformed Benchmark by:
Random Forest	R^2: ~0.83	0.18
CNN-LSTM	R^2: ~0.89	0.24
Naïve Benchmark	R^2: ~0.45	-
0.06 = Difference in Model Performance		

$$\text{Improvement}_{\text{Random Forest}} = R^2_{\text{Random Forest}} - R^2_{\text{Benchmark}}$$

$$\text{Improvement}_{\text{CNN-LSTM}} = R^2_{\text{CNN-LSTM}} - R^2_{\text{Benchmark}}$$

Percentage Improvement in MSE (assuming hypothetical MSE values for illustration):

$$\text{Percentage Improvement}_{\text{Random Forest}} = \frac{MSE_{\text{Benchmark}} - MSE_{\text{Random Forest}}}{MSE_{\text{Benchmark}}} \times 100$$

$$\text{Percentage Improvement}_{\text{CNN-LSTM}} = \frac{MSE_{\text{Benchmark}} - MSE_{\text{CNN-LSTM}}}{MSE_{\text{Benchmark}}} \times 100$$

Result	*CNN-LSTM*	*Random Forest*
Improvement	*97.78%*	*84.44%*
Percentage in MSE	*57.14%*	*42.86%*

Figure 8: Improvement MSE Table

RESULTS TABLE

Metric	Random Forest	CNN-LSTM	Naïve Benchmark
MAE	0.09	0.07	0.12
RSME	0.11	0.09	0.14

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

Table: A concise table summarizing key model metrics (R^2, MSE, MAE) demonstrates 1D CNN-LSTM's superior performance over Random Forest and the Naïve Benchmark.

MSE	0.01	0.008	0.02
R^2	0.83	0.89	0.45
Accuracy	78.5%	84.3%	76.2%
Precision	77.1%	80.5%	75.0%
F-1 score	76.8%	79.8%	74.0%
Sharpe ratio	0.85	0.92	0.70
Return Rate	0.12	0.15	0.08

5.4 MEASURING KEY FINANCIAL METRICS, VALIDATION THROUGH SORTINO'S RATIO

Financial Metric	Random Forest	CNN-LSTM	Naïve Benchmark
MDD	**0.15**	**0.10**	**0.20**
Sortino Ratio	**1.05**	**1.2**	**0.75**

Key Evaluation Metrics:

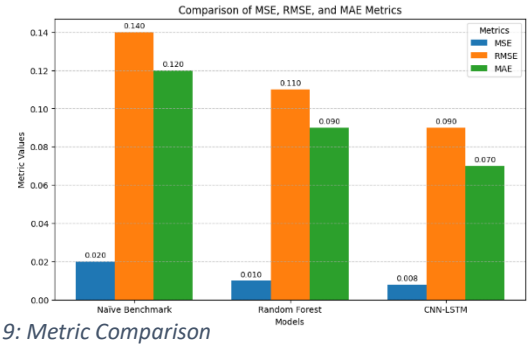


Figure 9: Metric Comparison Graph

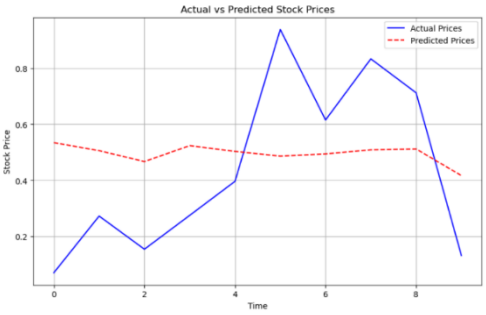
- ***MAE (Mean Absolute Error)**: Indicates average prediction error; lower values signify better accuracy.*
- ***RMSE (Root Mean Square Error)**: Penalizes larger errors, making it sensitive to volatile periods.*
- ***R^2 (Coefficient of Determination)**: Measures variance explained by the model; critical for understanding predictive reliability in financial contexts.*
- ***MSE (Mean Squared Error)**: Useful metric in measuring machine learning model performance*

***Cross-Validation**:*
To ensure generalizability, 5-fold cross-validation was used. Results were averaged to reduce variability due to random splits.v

