Predict Stock Log Return Using Innovative Deep Learning Models Leveraging Both Financial and Reddit Sentiment Features

Yuling (Max) Chen (Student ID: 21020403)
Huanqiu (Victoria) Wang (Student ID: 20946126)
Min Gyu (Mike) Woo (Student ID: 20665211)
Yaolun (Jason) Yin (Student ID: 20701426)

Instructor: Ali Ghodsi

Department of Statistics and Actuarial Science
The University of Waterloo

Project report submitted for the assessment of **STAT 940**
April 2023

1 Introduction

The prediction of stock log returns has received much attention in finance due to its potential applications in portfolio management, risk management, and trading strategies. In recent years, incorporating sentiment data from social media into stock trend prediction has become increasingly popular. In this project, we applied multiple deep learning models to predict the daily log return of Apple stock utilizing both financial and sentiment features. More specifically, we replicated an established deep learning framework WSAEs-LSTM, proposed by Wei Bao (2017). which employs a thorough set of financial features for prediction. In addition, we integrated sentiment features derived from Reddit comments using both SIA and Bert sentiment models. Our analysis compared the performance of WSAEs-LSTM with two baseline models (W-LSTM and W-RF), considering the inclusion and exclusion of sentiment features. We observed enhanced accuracy in predicting trends when incorporating the sentiment features. Most excitingly, we introduced a novel W-AutoFormer model, which yielded the best results.

In the following sections, we outline the data preprocessing steps in Section 2, including data sources, financial and sentiment feature construction, and rolling evaluation arrangements. In Section 3, we describe the architecture of the models, highlight the motivation of implementing these models. Lastly, we evaluate the models' performances, analyze the effectiveness of sentiment features, and conclude this report with our own insights in Section 4. To sum up the primary innovations of this project:

- The integration of sentiment features with financial time series data in the WSAEs-LSTM pipeline, utilizing Reddit data, which has seldom been analyzed, as opposed to the more popular choices of Twitter, financial news titles, and financial articles.

- From WASEs-LSTM to W-AutoFormer, we made an innovation to the model by replacing a portion of the existing pipeline with AutoFormer, an approach that has scarcely been applied in the financial field.

2 Data Preprocessing

2.1 Financial Feature Construction

We collected financial data for Apple Inc. (AAPL) from 2012 to 2022 using the Python package yfinance, which includes six basic price and volume fields. From these fields, we further computed additional market indicators, resulting in a total of 12 features. The full set of financial features are included in the Supplementary material section. Our target, the next-day log return, is calculated as $r_t = log(\frac{P_t+1}{P_t})$, where P_t is the closing price at time t.

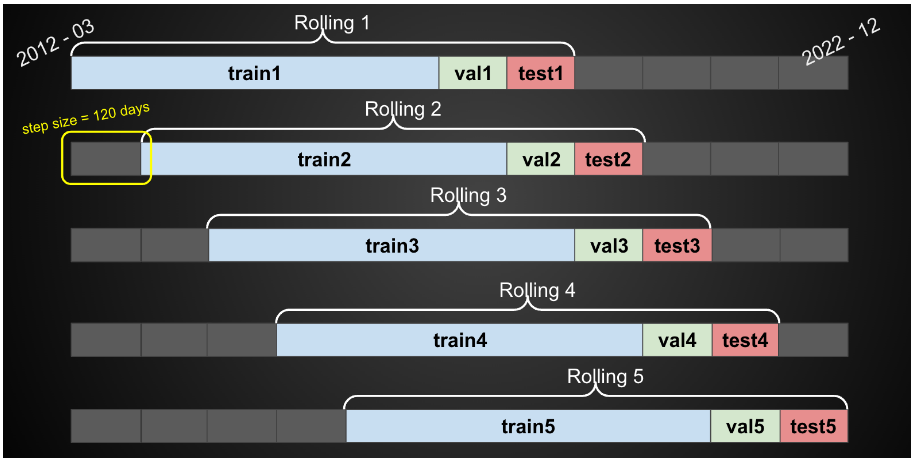
2.2 Sentiment Feature Construction

We scraped textual data from Reddit's r/Apple subreddit using the Pushshift API, which was selected for its ease of use and ability to collect large amounts of data. After that, we collected 200 Reddit posts per day for sentiment feature construction. To derive the sentiments, we leveraged the Sentiment Intensity Analyzer (SIA) tool from the NLTK package and the BERT model. The SIA tool assigned a compound score ranging from -1 to 1 to each post, which we averaged to obtain a single numerical score for each day. The BERT model assigned a label ranging from 1 to 5 and the corresponding probability to each post, which we used to calculate the weighted average score for each day.

