Homework2

RNN Models

Elman – Jordan

Course:

Deep Learning

Supervisor:
Dr. Mansoori

Atefe Rajabi

Spring 2024

Index

Problem Definition

This project aims to predict the Tehran Stock Exchange (TSE) Index using simple Recurrent Neural Networks (RNNs) based on historical stock data. By leveraging the sequential processing capabilities of Jordan and Elman networks, the model will analyze four years of daily opening, closing, and final index values to forecast future prices. The challenge is to effectively capture temporal dependencies inherent in stock data to optimize prediction accuracy. The effectiveness of the model will be validated by comparing its predictions against the actual index values, with a focus on improving investment strategies and economic research related to the TSE.

Data Exploration

Visualization tools and statistical summaries provide insights into the dataset’s characteristics, which can inform subsequent analyses and decision-making processes.

Importing Libraries

- `pandas`: Manages data in DataFrame structures, facilitating data manipulation.

- `matplotlib.pyplot` and `seaborn`: Powerful visualization libraries that help in creating a wide range of graphs and plots.

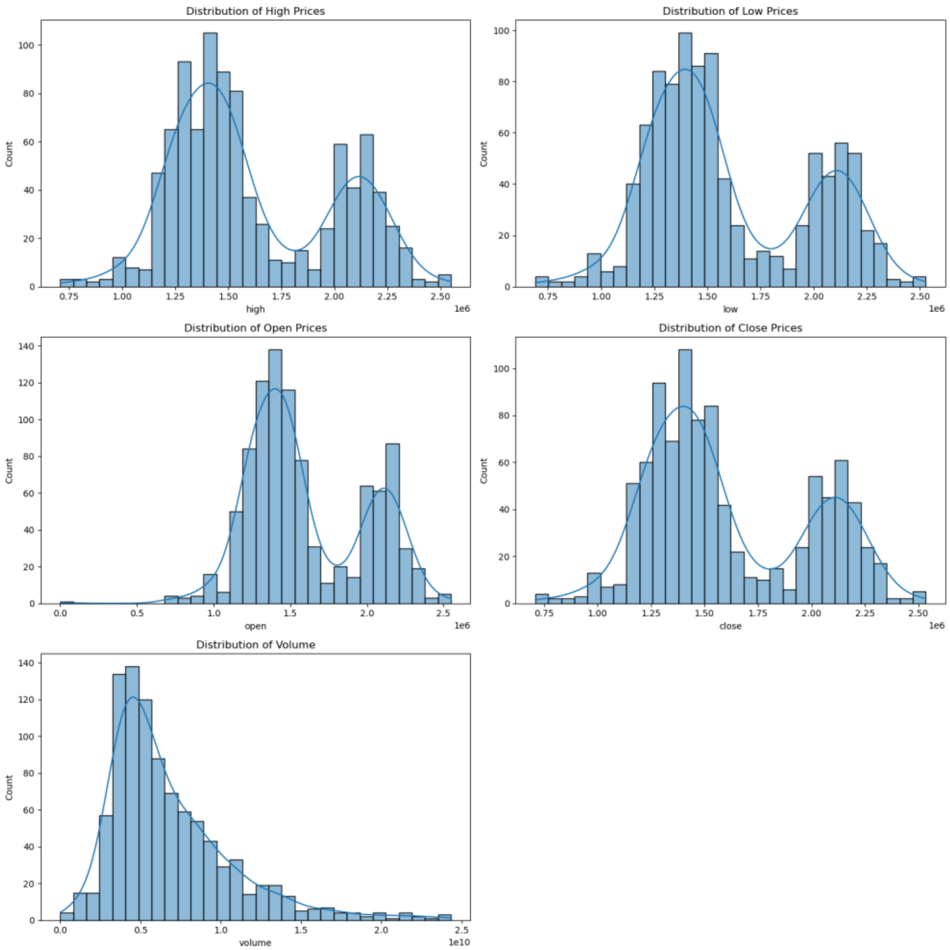
- `sklearn.preprocessing`: Contains scaling functions like StandardScaler and MinMaxScaler that normalize or scale data.

