Final Report - Financial Forecasting using Quarterly Filings

Kechengjie Zhu
kz2407@columbia.edu

Xuchen Wang
xw2747@columbia.edu

Yao Xiao
yx2696@columbia.edu

Zhenyu Yuan
zy2492@columbia.edu

Zixiang Yin
zy2444@columbia.edu

Abstract

In recent years, artificial intelligence has been widely used to predict stock prices, but limited attention was paid to companies' financial status which is one of the most influencing factors against stock price. Creating a high-quality performance forecast would definitely aid market participants such as investors to make better trading decisions and manage their portfolios more suitably while outperforming the market. For this reason, our research focused on forecasting companies' financial performance using quarterly released 10-K/10-Q filings including Balance Sheet, Income Statement and Cash Flow as well macroeconomic indicators such as GDP, CPI, unemployment rate, etc. We used company-wise ARIMA as the baseline model and built LSTM/Transformer to see if there is a performance improvement. According to experiments on the Nasdaq Composite components, LSTM has the closest accuracy compared with company-wise ARIMA models in terms of SMAPE.

1 Introduction

In the world of finance, predicting stock market performance is a crucial task that has significant commercial value. One important factor to consider when predicting stock market performance is a company's financial fundamentals, such as revenue, net income, and debt. These fundamentals provide valuable insights into a company's financial health and performance and can significantly impact its stock price. By accurately forecasting financial fundamentals, investors and analysts can make more informed decisions about the potential risks and rewards of different investment opportunities.

With a business goal of aiding market participants such as investors to make better trading decisions and manage their portfolios more suitably while outperforming the market, our project, supported and supervised by Simran Lamba from JPMorgan, focused on forecasting companies' financial performance using 1) quarterly released 10-K/10-Q filings including Balance Sheet, Income Statement and Cash Flow; and 2) macroeconomic indicators such as GDP, CPI, unemployment rate, etc.

Specifically, our project falls under the category of multivariate multi-target time series forecasting task. In other words, we are predicting multiple selected features for the next time step based on their historical values from previous time steps.

Preprint. Under review.

2 Dataset and Exploratory Data Analysis

2.1 Data Collection

As mentioned previously, we leveraged financial fundamentals and macroeconomic indicators to build our models. The former was sourced from a public stock market data API called EOD Historical Data (EODHD). For a given list of stock symbols, it provides financial fundamentals in various temporal dimensions covering detailed terms in three financial statements (Balance Sheet, Income Statement, and Cash Flow Statement). We have therefore pulled the quarterly fundamentals of 2,321 Nasdaq Composite components for at least 5 years quarterly fundamental data for all components of the NASDAQ Composite from the API and ended up with a dataset of 171,560 entries as well as 128 features. The data collected was originally in JSON format, and we parsed it to a large CSV using a preprocessing Python script.

2.2 Data Ethics

It is worth clarifying that our data are the financial statements of listed enterprises in the U.S. stock market, which are required to be disclosed by policies and laws. For macroeconomic data such as GDP, CPI, etc, the same argument holds as well. Therefore, the collected dataset does not involve any sensitive information, and we do not have any problems related to data privacy and other ethical concerns.

2.3 Handling Missing Values

During the data collection, we discovered that missing values were prevalent in most features as shown in the three bar charts below. Considering the fact that important features in financial statements generally do not contain lots of missing values regardless of the size of companies, we empirically set a cut-off point of 5% to filter out "bad" features whose missing values percentage is greater than this threshold.

After the preprocessing above, we filtered out about 85% of the features and ended up with 17 "good" ones from three financial statements. Then, we decided to drop all companies that contain NAs in the remaining features from the dataset directly since we did not have a good strategy to fill them, especially in the context of the time series forecasting task. Eventually, we got an NA-free dataset of 16,368 data entries and 17 features.