In essence, this step generated two daily sentiment scores produced by two distinct models. We combined these scores with the financial features, resulting in a total of 14 features. We utilized two separate sentiment models as we believe each score might capture different aspects of the textual information.

2.3 Rolling Evaluation Methodology

The following figure demonstrates our rolling arrangement for training, validating and testing throughout the whole sample period. Both baseline models and the WSAEs-LSTM model are assessed on all five consecutive testsets. However, the W-AutoFormer is not evaluated using this particular arrangement. Details regarding the arrangement for the W-AutoFormer can be found in Section 3.3.



3 Models

3.1 Baseline Models

First, we assessed the performance of LSTM and Random Forests (RF) as our baseline models for predicting Apple stock's log returns, with and without sentiment data. Figure 6 shows that the inclusion of sentiment data in the LSTM model provides superior results. Conversely, RF generated nearly constant predictions, failing to capture the volatility of the true log return. The performance of the models varies across time periods, but both demonstrate accurate predictions with an acceptable mean squared error of approximately 10^{-3}. While this is a reasonable error level, we expect to see improved performance from the more advanced models in the subsequent sections.

3.2 WSAEs-LSTM

The WSAEs-LSTM model comprises three components. This section highlights the motivation behind each layer and describes the deviations from the original paper. For mathematical equations, please refer to the supplementary materials and the original paper. The flowchart of the architecture is found in Figure 2.

1. **Wavelet Transformation:** WT is employed to denoise the raw inputs. It denoises data because it effectively separates the signal components from the noise components. When applying the wavelet transformation, the data is decomposed into a set of wavelet coefficients, representing different frequency bands. Wei Bao (2017) utilize Haar wavelets for decomposition and subsequently reconstruct the data through a series of projections on the mother and father wavelets.

2. **Stacked Autoencoders:** SAEs aim to learn a deeper, more compressed representation of inputs. The SAEs is essentially 4 single-layer autoencoders (AEs) stacked together. Stacking multiple single encode-decode processes together and training them separately is more effective than training one deep encoder-decoder with multiple hidden layers, as the latter may accumulate randomness and errors across the hidden layers and is more susceptible to overfitting.

Each AE consists of one input layer, one hidden layer, and one reconstruction layer (output layer). We set the hidden layer size in each AE to 8, smaller than the 10 used by Wei Bao (2017), as our feature set is smaller. The loss between the input and reconstructed layers is the summation of three terms (Refer to Wei Bao (2017) for mathematical details):

- Squared Reconstruction Error: measures variation by summing squared differences
- Weight Decay: prevents overfitting
- Sparse Penalty Term: forces the model to extract the most meaningful representation by activating only a small number of neurons in the hidden layer.

It is crucial to note that the weights and biases of the reconstruction layer are discarded after training each single-layer AE. It is the hidden layer's parameters that are passed forward, as they represent the learnt abstracted features. We initialize the SAEs only once during the first rolling. For subsequent rollings, we continue with the SAEs learned in the previous rolling. As a result, the loss of each AE decreases across rollings as shown in Figure 4.

3. **LSTM:** The outputs of the SAEs are then input into LSTM to generate the predicted log return. The LSTM model is configured with 5 hidden layers, each containing 100 neurons and a delay of 4. Unlike the SAEs, the LSTM is trained independently across rollings, meaning it starts from scratch in each rolling. While the paper trained the model for 5000 epochs, we only trained 600 epochs due to limited time and resources.

3.3 AutoFormer

Autoformer is a recent work by Wu et al. (2021) published in NeurIPS 2021. As an improvement of Transformer (Vaswani et al. (2017)), the contribution and novelty of Autoformer are highlighted in 2 aspects: First, long-term time series data usually has intricate and complicated temporal patterns, which can overwhelm the point-wise self-attention Transformers models. Since Transformer models only calculates the relation between selected points, the computational complexity rises quadratically with the sequence length. To tackle the intricate temporal patterns, Autoformer incorporates a decomposition structure that splits the time series into seasonality and trend, and only extracts the temporal dependency among sub-series based on the periodicity of the original time series. This extends the dependency from point-wise to period-wise.

