REPORT

I. Introduction

The purpose of this report is to describe the journey through Tasks 1 to 5, focusing on the experiments, challenges, and findings for each step. The tasks aimed to apply deep learning models to stock price prediction, trading signal identification, and portfolio optimization, progressing systematically from basic predictions to actionable investment decisions. The report reflects on both successes and challenges to provide a comprehensive overview of the learning process.

II. Task-by-Task Journey

1.1. Nasdaq Stock Price Prediction

Tackling Task 1 turned out to be unexpectedly challenging, despite being the simplest among all tasks. With no prior experience in stock price prediction, I had to invest significant effort in understanding the fundamentals of stock prices and their behavior in financial markets.

a. Understanding Stock Prices and Dataset Exploration

One of the initial hurdles was understanding the significance of features like Open, Close, Low, High, Adjusted Close, and Volume in stock price prediction. These features form the backbone of a reliable prediction model. I spent most of my time preparing and exploring the dataset, using the yfinance package to create an overview table of stock data for various companies. This table was crucial in defining criteria to select relevant companies for analysis.

However, processing data from multiple companies posed significant computational challenges. Concatenating features across several companies often became time-consuming and resource-intensive. This issue forced me to rethink how to streamline data preparation for efficient model training.

b. Narrowing the Scope to Technology Sector

Initially, my plan was ambitious—I aimed to develop separate models for multiple market segments, including technology, financial services, healthcare, and industrial sectors. My goal was to visualize predictions for each segment and identify correlations during significant social or economic events. However, I eventually decided to focus on the technology sector. This decision

was driven by my familiarity with the sector and the availability of sufficient, high-quality data, which allowed me to proceed with a more focused and manageable approach.

c. Splitting Time-Series Data

Splitting time-series data for training and testing was another major challenge. Unlike traditional datasets, time-series data cannot be split randomly (e.g., 80/20 or 70/30) because doing so disrupts the temporal order. Instead, the data must be split sequentially to preserve the chronological integrity of the information. Initially, I found this concept confusing, but I soon realized that maintaining the temporal sequence was critical for producing reliable predictions. Ensuring no data leakage between the training and test sets became a key focus during this stage.

d. Normalization and Denormalization Challenges

Normalization of data was an essential step in preparing the dataset for model training, but it presented its own set of challenges. Applying the MinMaxScaler initially led to unusual outputs during model evaluation and plotting. These issues prompted me to write a custom normalization function, which offered more control and flexibility.

Denormalization, which is necessary to interpret predictions in their original scale, also required significant attention. This process helped me better understand how to evaluate the model's performance. Additionally, I learned the importance of always testing the model on the test set rather than the training set. By plotting results for both sets side by side, I was able to gain deeper insights into the model's accuracy.

e. Model Selection and Data Integration

I experimented with three different models—Conv1D, LSTM, and GRU. All three models performed reasonably well without significant fine-tuning, yielding similar results. However, combining data from multiple companies into a single dataframe introduced additional complexities.

Initially, I considered reshaping the data to have a uniform structure by filtering companies with datasets starting before the year 2000. While this approach ensured consistency in time frames, it also resulted in the loss of valuable information, limiting the model's ability to capture diverse patterns. After careful consideration, I opted to concatenate all company data sequentially into a single dataframe. This approach aligned with my goal of predicting the overall trend for the technology sector rather than focusing on individual company predictions.

By the end of this process, I had a functional dataset ready for training. This experience taught me critical lessons about data handling, model preparation, and prediction techniques, setting the stage for successfully tackling subsequent tasks.

1.2. Forecasting for kth Day

The kth-day forecasting subtask introduced a new challenge: creating time-window samples. Once I understood the importance of structuring input-output pairs in a sequential manner, completing this subtask became more straightforward. The process involved preparing the data by using sequences of past observations to predict stock prices for a specific day in the future. This required careful attention to ensure that the models could effectively handle the sequential nature of time-series data. By implementing this structure, I was able to train the models to forecast accurately for any kth day.

1.3. Predicting k-consecutive Days

The k-consecutive days forecasting subtask required extending the model to generate multiple outputs simultaneously. This involved modifying the output layer to produce k predicted values in one pass. Initially, I overlooked this adjustment in the fully connected layer, leading to errors during training. After addressing this issue, the model was able to perform well, producing reliable predictions for multiple consecutive days. The results demonstrated that the model could capture trends over extended time horizons, providing valuable insights into future stock price movements.