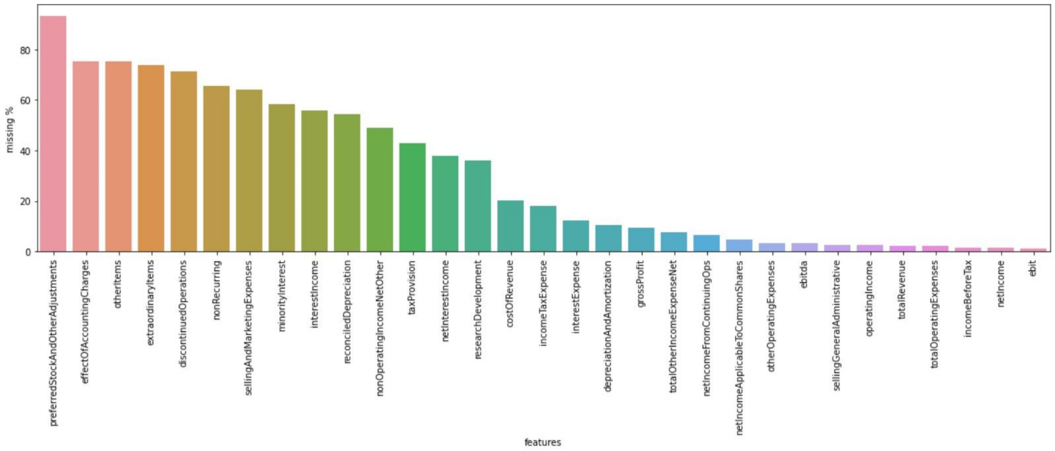

Figure 1: Percentage of missing values for Income Statement

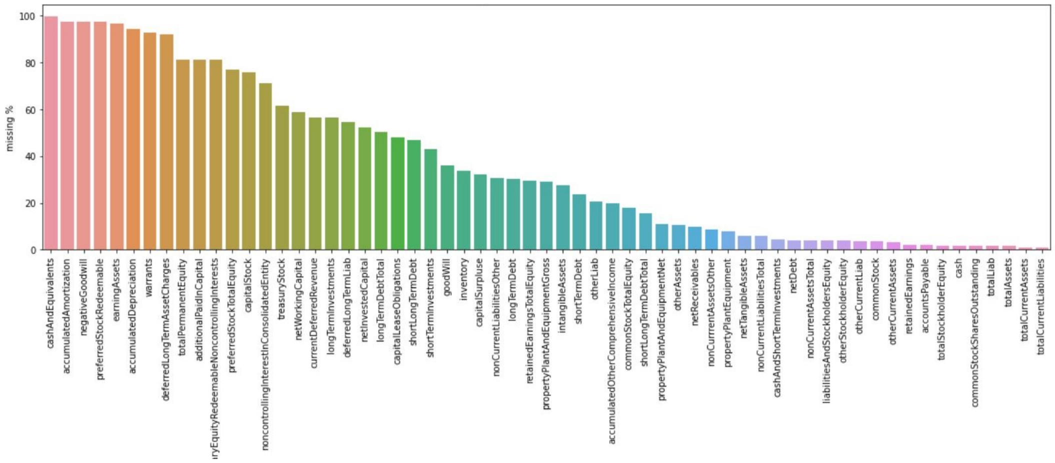

Figure 2: Percentage of missing values for Balance Sheet

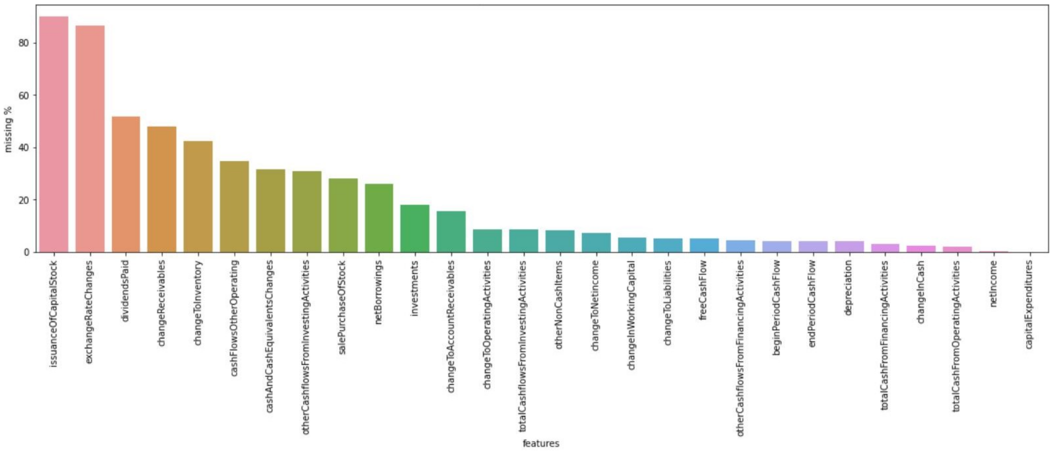

Figure 3: Percentage of missing values for Cash Flow

2.4 Feature Selection

2.4.1 Macroeconomic Features

For macroeconomic features, we selected five indicators of U.S. that are relevant to large companies' performance, and they are quarterly: 1) Gross Domestic Product; 2) Unemployment rate; 3) Consumer Price Index; 4) Producer Price Index; 5) Industrial Output.

2.4.2 Financial Features

Since financial fundamentals in statements are inherently highly correlated, we selectively deleted features by drawing a heatmap shown below and checking pairwise correlation. Finally, we got 7 types of fundamentals to be financial features as well as targets for our time series forecasting models:

- Balance Sheet - total Liability (*LIAB*), total Stockholder Equity (*EQUITY*), common Stock Shares Outstanding (*SSO*)
- Income Statement - Earnings Before Interest and Taxes (*EBIT*), Operating Expenses (*OE*),
- Cash Flow - other Cashflows From Financing Activities (*CFFA*), Capital Expenditures (*CE*)