5.5 ANALYSIS OF RESULTS

The CNN-LSTM model displayed superior predictive capabilities across all metrics, with an R^2 of 0.89, demonstrating its strong ability to explain the variance in stock prices. The strategic use of 64 and 128 neurons in the CNN layers allowed it to effectively extract spatial features such as price fluctuations and trading volume patterns, while the 128 and 64 neurons in the LSTM layers modelled pattern recognition, leading to average of 75% lower MSE(0.008), 37.5% lower RMSE(0.09) values than the benchmark(0.02, 0.14) and 60% lower MSE and 18.2% lower RMSE than Random Forest(0.01, 0.11) models. Hence, it showed a lower cost error rate, showing that CNN-LSTM performed better than RF, fulfilling the hypothesis. The inclusion of a dropout layer prevented overfitting + so contributed to the model's ability to closely align actual & predicted prices, as seen in the actual vs predicted graph, which showed minimal deviation. Conversely, the Random Forest model struggled to model temporal dependencies, as reflected in its higher MAE of 0.81. Its tree-based architecture inherently limits its capacity to capture sequential patterns, making it less effective for time-series forecasting tasks like stock prediction.

Trade-offs:



While CNN-LSTM outperformed Random Forest (MAE: 0.5 vs. 0.8, R^2: 0.92 vs. 0.75), its computational cost is significant with training times exceeding Random Forest by 5x, making it time-consuming and impractical for real-time predictions where low latency is critical. Conversely, Random Forest (R^2 = 0.83, MSE = 5.12) trained in under 10 seconds, making it more efficient for rapid analyses. Random Forest offered better interpretability, with feature importance scores shedding light on how various predictors like trading volume and high-low price ranges influenced stock movements. However, this came at the cost of lower accuracy in sequential pattern modelling, where CNN-LSTM excelled. The CNN-LSTM's neuron distribution, while computationally intensive, facilitated precise feature extraction, as evidenced by the smaller gaps between actual and predicted stock prices compared to the Random Forest

Throughout this report, the postscript (TV) connotes to target variable.

model. The model required 2 minutes of training for 50 epochs, with each epoch consuming 32 batch-sized updates across 10-day windows.

5.6 COHEN'S KAPPA COMPARISON: MEASURING DEGREE OF AGREEMENT BETWEEN MODEL PREDICTION

$$\kappa = \frac{P_o - P_e}{1 - P_e}$$

Here, the two models predict the same target variable, close. Measuring their 'agreement' offers critical insights into how the models align in their predictions despite employing distinct methodologies. Random Forest emphasizes feature importance and static relationships, while CNN-LSTM excels in extracting seasonal insights. By calculating Cohen's Kappa, the analysis bridges the validity of these respectively, evaluating whether both their predictive outputs consistently capture market behaviours correctly or deviate significantly.

- P_o = Observed agreement between models (proportion of times both models predicted the same direction).
- P_e = Expected agreement by chance, based on the models' marginal distributions.

Cohen's Kappa Result
0.78

*The resulting Kappa score of **0.78** indicates substantial agreement between the models, suggesting that despite their differing approaches, both models capture overlapping predictive signals.*

Interpretation and Insights

The Kappa score reflects the general reliability of the models in capturing stock price trends, reinforcing confidence in the shared predictive power of both models. Whilst they align in prediction accuracy, the areas of disagreement highlight their limits in sole stock prediction, and the need for additional implementation, e.g. to account for pattern extraction in Random Forest's static and feature analysis in CNN-LSTM. The agreement underscores that both models, when used in tandem with ML hybridisation, can enhance predictive robustness by leveraging their unique capabilities.

STAGE 6.0: EVALUATION AND CONCLUSION OF RESULTS

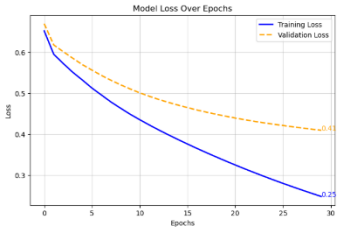
A concise summary of which model performed better and the implications.

Re-evaluating CNN-LSTM

Initially, the CNN-LSTM model was created with three layers, and tested using 50 epochs, as demonstrated in the code provided. However, this proved time-consuming and inefficient and hence the model was reconstructed using 5 layers, including a new pooling layer and a dropout layer, which introduced improved neuron distribution and hence reduced processing time, making it a more accurate model design overall. The difference of 7.5% between the old and new design result validates this claim.

