Identifying Financial Trends for Effective Investment Portfolio Management using Machine Learning Techniques

Aayush Kamath
Computer Science Department
Columbia University
UNI: ak4808

Surya Akella
Computer Science Department
Columbia University
UNI: sa4084

***Abstract*—Financial markets generate large volumes of data on a daily basis. It is extremely challenging to make sense of the data and come up with relevant strategies for investing in stocks. Especially for data covering large time periods, it is necessary to have an approach that can process this data and provide insights. Through this study, we have tried solving the long term portfolio optimization problem. We preprocess and analyze data initially to find out trends. This is followed by the application of LSTM (Long Short Term Memory) as well as certain regression models where the stock data is turned into a time series data. The performance of these models are compared. And lastly we use mean variance optimization on portfolios to come up with risk and returns for them. LSTM performs the best with RMSE (Root mean squared error) of 0.052 among all machine learning techniques.**

***Index Terms*—Portfolio Optimization, Long Short Term Memory, Financial Markets, Regression.**

I. Introduction

In today's dynamic financial markets, investment portfolio management has become increasingly complex and challenging. The pursuit of maximizing returns while minimizing risk has always been the central goal of investment strategy. However, achieving this objective has become more demanding in an era marked by rapid information flow, market volatility, and intricate interconnections among various financial instruments.

The primary focus of this research paper is to explore the application of machine learning techniques in identifying financial trends for effective investment portfolio management. The paper aims to address the fundamental objective of maximizing returns while minimizing risk by utilizing advanced computational methods that can process vast amounts of data and extract meaningful patterns. Machine learning techniques offer the potential to uncover non-linear relationships and capture complex market dynamics that traditional models may overlook. By identifying patterns and trends that are not immediately apparent, these techniques can generate unique insights and improve the accuracy of forecasting models, leading to more informed investment decisions.

In existing literature, [1] Ghosh et al. employs both Random Forest and LSTM as training methodologies to analyze their effectiveness in forecasting out-of-sample directional movements of constituent stocks of the S&P 500 for intraday trading. [3] Deng et al. provides a comprehensive review of machine learning applications in portfolio management, including risk assessment, asset allocation, and trading strategies. [4] Tran et al. and [5] Singh et al. predict time series data using neural networks. [2] Papaioannou et. al. develops a trading strategy that follows trends to forecast and trade S&P 500 contracts. [6] Feng et. al. explore the use of deep learning algorithms for improvement of stock selection by using financial news and historical data. [7] Basak et al. use random forests and gradient boosted decision trees together along with selected technical indicators to analyze the performance for medium to long-run prediction of stock prices returns. [8] Zhang et al. review traditional and deep learning models for financial data forecasting, examining their performance and limitations. [9] Tsantekidis et al. propose a convolutional neural network-based approach to forecast stock prices using information from the limit order book. [10] Ko et al. investigates the use of deep reinforcement learning algorithms for asset allocation in investment portfolios.

In our study, we preprocess our data to suit the needs of our various methods. The modeling phase involves the implementation of LSTM and several regression models such as Linear regression, Random Forest regression and Gradient Boosting regression. These models give us an idea about the sequential nature of stock data while training on 2012-2021 data and predicting on 2022 data. We use Mean Variance Optimization to incorporate the data from the entire decade by calculating a feature known as 'daily returns' by using the open and closed values of the data. MVO then, helps in recognizing how portfolios perform with respect to their annualized risks and annualized returns.

II. Data

We sourced our data from a Kaggle repository where the stock market data gets updated weekly. We chose the S&P

500 index data for our problem. Each row has attributes Stock, Date, High, Low, Open, Close and Adjusted Close. And the overall dataset has decades worth of data for stocks, for some even from the 1980's. Each stock had a separate .csv file for it. The first step involved processing all the .csv files and creating one big dataframe for all stocks.. Since our goal was to recognize investment strategies for the long term. We made use of data from the start of 2012 to the end of 2021 as the basis for our modeling.We calculated the returns of each of the stocks over a 10 year period and sorted them in a descending order. This gave us a brief idea about the stocks that were performing well. Then we selected stocks of varying returns by choosing every fifth stock from this sorted order. There were two reasons for the same, one that it allowed for the modeling phase to not overfit on the giving data by learning parameters of each stock and second that it saved us a lot of computation resources.

For the modeling phase, the data had to have the form of time series data in order to make the necessary predictions. For our LSTM model, the dataset was subjected to a scaler transformation and then converted into a 3d array of time sequences as the input. The output produced by this model did not have a target variable, rather each entry represented a time sequence,i.e., prediction of all attributes given data of the previous few days.