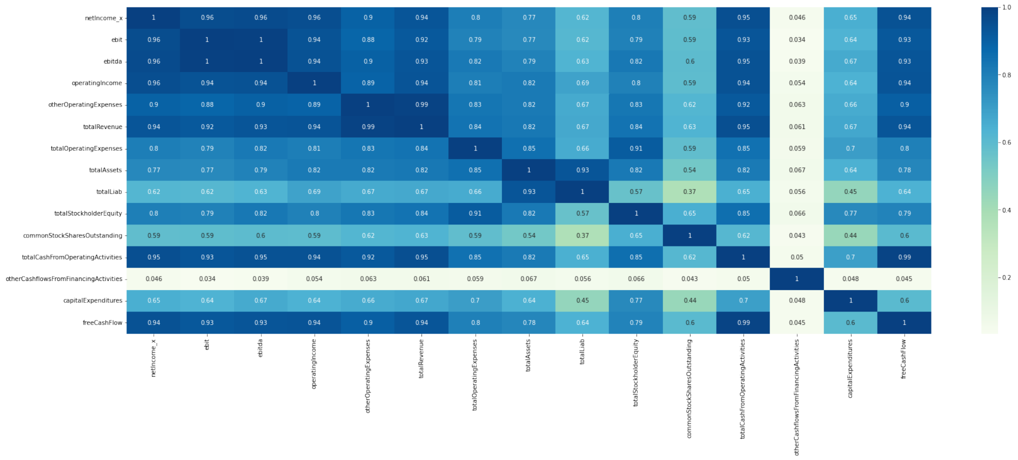

Figure 4: Correlation Heatmap

2.5 Seasonality

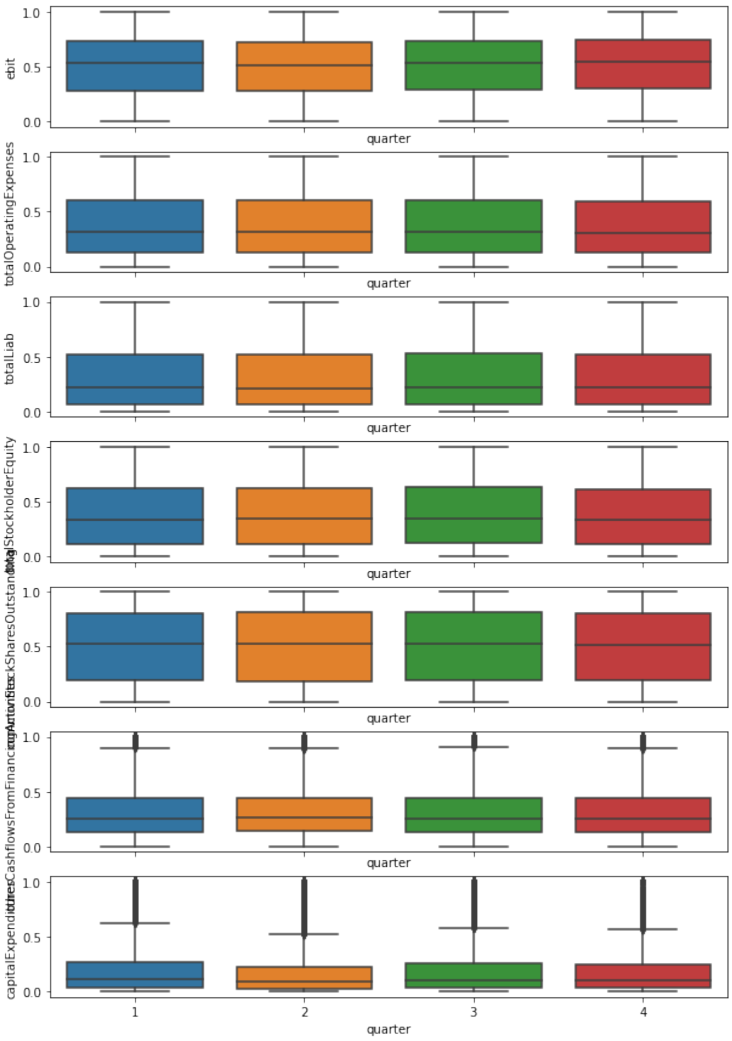

Figure 5: Seasonality Check

Since we are doing a time series analysis, we need to check if there is any seasonality in our data. According to the boxplots above on 7 targets, there is no significant seasonality.

3 Related Work

Predicting the financial fundamentals of companies is an important task for various stakeholders such as investors, analysts, and credit rating agencies. Many methods have been proposed for this purpose, including traditional statistical techniques and machine learning methods.

One of the earliest and most commonly used regression techniques is linear regression, which models the relationship between the dependent variable and independent variables as a linear function. Linear regression has been widely applied in the financial domain, particularly in the prediction of stock returns and financial ratios. However, it has some limitations, such as the assumption of linearity and the inability to capture non-linear relationships. To address these limitations, researchers have proposed various non-linear regression techniques such as polynomial regression, decision tree regression, and support vector regression.

Polynomial regression can capture non-linear relationships by fitting a polynomial function to the data, while decision tree regression builds a tree-like model to make predictions based on the value of the independent variables. Support vector regression is a type of machine learning technique that uses support vectors to define a hyperplane that maximizes the margin between the dependent and independent variables.

Recent studies have also explored the use of more advanced machine learning techniques for predicting financial fundamentals. For example, some researchers have used deep learning techniques such as artificial neural networks and convolutional neural networks to predict financial ratios and stock returns. These techniques have the advantage of being able to capture complex patterns in the data, but they also require large amounts of data and computational resources to train.