2. Vietnam Stock Price Prediction

Tackling the Vietnam stock price prediction tasks (Tasks 2.1, 2.2, and 2.3) followed a similar trajectory to the Nasdaq stock price prediction task. The main difference lay in the need to explore and acquire contextual knowledge about the Vietnamese market. This step was critical to understanding the unique aspects of the dataset and adapting my methodology accordingly. After some consideration, I chose the banking sector as my focus. This decision was driven by its familiarity and the availability of sufficient data for analysis, training, and testing.

The experience gained from Task 1 significantly streamlined the process for these subtasks. The training and testing methodologies remained the same as those applied to the Nasdaq dataset, which eliminated much of the trial-and-error phase. Tasks such as dropping missing values (using dropna) and general data preparation were smoother, thanks to the experience accumulated in the

previous task. These prior lessons allowed me to focus more on refining the models and interpreting the results rather than grappling with basic data handling challenges.

The performance of the models—Conv1D, LSTM, and GRU—was consistent with my findings in Task 1. All three models delivered comparable results, with no significant differences in accuracy or predictive capabilities. This reinforced the robustness of these architectures for stock price prediction tasks, regardless of the dataset.

One significant challenge arose during the computational phase. By this point, using Google Colab had become increasingly problematic due to resource limitations. Training the models on larger datasets required more processing power than Colab could provide, leading to delays and inefficiencies. To address this issue, I transitioned to running the models on a local machine equipped with a GPU, which I borrowed from a friend. This shift in computing resources greatly improved processing speeds and allowed me to handle the larger dataset more efficiently.

In summary, the Vietnam stock price prediction tasks were smoother and more manageable due to the foundation built during Task 1. The challenges were less about methodology and more about contextual understanding and computational power. These tasks reinforced my confidence in applying deep learning models to time-series data while also emphasizing the importance of scalable computational resources for future projects.

3. Trading Signal Identification

For Task 3, the goal was to identify buying and selling signals in the Vietnamese stock market. The process leveraged the knowledge and tools developed in previous tasks, with some modifications to address the specific challenges of trading signal prediction.

In **3.1: Buying Signal Identification**, I reused the time-window function from Task 2.1 to prepare the dataset. The model architecture was similar to earlier tasks, but the last layer was adjusted to use a sigmoid activation function to predict the probability of a buying signal. Initially, I used a threshold of 0.5 to classify signals, and the Conv1D model performed smoothly. Visualizations of the predicted probabilities against actual signals were straightforward to generate, and the results were consistent with expectations.

However, **3.2: Selling Signal Identification** introduced challenges due to significant class imbalance in the dataset. The data was heavily skewed, with approximately 90% representing no signal and only 10% representing a signal across training, validation, and test sets. This imbalance

made the default threshold of 0.5 unsuitable, as it skewed evaluation metrics. To address this, I calculated an optimal threshold to balance metrics like accuracy, precision, and recall. Additionally, I compared the performance of different models—Conv1D, LSTM, and GRU. Among them, the GRU model performed the best, particularly in managing the imbalanced data and capturing sequential dependencies effectively.

Overall, Task 3 demonstrated the importance of tailoring evaluation strategies, such as threshold tuning, to address class imbalances. The GRU model proved to be the most robust for this task, and the reusability of functions and methods from previous tasks helped simplify the implementation process.

4. Portfolio composition, risk management and portfolio optimization for Vietnam market

In Task 4, I reused and fine-tuned the time-window function from Task 2.3 to incorporate additional features, adjusting the window size and prediction days to calculate the percentage change between the latest actual price and the seventh-day predicted price. Two key metrics were computed: **Percentage Change (%)**, indicating potential returns, and **Risk (Range)**, reflecting price volatility. These metrics were normalized and combined using the formula:

$$\text{Combined Score} = (\text{weight\_return} \times \text{Normalized Return}) + (\text{weight\_risk} \times (1 - \text{Normalized Risk}))$$

This score balances returns and risks, allowing investors to rank companies based on their investment potential. Higher scores represent better opportunities, combining high returns with low risks. This method ensures optimized, data-driven portfolio selection, tailored to investor preferences through adjustable weights.

III. Conclusion

Each task built on the previous one, advancing from simple stock price predictions to actionable portfolio recommendations. Task 1 established a foundation in data preparation and model training, while Task 2 demonstrated the adaptability of these methods to the Vietnamese market. Task 3 introduced the challenges of trading signal identification, emphasizing the importance of balancing metrics for imbalanced datasets. Task 4 successfully bridged predictions with investment strategies, using normalized metrics to optimize portfolios.

This journey highlighted the importance of iterative learning, leveraging past work, and adapting to challenges like resource limitations and data imbalance. By the end, I developed robust methodologies for applying deep learning models to real-world financial problems, achieving a balance between predictive accuracy and practical utility.