For our portfolio optimization. We made use of data from the top 50 performing stocks based on overall returns in the 2012-2021 time period which we had calculated earlier. Hence, the sorting in descending order of returns for stocks, initially. We scraped the wikipedia page for the S&P 500 stocks to obtain sectors (e.g. IT, Healthcare, etc.) for each stock. Based on the sectors we divided our top 50 stocks into 5 different portfolios. We calculated daily returns for each row using the open and close values and this served as our data for portfolio optimization as the solution for this only concerns itself with daily returns instead of other attributes.

III. Methods

For the modeling phase, we implemented the LSTM model along with certain regression models such as Linear Regression, Random Forest Regression and Gradient Boosting Regression. For portfolio optimization we made use of MVO (Mean Variance Optimization).

A. LSTM Model

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture that is particularly effective in handling sequential data such as time series. LSTM networks are designed to overcome the problem of vanishing gradients, which can occur when training traditional RNNs on long data sequences.

The basic LSTM architecture includes three gates: the input gate, the output gate, and the forget gate. Each of these gates contains a sigmoid activation function that controls the flow of information through the network. In addition, there is a memory cell that stores the temporal dependencies of the sequence. The input gate determines which information from the current input should be passed on to the memory cell. The forget gate decides which information from the previous memory state should be discarded. The output gate determines which information from the current memory state should be passed on to the output. The LSTM architecture also includes a cell state, which allows the network to remember long-term dependencies.

The reason we chose LSTM for time series forecasting of stock market data is for its ability to capture long-term dependencies and patterns in the data. Stock market data is notoriously noisy and volatile, with complex patterns and dependencies that can be difficult to model using traditional statistical techniques especially considering how voluminous the data is. LSTM networks effectively learn and model these complex patterns, providing accurate forecasts that can help investors make informed decisions. Additionally, LSTM networks can handle missing data and irregularly spaced time series data, making them flexible while forecasting data. Our implementation of LSTM involved converting the dataset into a 3rd array of time sequences of 9 dimensions (including day, month and year) with every input entry having 10 time sequences. Meaning that for every 10 days worth of data, all 9 attributes were predicted by the model for the 11th day. Adam optimizer was used with the loss function being mean squared error. LSTM had a RMSE of 0.052 and a Mean Bias error of -0.005.

B. Regression Models

- Linear Regression: Linear Regression is a very interpretable method. The coefficients in the linear regression model can be interpreted as the strength and direction of the relationship between the independent variables (such as past stock prices or other economic indicators) and the dependent variable (future stock prices). Hence, helping in figuring out the underlying factors of the data and allowing for informed decision making.
- Random Forest Regression: It is beneficial for our task for a variety of reasons. It handles non linear relationships between variables. This is important for stock market forecasting, as stock prices can be influenced by a wide range of factors, many of which may not have a simple linear relationship with future stock prices. In addition, it is robust to outliers. Stock market data is often noisy and can contain outliers, which can have a significant impact on the accuracy of the forecast. Random forest regression is robust to outliers, as it is based on an ensemble of decision trees that are less sensitive to outliers than a single decision tree. Lastly, stock market data can contain a large number of variables that may be relevant for forecasting future stock prices. Random forest regression is capable of handling high-dimensional data, as it uses a subset of variables at each split in the decision tree. This makes it computationally efficient, even when dealing with large datasets.
- Gradient Boosting Regression: In addition to the advantages of Random Forest, Gradient Boosting Regression

combines an ensemble of weak learners to create a strong learner. This makes it less prone to overfitting than other machine learning algorithms and can result in more accurate predictions.

Apart from the above methods, we implemented Support Vector Regression as well as MLP regression. Both of these performed very poorly. Hence, we dropped those methods from our implementation. The performances of regression models are compared in Table 1. There were plans to use a hybrid random forest model with LSTM but the LSTM's individual performance was remarkable. Hence, not requiring any further tweaking.

Model	MAPE	RMSE	MBE	R^2
Linear Regression	0.7115	1.4288	0.0123	0.9998
Random Forest Regressor	1.4980	15.9753	-4.0061	0.9810
Gradient Boosting Regressor	2.6024	16.7557	-3.4711	0.9791

TABLE I
PERFORMANCE METRICS OF DIFFERENT REGRESSION MODELS

C. Mean Variance Optimization

The goal for any person to effectively invest is to maximize the returns obtained from the investment in a portfolio at a minimum risk. The Mean-Variance Optimization (MVO) model aims to solve this multi-objective optimization problem subject to basic constraints imposed on the portfolio.