Overall, regression techniques have been widely applied in the prediction of financial fundamentals, and have demonstrated promising results in terms of accuracy and reliability. However, these techniques treat each data entry independently and ignore the potential interrelation among data in continuous time steps. For this reason, we decided to take a different methodology and approach the question from the perspective of time series analysis.

4 Methods

Mathematically, we aimed at building a multivariate multi-target model f_θ that takes historical time series data $\mathbf{X}$ and an integer k as input, and output Y_{t+1} where k is the number of time steps looked back:

$$\mathbf{X} = \big((Y_{t-k+1}, X_{t-k+1}), (Y_{t-k+2}, X_{t-k+2}), \dots, (Y_t, X_t)\big)^T \in R^{k\times 12}$$

$$Y_i \in R^7 \quad \text{and} \quad X_i \in R^5$$

Before training our models, we did a min-max scaling on our features company by company. This is because companies' size varies from one to another, and they are comparable only after scaling. In order to evaluate the performance of our models after training, we used **Symmetric Mean Absolute Percentage Error (SMAPE)** as the metric where N is the total number of samples in the transformed dataset:

$$\textbf{SMAPE} = \frac{200\%}{N}\sum_{i=1}^{N}\frac{|f_\theta(\mathbf{X}_i) - Y_i|}{|f_\theta(\mathbf{X}_i)| + |Y_i|}$$

4.1 ARIMA baseline

For the baseline model, we decided to use ARIMA, a classic statistical model that can be used to analyze a wide range of time series data, including data with trends, seasonal patterns, and noise, and is relatively robust to violations of the assumptions underlying the model, such as non-constant variance or autocorrelation.

Considering the difference in companies' financial status, we built company-specific ARIMA models for all 157 companies in our dataset so that the model can take into account factors that are specific to the company being modeled, such as the company's industry, management team, and financial performance.

We implemented company-specific ARIMA models using the package pmdarima in Python, which could automatically help us search for the optimal hyperparameters p, d, and q, which controlled the number of lags, the order of difference, and the number of lagged forecast error in the model respectively.

4.2 LSTM

Recurrent neural networks (RNNs) and long short-term memory (LSTM) models are neural networks designed for processing sequential data. LSTMs, a type of RNN, have additional memory cells and gates that allow them to better retain information from long sequences and are particularly well-suited for predicting financial fundamentals because they can handle long-term dependencies, such as trends or seasonal patterns, which can be difficult for traditional models to capture. In addition, LSTMs are relatively robust to noise, making them effective at predicting financial fundamentals in the presence of noise. Financial data can often be noisy, with fluctuations that may not be indicative of underlying trends or patterns.

A general architecture of RNNs and LSTM model as well as its four types of gates are elaborated below:

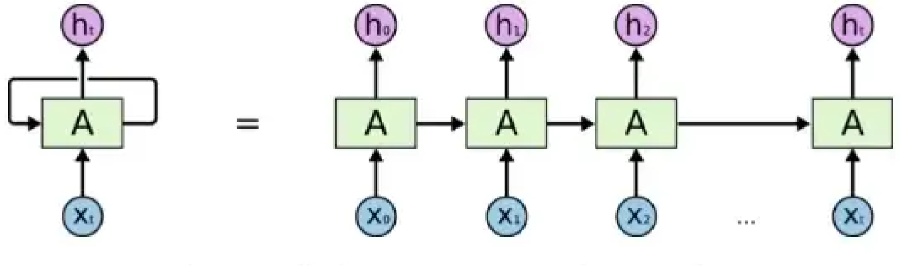

Figure 6: A general RNN architecture

Forget Gate:

$$f_t = \sigma\big(W_f \cdot [h_{t-1}, x_t] + b_f\big)$$

Input Gate:

$$i_t = \sigma\big(W_i \cdot [h_{t-1}, x_t] + b_i\big), \qquad \tilde{C}_t = \tanh\big(W_C \cdot [h_{t-1}, x_t] + b_C\big)$$

Update Gate:

$$C_t = f_t \times C_{t-1} + i_{t-1} \times \tilde{C}_t$$

Output Gate:

$$o_t = \sigma\big(W_o \cdot [h_{t-1}, x_t] + o_i\big), \qquad h_t = o_t \times \tanh\big(C_t\big)$$

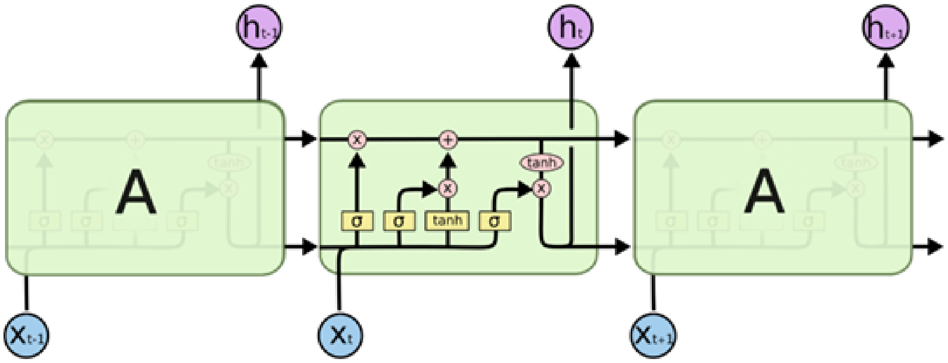

Figure 7: A general LSTM architecture

Through extensive experimentation, we finally implemented the LSTM-based architecture model shown below for our forecasting task. The asterisk sign means that the units in LSTM are a hyperparameter to tune during model selection.