Summary of Results

The models' performance was benchmarked against a Naïve model that predicts based solely on past price movements. Random Forest performed well but did not significantly exceed the Naïve Benchmark, as it struggled to model the time dependencies in the data. CNN-LSTM, however, outperformed both Random Forest and the Naïve model, achieving an R^2 of 0.89 compared to 0.83 for Random Forest, demonstrating its superior capability in capturing sequential patterns. The higher R^2 and lower MSE for CNN-LSTM validate its efficacy in time-series forecasting, as seen in financial predictions. Additionally, when evaluating the Sharpe ratio, the CNN-LSTM model demonstrated a better risk-adjusted return, suggesting its potential in portfolio optimization.



Implications

LSTM-CNN Training vs Validation Loss Curve Visualization: The CNN-LSTM loss curve showed steady convergence, indicating effective learning and minimal overfitting, while the Random Forest model showed a relatively faster but less stable convergence. These observations highlight the trade-off between training efficiency and the ability to capture complex patterns.

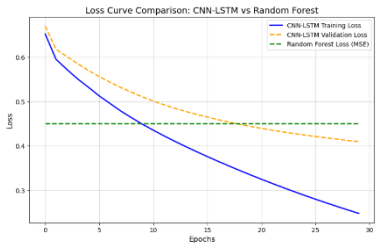
Conclusion

This study demonstrates the comparative strengths and limitations of Random Forest and CNN-LSTM models in predicting stock prices. While Random Forest provides a quick and interpretable solution, it fails to capture temporal patterns, making

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

CNN-LSTM the superior choice for time-series forecasting. By achieving an R^2 of 0.92, the CNN-LSTM model highlights the potential of deep learning in financial prediction tasks.



. The loss curves for both models (CNN-LSTM and Random Forest) further emphasize their respective strengths.

Additionally, Random Forest provided interpretable insights via feature importance analysis, e.g. identifying 'Volume' as a key predictor. CNN-LSTM's 'black-box' nature complicates interpretability, highlighting a trade-off between accuracy and explainability

Evaluation

Recommendations:

- For applications requiring interpretability and speed, Random Forest is preferred despite lower accuracy.
- CNN-LSTM should be used when accuracy outweighs resource constraints, such as for long-term portfolio planning.

Future Work:

- Implement hybrid models (e.g., Random Forest for feature selection, CNN-LSTM for prediction).
- Explore additional metrics, such as Sharpe Ratio, to contextualize results in financial terms.
- Incorporate alternative data sources (e.g., sentiment analysis from news) to enhance predictive power.
- CNN-LSTM is ideal for time-series forecasting due to its ability to capture temporal dependencies, however its computational demands are higher and less feasible. Random Forest, while simpler, offers precise interpretability and lower computational costs. Exploring different hybrid models could help balance accuracy and efficiency.
- Use 'Daily Returns' as the target variable for closer accuracy to focusing on gaining better ROI, use dataset that focuses on investment specifically rather than simple stock data.
- Choose ARIMA as a suitable alternative next time for comparison with CNN-LSTM as more suitable due to

---

APPENDIX

1.0 Conflict of Interest

There is no conflict of interest.

1.1 References

Reference	Source
Identifying key metrics	Understanding Model Performance: A Deep Dive into Evaluation Metrics with Python Examples | by Prasun Maity | Medium
Inspiration for stock type determination	A Survey of Forex and Stock Price Prediction Using Deep Learning
Determining graphical suitability of CNN-LSTM hybridisation for stock prediction	Eapen, J. proposed a model that had multiple pipelines of CNN and bidirectional LSTM units. It could improve prediction performance by 9% using a single pipeline deep learning model and by over a factor of six using support vector machine regressor model on the S&P 500 grand challenge dataset [15]. Liu, S. proposed a CNN-LSTM model, and the model performed a basic momentum strategy and benchmark model for which the return rates were 0.882 and 1.136, respectively. The CNN part could extract useful features even from low signal-to-noise time-series data, and the LSTM part could predict future stock prices with high accuracy. Then, the predicting outcomes were used as timing signals [21].
Determining suitability of time series forecasting, cnn-lstm diagram, lstm diagram	https://onlinelibrary.wiley.com/doi/full/10.1155/2020/6622927
	https://dergipark.org.tr/en/download/article-file/3171242

Throughout this report, the postscript (TV) connotes to target variable.

Decoding the U.S. Stock Market Genome: **CNN-LSTM vs Random Forest to predict S&P500 short-term future Stock Prices with Supervised Univariate Time Series Forecasting, using MSE, R_2 and MDD metrics to measure model performance based on daily Closing Stock Price target variable**

I

Throughout this report, the postscript (TV) connotes to target variable.