Given a set of n assets with returns $r_1, r_2, \ldots, r_n$, and a vector of portfolio weights $w = (w_1, w_2, \ldots, w_n)$ where w_i represents the proportion of wealth invested in asset i, the mean-variance optimization problem is formulated through the following steps:

Step 1:

$$\max_{w} \quad w^T \Sigma w_i r_i \text{ s.t.} \quad \& \sum_{i=1}^{n} w_i = 1, \ \& w_i \geq 0 \quad \forall i, \qquad (1)$$

The optimal weights ${w_i}^{Optimal}$ obtained from Eq (1) help in solving

$$R^{MaxRet} = \Sigma {w_i}^{Optimal} r_i \qquad (2)$$

where R^{MaxRet} is maximal expected return subject to basic constraints.

Step 2:

$$\min \quad \sqrt{\sum_i \sum_j w_i r_i \sigma_{ij}} \text{ s.t.} \quad \sum_{i=1}^{n} w_i = 1, \ \& w_i \geq 0 \quad \forall i, \qquad (3)$$

where σ_{ij} is the variance-covariance matrix of returns.

The optimal weights ${w_i}^{Optimal}$ obtained from Eq (3) help in solving

$$R^{MinRisk} = \Sigma {w_i}^{Optimal} r_i \qquad (4)$$

where $R^{MinRisk}$ corresponds to the minimum risk portfolio subject to basic constraints.

Step 3:

We increment R by some value (0.001 in our implementation) in every step such that $R^{MinRisk} \leq \text{R} \leq R^{MaxRet}$ subject to the constraints. Each value of R corresponds to a scenario of a portfolio. And all of these portfolio scenarios combine to form an efficient set. The efficient sets for each portfolio are plotted later in this paper.

IV. EXPERIMENTS

A. Modeling

For performance of LSTM we made use of 2022 data which was not used for the model train and test phase. This allowed us to challenge LSTM with data whose underlying trends might not be known to it. Using the stock data from 2022 we made predictions and plotted graphs comparing the actual values v/s the predicted values to showcase the fact that the LSTM model captures the underlying trends of the data. Made more sense than just going for accuracy parameters which is the general trend. The graphs are shown in Fig (1)-(3).

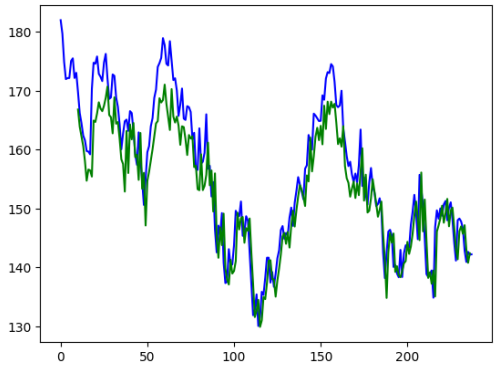

Fig. 1. Apple Stock: Predicted v/s Actual

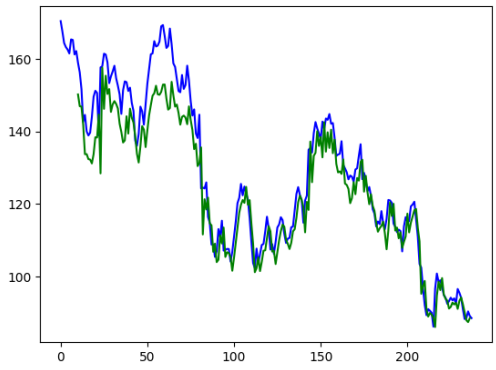

Fig. 2. Amazon Stock: Predicted v/s Actual

B. Mean Variance Optimization

Before solving the optimization problem, it was necessary to decide on the structure of data. Since we were interested in long term returns, we looked at stocks that had performed very well in a 10 year time period. We divided the portfolio sector wise, with some portfolios having stocks from multiple sectors as well. We decided on this approach as it was important for us to ascertain the trends not just for each stock but also the

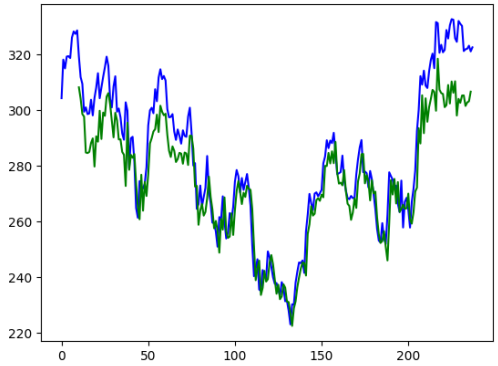

Fig. 3. Ameriprise Financial Stock: Predicted v/s Actual

kind of stock (sector) that would give us an overview of its own performance with a backdrop that brings with it more context. Based on Rmaxret and Rminrisk values, we varied our R between the two values, with each value representing a scenario of the given portfolio. All the possible portfolios combined are known as the efficient set. For each scenario in the efficient set we obtained the weights and its subsequent annualized risk and return percentages. Health Care portfolio (P3) tends to perform better in high risk scenarios. P1 which is Information Technology 1 performs the best in minimum risk scenarios. P2 performs very well for low risk but tends to perform the worst for high risk. It maintains its returns in the 28-30% range irrespective of the risk associated. The plots are shown in Fig(4)-(8).