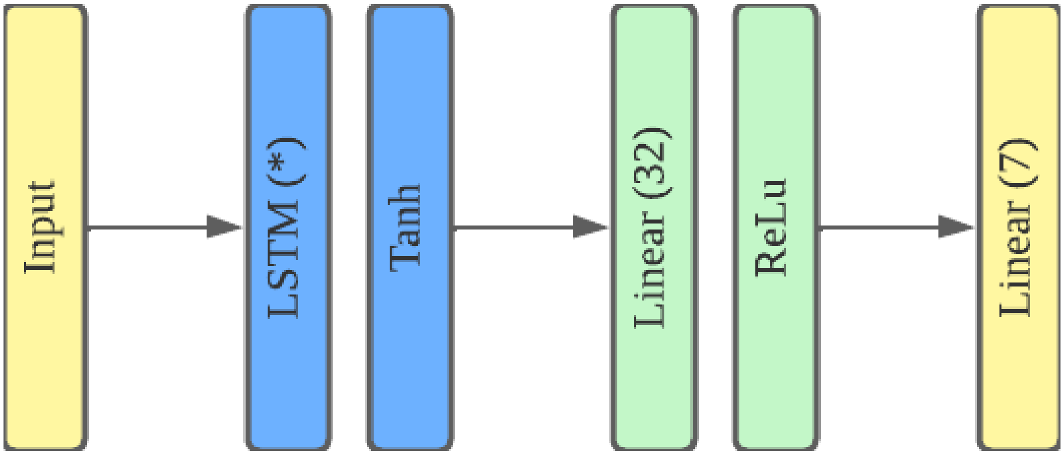

Figure 8: Implemented LSTM-based model

4.3 Transformer

The transformer is a neural network that consists of multiple layers of self-attention and feedforward layers. The self-attention layers allow the model to weigh the importance of different input features when making predictions, which helps the transformer perform well for tasks that require the model to consider the context or relationships between different input features. The feedforward layers consist of multiple fully connected layers that process the input data and generate output predictions shown below.

One key feature of the transformer is that it does not use traditional recurrence or convolutional structures, which are commonly used in other neural network architectures. Instead, the transformer relies solely on self-attention mechanisms to process the input data. This makes the transformer a particular fit for our time series task that requires the model to consider long sequences of financial fundamentals.