Visualizing Data Distributions

Histograms and density plots are created for various columns like ‘high’, ‘low’, ‘open’, ‘close’, and ‘volume’. These visualizations help in understanding the distribution of values within each financial metric:

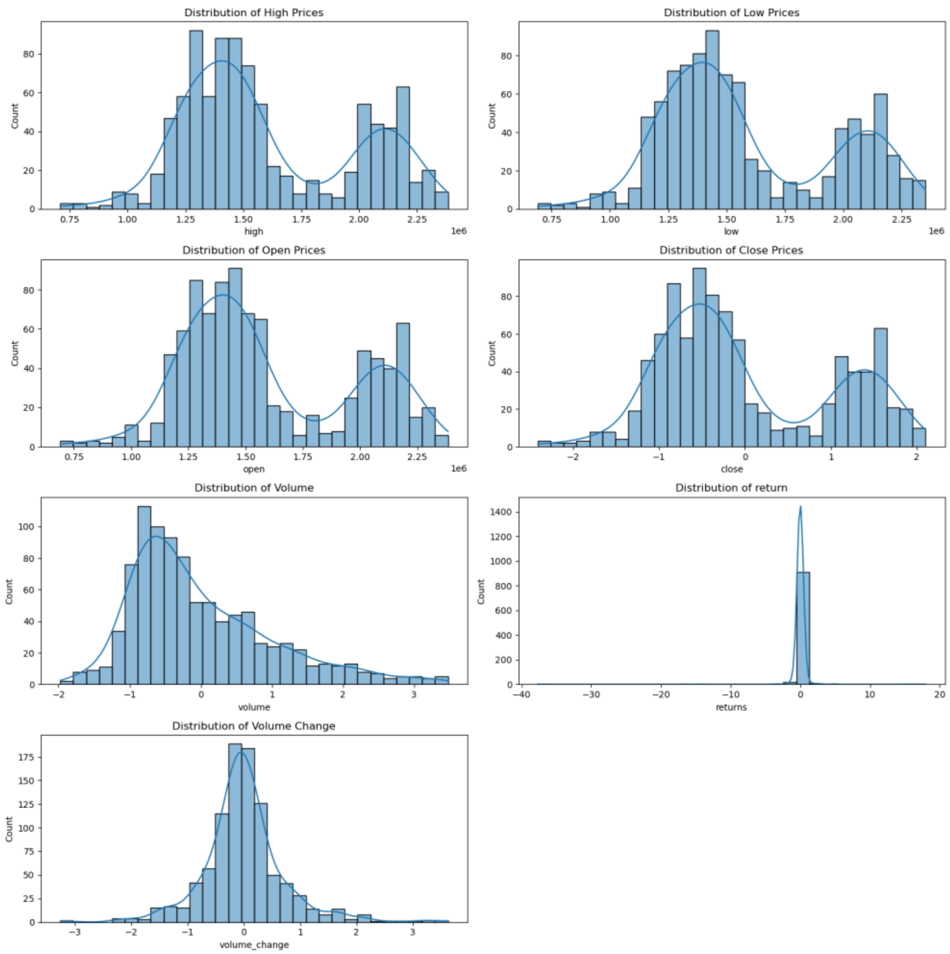
- Histograms: Show the frequency distribution of values within each metric, using bins to represent the data range.

- Density Plots (KDE): Provide a smoothed version of the histogram, giving a clearer view of the data’s distribution shape.


Distribution of High Prices
Distribution of Low Prices
Distribution of Open Prices
Distribution of Close Prices
Distribution of Volume
Count
high
low
open
close
volume
1e6
1e10

Handling Additional Data (Cleaned Dataset)

- Visualizations for the cleaned dataset extend to include newly engineered features such as ‘returns’ and ‘volume change’, illustrating their distributions.



Normalization

This part involves several critical steps: data loading, cleaning, visualization, normalization, feature engineering, correlation analysis, and data export.

1. Importing Libraries

- `pandas`: For data manipulation and analysis.

- `seaborn` and `matplotlib.pyplot`: For data visualization.

- `numpy` and `scipy.stats`: For numerical operations and statistical calculations.

3. Data Cleaning

- Missing Values: First step is to check for and display the number of missing values in each column of the DataFrame.

- Duplicate Rows: It checks for duplicate rows in the dataset and displays their count.

4. Outlier Detection and Handling

- Visualization of Outliers: Box plots are created for each column in the dataset to visually identify outliers.

- Z-score Method for Outliers:

  - Outliers are detected using the Z-score method, considering rows with any column having a Z-score greater than 3 as outliers.

  - These outliers are then removed from the dataset.

5. Feature Engineering

- Normalization:

  - The ‘close’ and ‘volume’ columns are normalized using the standardization technique (mean 0 and standard deviation 1).

- Creation of New Features:

  - `returns`: Calculates the percentage change in the closing price from one day to the next.

  - `volume_change`: Computes the difference in trading volume from one day to the next.

6. Correlation Analysis

- A correlation matrix for the DataFrame (excluding any non-numeric columns) is computed and visualized using a heatmap. This step helps identify the strength and direction of relationships between different features.

RNN Models:

Elman Networks

Elman networks, or Simple Recurrent Networks (SRNs), feature a hidden layer that feeds back into itself, providing the network with a form of memory. This feedback loop allows the network to maintain state information across time steps, making it good for tasks where context or the sequence history is important. In terms of stock market prediction, which often relies on the analysis of trends and patterns over time, Elman networks could be advantageous as they can remember and process inputs from previous time steps to forecast future prices.