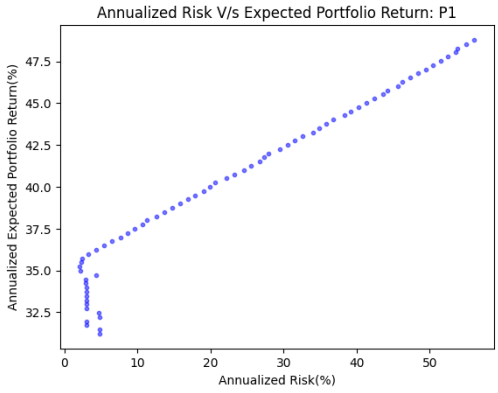

Fig. 4. Portfolio 1: IT 1

V. System Overview

We have a ReactJs frontend dashboard that gives an overview of our implementation through visualizations. The main dataset containing data of every stock is stored in SQLite database. The machine learning models have been trained on truncated datasets to avoid overfitting and high computation times and subsequently stored. For example, the LSTM model (LSTM_80.h5 file) is trained on 80 stocks from the S&P 500 over 10 years worth of data. Similarly we have a regression model as well, saved in the form of a pickle file that takes

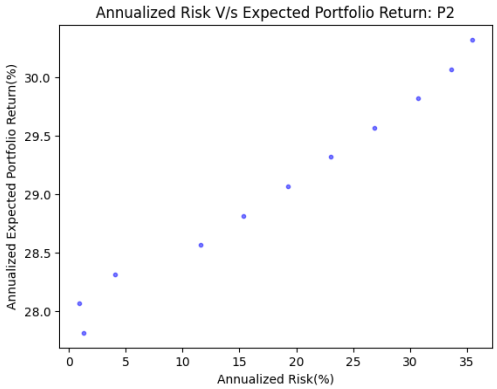

Fig. 5. Portfolio 2: IT 2

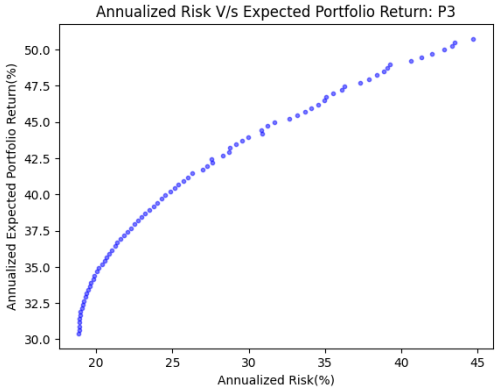

Fig. 6. Portfolio 3: Health Care

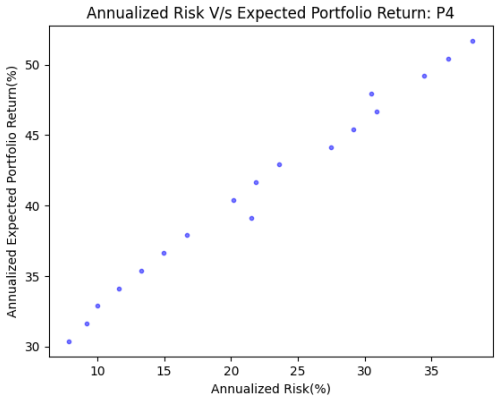

Fig. 7. Portfolio 4: Consumer Discretionary + Others

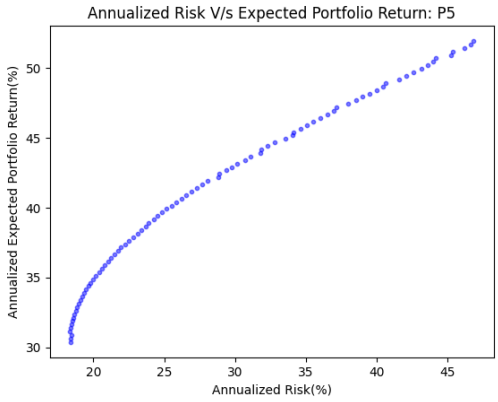

Fig. 8. Portfolio 5: Coomunication Services + Industrials + Financials

in values of all attributes in order to predict the close value, which is its target variable. This varies significantly from the LSTM model, which doesn't have a target value but its output is an array of time sequences with each entry representing a prediction of a group of time sequences preceding it. The plot for the predicted values by LSTM v/s actual values is shown to the user. To better visualize stock data we also implement a function where the user enters the name of the stock, the variable to be plotted and the start and end date, and the system returns a plot of that stock for that given variable in the time period ranging from the start to the end date. The return % is also calculated and provided to give the user an idea of how the stock changed over time. The user can also place the pointer on the graph to look for the values on specific dates. The SQLite database is queried in order to provide the necessary information for the plot. Lastly, we have uploaded our annualized risk v/s returns plots as well on the dashboard, in order to give a visualization to the user regarding the change in returns of various portfolios while varying the risk. The portfolio (risk-return) plots are stored on a cloud service called imgbb which coordinates with the reactjs frontend to display them when necessary. The Django REST framework is used to build all these APIs and communicate with the frontend for displaying the output for these operations. The architecture diagram is shown in Fig 9.

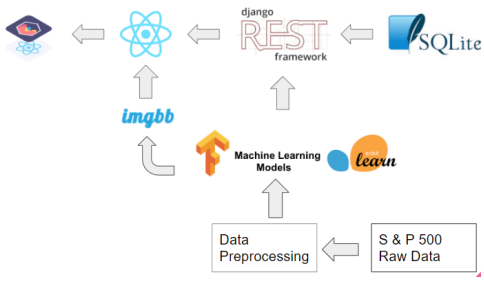

Fig. 9. Web Application Architecture

VI. Conclusion