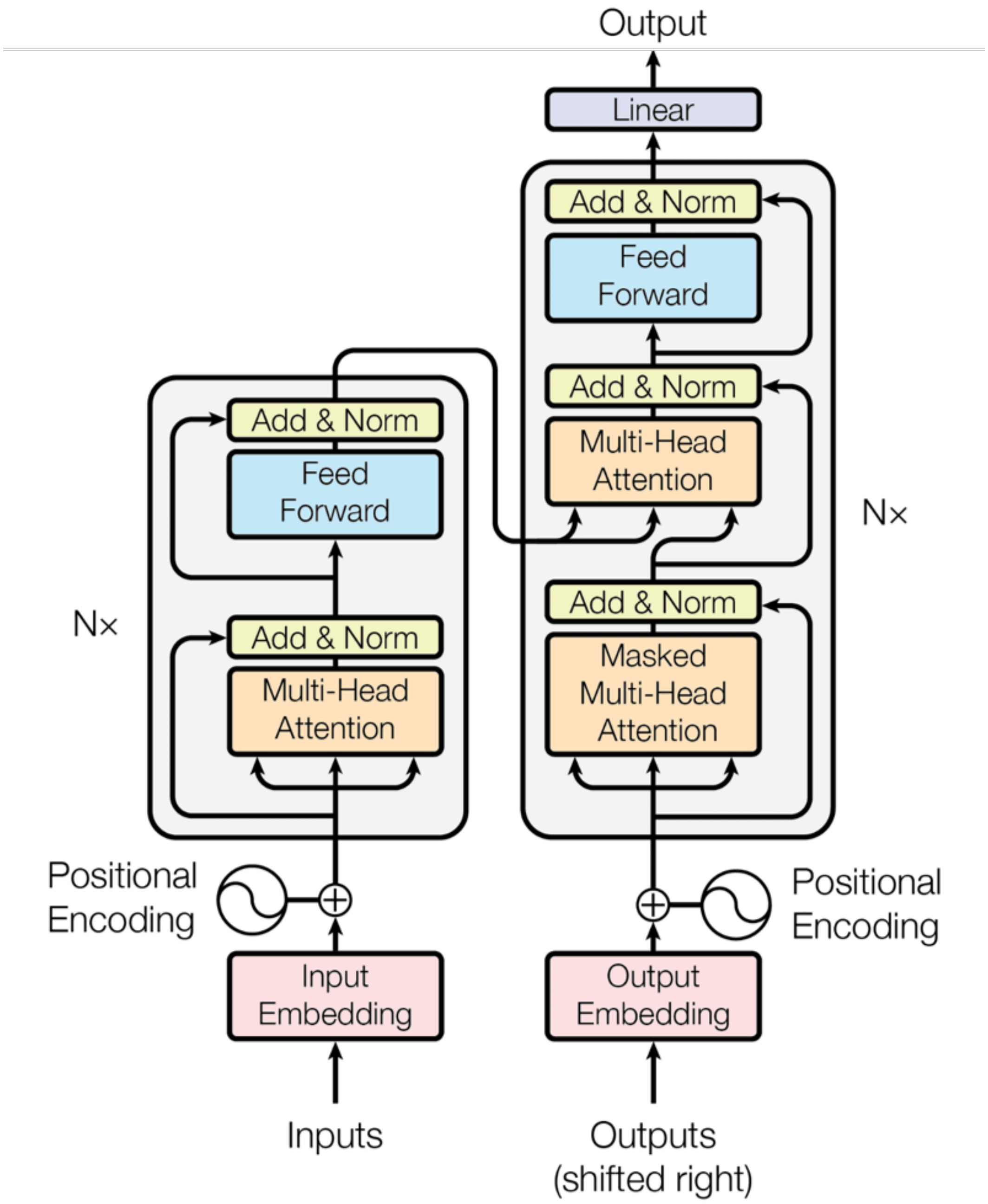

Figure 9: A general Transformer architecture

In fact, Transformers are first proposed and then widely used in Natural Language Processing relevant takes but not for time series forecasting. For this reason, we did a slight change to fit our task based on the general architecture of the Transformer above. In our implementation, the transformer has the following key settings: 1) a dropout rate of 0.1; 2) 10 heads multi-head attention layer; 3) 6 encoder blocks and 6 decoder blocks. Generally speaking, the transformer suffers from the problem of overfitting due to its complex architecture. However, we discovered the model did not overfit our dataset, which justifies the use of a relatively small rate of dropout.

5 Results

In this section, we talked about the procedure for tuning the hyperparameters for three types of models individually. In the end, we compare their performance by calculating the SMAPE metric evaluated on the hold-out test data.

5.1 ARIMA

As mentioned in the previous section, there are three hyperparameters (p, d, q) controlling the number of lags terms, the order of difference, and the number of lagged forecast error terms respectively in ARIMA that need pre-defined. However, the package we used helped us automatically decide the optimal values for the three hyperparameters based on the Bayesian Information Criterion. Therefore, there is actually no manual model selection for our ARIMA models.

5.2 LSTM

For LSTM-based models, we tuned two hyperparameters, i.e. the number of LSTM units and the number of steps looked back, using a grid search strategy. We limited our attention to those hyperparameters since they are empirically important. The optimal model is decided based on the Mean Squared Loss (MSE) on the validation set.

We first built our LSTM-based models using financial features only to predict their values in the future. The result is presented as a heatmap below. It is clear that the optimal model is built on a *look_back* of 14 quarters and 76 LSTM units.

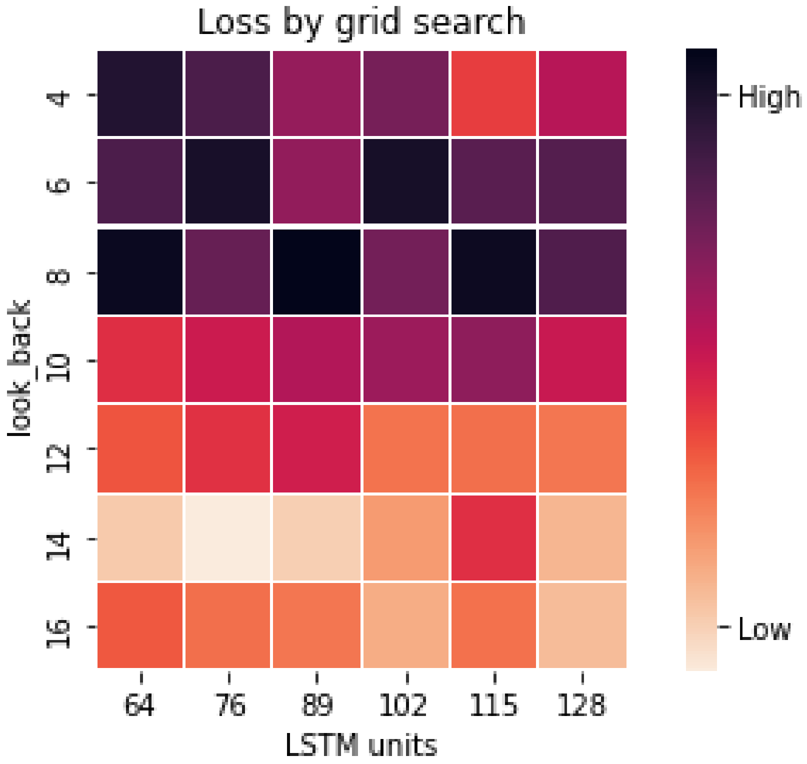

Figure 10: Gridsearch for LSTM with financial features only

For LSTM-based models built on financial and macroeconomic features, we drew the same heatmap shown below. This time the optimal model is achieved with a *look_back* of 16 quarters and 115 LSTM units.