Jordan Networks

Jordan networks also have a form of memory, but it operates differently. In a Jordan network, the output from the network feeds back into the network as input. This structure can be useful for tasks where the future state is a modification or continuation of the previous state. However, in stock price prediction, where the goal is often to predict the next value based on a series of past values, the feedback of the output might not provide as much relevant context as the feedback of the hidden state itself.

Prediction

Library Utilization

Key Python libraries are utilized for data manipulation, numerical operations, deep learning model creation, and visualization. The environment is configured to ensure consistency and reproducibility in results through the setting of seeds, stabilizing the randomness in data shuffling and model weight initialization.

Data Handling and Preparation

Financial datasets typically contain sequential data such as daily closing prices, trading volume, and returns. For effective modeling:

- The dataset is indexed and organized by date to maintain the sequence order.

- Data is transformed into sequences that represent windows of consecutive data points. These sequences help the model learn from past information to predict future values.

Optimization of Window Size

To enhance model accuracy, different window sizes are tested to determine the optimal number of past days the model should look at to predict future stock movements. This involves:

- Training a simple neural network for each window size.

- Evaluating each model's performance to select the window size that minimizes error, specifically using mean squared error as the metric.

Model Development and Training

Two types of RNN architectures are explored:

- Elman RNN

- Jordan RNN

Both models are trained using historical stock data, and their performance is tracked through the calculation of loss metrics during training, which helps in understanding how well the model is learning.

Performance Evaluation and Visualization

- Models are evaluated based on their prediction accuracy and the loss during training phases. Training and validation loss graphs are plotted over epochs to visualize learning trends.

- Predictions from the models are graphed alongside actual data to visually assess how well the models predict new data.

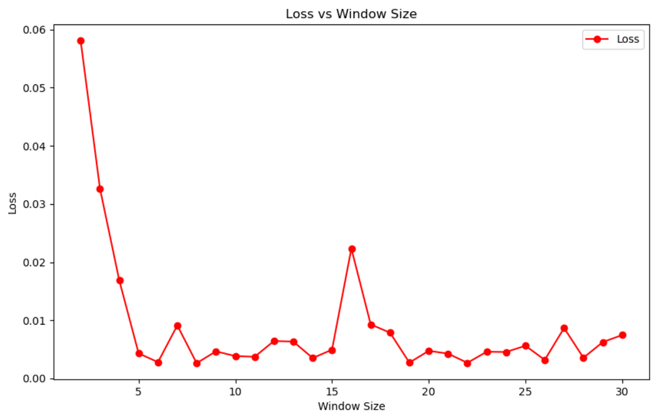
- The architecture of each model is visualized to provide insights into the model’s structure and the flow of data through the network.

Outcome and Reporting

The process aims to produce models that can accurately predict stock market trends, assisting in investment decisions and risk management. Outputs including detailed loss metrics, prediction visualizations, and architectural diagrams are systematically organized, providing a comprehensive understanding of model behaviors and capabilities.

Results:

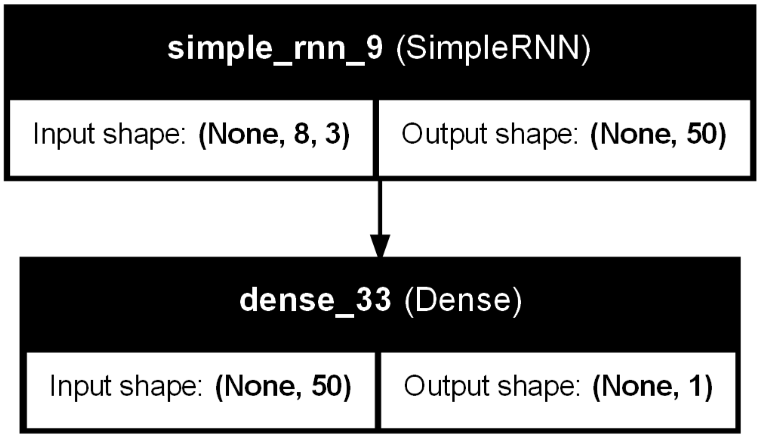
Exploring different window size for finding best match:



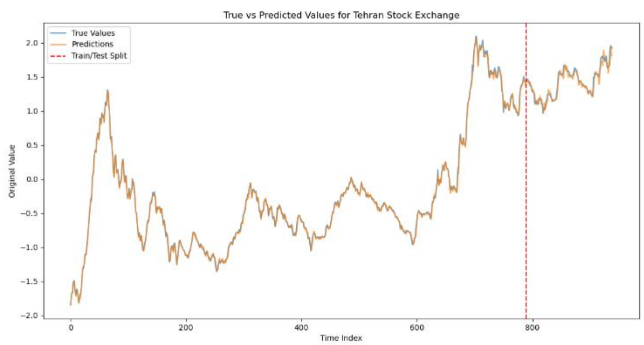
A window with size 8 is seemed to be the best in minimizing loss function.