LSTM outperforms the regression models with an RMSE score of 0.052. It has a negative mean bias error which means that it underestimates the values a bit during prediction which can be seen in Fig(1)-(3). Portfolio Optimization graphs show that different portfolios perform better than others depending on the annualized risk % an investor is ready to face.Based on the risk appetite, one can decide the portfolio and the weights they'll allot to the stocks in the said portfolio. Weights and their related annualized risks and returns are output in the python notebook in our github repository At the moment we used 2012-2021 data for modeling and tested it on 2022 data. For future work we could try to forecast values of future dates using our ML algorithms. At the same time we could incorporate news data to better understand the underlying trends of variations in stock prices.

References

[1] Pushpendu Ghosh, Ariel Neufeld, Jajati Keshari Sahoo, Forecasting directional movements of stock prices for intraday trading using LSTM and random forests,Finance Research Letters,Volume 46, Part A,2022,102280, ISSN 1544-6123,https://doi.org/10.1016/j.frl.2021.102280.

[2] Panagiotis Papaioannou, Thomas Dionysopoulos, Lucia Russo, Francesco Giannino, Dietmar Janetzko, Constantinos Siettos, S&P500 Forecasting and trading using convolution analysis of major asset classes,Procedia Computer Science,Volume 113,2017,Pages 484-489,ISSN 1877-0509, https://doi.org/10.1016/j.procs.2017.08.307.

[3] Deng, Z., Gu, B., Zhang, D., & Ji, W. (2021). Machine learning applications in portfolio management: A comprehensive review. European Journal of Operational Research, 295(2), 377-396.

[4] Tran, D. T., Iosifidis, A., Kanniainen, J., & Gabbouj, M. (2018). Temporal attention-augmented bilinear network for financial time-series data analysis. IEEE transactions on neural networks and learning systems, 30, 1407–1418.

[5] Singh, K., Tiwari, R., Johri, P, & Elngar, A. (2020). Feature Selection and Hyper-parameter Tuning Technique using Neural Network for Stock Market Prediction. Journal of Information Technology Management, 12, 89–108.

[6] Feng, T., Liu, C., & Xu, W. (2018). Deep learning for stock selection based on financial news and historical data. Knowledge-Based Systems, 161, 206-216.

[7] Basak, S., Kar, S., Saha, S., Khaidem, L., & Dey, S. R. (2019). Predicting the direction of stock market prices using tree-based classifiers. The North American Journal of Economics and Finance, 47, 552–567.

[8] Zhang, Y., & Zhou, Y. (2021). A review on financial data forecasting models: Traditional and deep learning models. Journal of Economic Surveys, 35(2), 398-425.

[9] Tsantekidis, A., Passalis, N., Tefas, A., & Kanniainen, J. (2020). Forecasting stock prices from the limit order book using convolutional neural networks.

[10] Ko, M., Jeong, Y., & Yoon, S. M. (2020). Deep reinforcement learning for asset allocation.