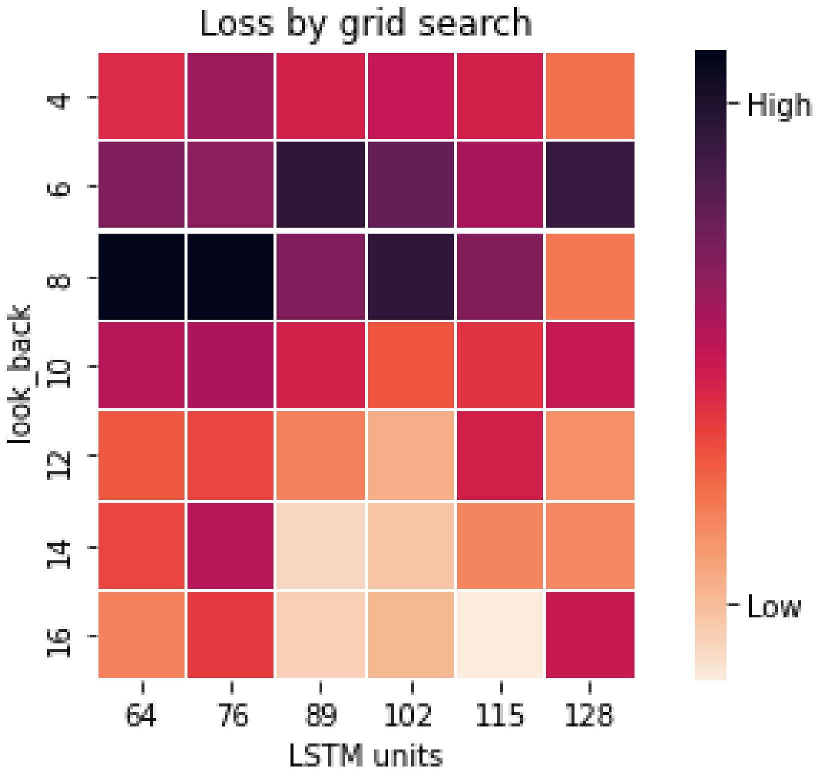

Figure 11: Gridsearch for LSTM with financial and macroeconomic features

5.3 Transformer

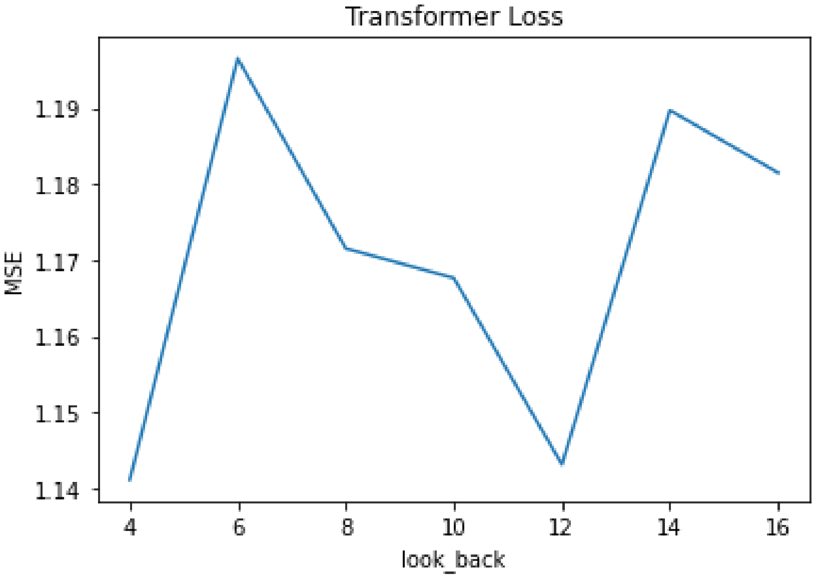

Figure 12: Gridsearch for Transformer with financial features only

Due to limited computing resources, we only tune the number of steps looked back for our Transformer model. Same to the model selection process, we used MSE loss as the criterion to decide the best model.

Based on the plot above, the optimal model built on financial features only is achieved with a *look_back* of 4 quarters.