Elman:

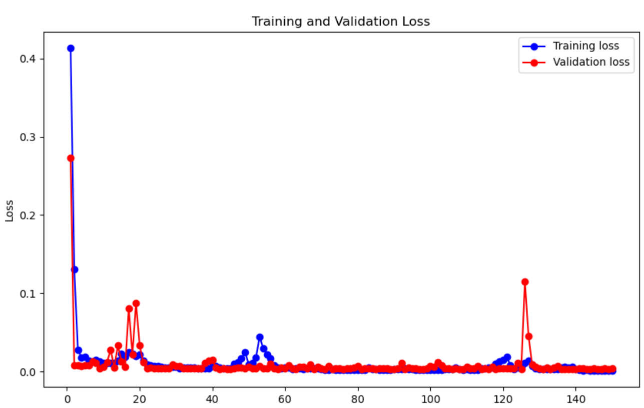
Elman Architecture:



Elman Prediction:

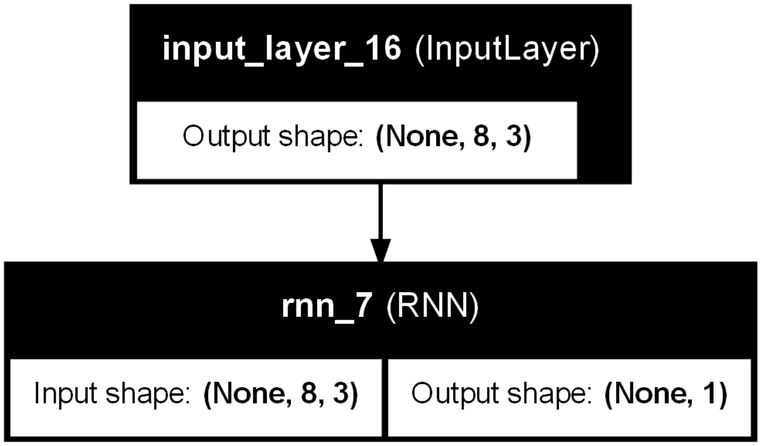


Loss over epochs:

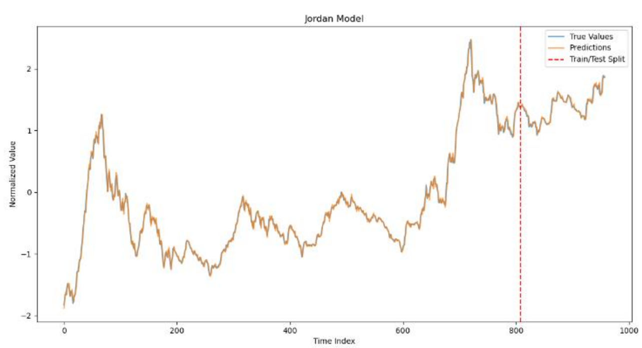


Jordan:

Jordan Architecture:



Jordan Prediction:



Jordan Loss over epochs:

Discussion:

Choosing between Elman and Jordan networks for stock market price prediction depends on the specific characteristics of each network and the nature of the task at hand.

Which is Better for Stock Market Price Prediction?

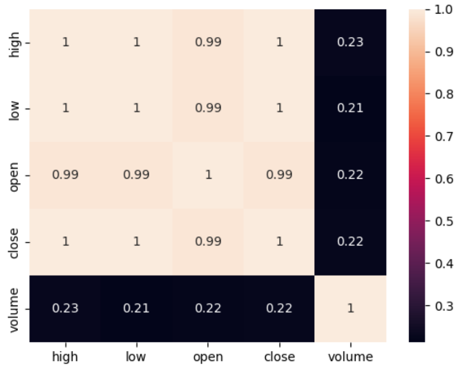
- Relevance of Input Data: Elman networks might be more suitable if the prediction relies heavily on the intricate patterns and relationships in the input data over time, as they can retain detailed information from the hidden states of previous time steps.

- Nature of Task: Jordan networks could be considered if the model benefits from directly leveraging the output as part of the input for the next step, which might be more aligned with tasks where the next state is a direct progression from the previous output.

For stock market price prediction, Elman networks generally might be more suitable due to their ability to integrate and learn from the hidden state feedback, which more directly captures the nuances of past market behavior and trends. However, the best choice can also depend on specific modeling goals, data characteristics, and other design considerations, such as how the network is structured, trained, and tuned.

Feature selection:

Correlation between features:



Based on the correlation analysis, the features ‘low’, ‘high’, and ‘open’ do not provide much new information, so they are not used. Instead, new features that have been mentioned previously—’returns’ and ‘volume change’—are introduced.