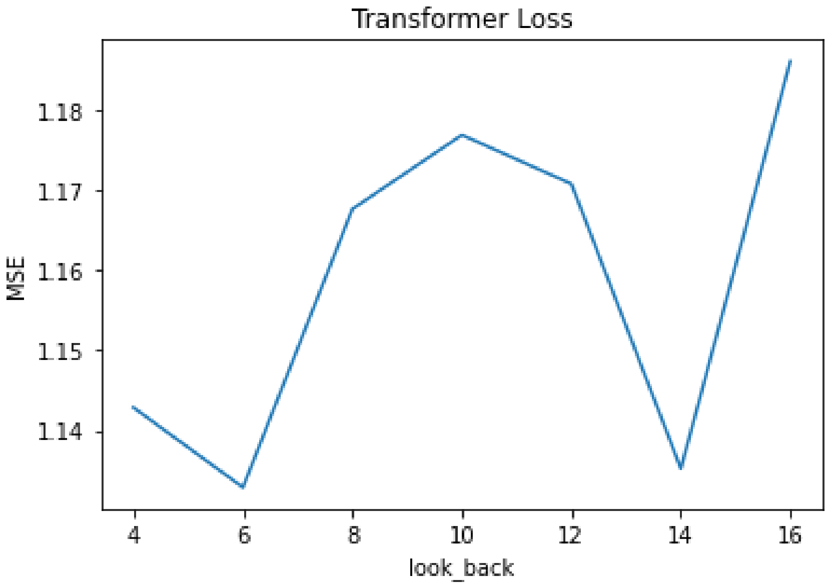

Figure 13: Gridsearch for Transformer with financial and macroeconomic features

However, for the transformer trained on both financial and macroeconomic features, the best model was built with a *look_back* of 6 quarters.

5.4 Comparison

Based on optimal models' prediction on the hold-out test set, we summarized their performance in the following table using SMAPE as the evaluation metric.

	EBIT	OE	LIAB	EQUITY	SSO	CFFA	CE
ARIMA	23.57	19.90	8.83	9.16	20.18	34.36	44.83
LSTM	23.42	19.99	8.68	11.53	21.20	37.91	48.72
LSTM with macro	23.51	22.23	10.92	10.83	21.14	43.23	47.06
Transformer	40.01	44.56	24.95	28.53	43.97	58.85	80.80
Transformer with macro	38.88	45.26	20.59	27.99	42.33	65.42	72.37

Figure 14: Model performance comparison

According to the table above, we have the following key observations:

- The company-specific ARIMA models almost dominated the result except for features *EBIT* and *LIAB*. This is reasonable to see this result since all other models did not take into account the factor of companies' differences.
- Despite the dominance of ARIMA models, the gap between ARIMA and LSTM-based models is quite limited, meaning that the systematic LSTM models could work as well as the company-specific ARIMA models.
- Regardless of LSTM or Transformer, the macroeconomic features did not play an active role in performance improvement for most targets.

- Given the result that Transformers perform among all models as well as the small *look_back* of two optimal deep learning models, we infer that there is no long-term dependence in financial fundamentals.

6 Software

All methods, models, and visualization were implemented in the Google Colab notebook environment. Packaged involved to build ARIMA, LSTM, and Transform are available on GitHub open-source repository. Since we wrote code in the Colab notebook, each code segmentation is clearly named and labeled. Included sections are: Preprocessing, Exploratory Analysis, Utils functions, ARIMA - Baseline, Multivariate LSTM, and Transformer. All the code for reproducing our project is available in the provided GitHub link here. Considering that the dataset is pretty large, we didn't include it in our repository.

7 Conclusion

In conclusion, we focused on solving the task of financial fundamentals forecasting from the perspective of time series. Three models were built mainly on financial features with/without macroeconomic indicators. By comparison, company-specific ARIMA models dominate the result most, but there is a limited gap between the performance of company-specific ARIMA models and that of the systematic LSTM-based model. An insight was concluded from our project that there is no long-term dependency within companies' financial fundamentals.

8 Future Work

There are a few limitations of our project that we should work on in the future to improve the model performance. First, If more computing resources are available, we should try to tune more hyperparameters as for now only two and one of them were tuned for LSTMs and Transformers respectively. Secondly, we should make extensive changes to the basic transformers given their bad performance on our dataset. Thirdly, we dropped almost 90% data during cleaning due to the existence of missing values. If we can figure out a way to appropriately fill them, our dataset size would grow greatly, making it possible for us to improve the model performance further.

9 Contribution

Kechengjie Zhu (kz2407): calculated feature correlation. selected features.

Xuchen Wang (xw2747): drew correlation heatmap. selected features.

Yao Xiao (yx2696): looked for evaluation metrics. documented meetings' content. selected features.

Zhenyu Yuan (zy2492): collected data.

Zixiang Yin (zy2444): collected, cleaned, and prepossessed data. literature review. exploratory data. analysis. model training, selection, improvement, and implementation. wrote reports.

References

Antony Papadimitriou, Urjitkumar Patel, Lisa Kim, Grace Bang, Azadeh Nematzadeh, and Xiaomo Liu. 2020. A multi-faceted approach to large scale financial forecasting. In Proceedings of the First ACM International Conference on AI in Finance (ICAIF '20). Association for Computing Machinery, New York, NY, USA, Article 5, 1–8.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2019. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting (2019).

Jeffrey E Jarrett and Eric Kyper. (2011) ARIMA modeling with intervention to forecast and analyze chinese stock prices. International Journal of Engineering Business Management 3, 3 (2011), 53–58.

Peter Whittle. 1951. Hypothesis testing in time series analysis. Vol. 4. Almqvist & Wiksells boktr.

Xiaodong Sun, Dinesh K Gauri, and Scott Webster. 2011. Forecasting for cruise line revenue management. Journal of Revenue and Pricing management 10, 4 (2011), 306–324.