Second, Autoformer also proposed an auto-correlation mechanism that discovers and measures the dependencies along the sequence, as well as aggregating the past information. Auto-correlation plays the same role as the attention weight in the Transformer model (Vaswani et al. (2017)), which highlights the positions on the sequence that are deemed to have more association with the proceeding position to be predicted. However, the former is more relevant in the time series analysis literature. The proposed period-wise autocorrelation mechanism is visualized in Figure 7.

The Autoformer architecture, as shown in Figure 5, consists of 2 parts: an encoder and a decoder. The encoder abolishes the trend from the original time series and focuses on learning the seasonality. Then, it passes the extracted seasonality to the decoder. The decoder, on the other hand, stacks the autocorrelated seasonality from the encoder, while simultaneously accumulates the trend cyclicality over time. While the encoder takes the target time series data as input and encodes the seasonality, the decoder's inputs are the initialized seasonality and trend, which are respectively zeros and the mean of the target time series.

The outputs of this encoder-decoder are the stacked seasonality and accumulated trend, which sum up to the final prediction of the target time series.

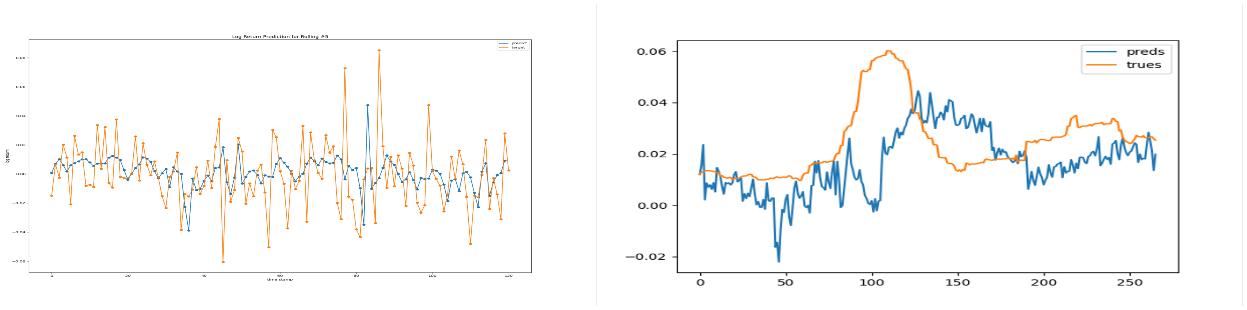
4 Discussion & Conclusion

The WSAEs-LSTM model is evaluated across five testing periods, both with and without sentiment features (Figure 3). Our main observations and conclusions are as follows:

- Volatility increases with rollings, regardless of whether sentiment features are included. This could suggest that the SAEs extract more information as learning iterations increase.
- In the earlier rollings, the model without sentiment features (financial features only) tends to produce greater volatility and outperforms in capturing trends. However, in later rollings, the opposite is observed. This may be due to the larger feature size resulting from the addition of sentiment features, making it challenging for the SAEs to extract compressed information without adequate learning time.
- In the last rolling, although the model with sentiment features demonstrates a slightly more accurate trend than the one without, it is unclear if using sentiment features yields qualitative improvements. Further analysis could involve using more rollings to determine if a more significant difference occurs in later rollings when incorporating sentiment information.

While both the baseline and the WSAE+LSTM have shown the significance of sentiment features, the Autoformer is trained with the sentiment features included in the input. With the fast convergence within 10 epochs, the observation of the Autoformer prediction (Figure 1 (b)) is 2-folded:

- Autoformer tends to overestimate the volatility of the time series, as opposed to WSAE+LSTM that tends to produce a smoothed prediction.
- Autoformer is also sensitive to the trend and lags the true trend, which is due to its trend-cyclicality accumulation structure.



((a)) WSAE+LSTM Prediction

((b)) Autoformer Prediction

Figure 1: Highlight of Prediction Results (true target vs prediction)

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wei Bao, Jun Yue, Y. R. (2017). A deep learning framework for financial time series using stacked autoencoders and longshort term memory. 12(7).

Wu, H., Xu, J., Wang, J., and Long, M. (2021). Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430.

Supplementary Material

Full Financial Features

- Open: The daily open price
- High: The daily highest price
- Low: The daily lowest price
- Close: The daily closing price
- Adj Close: The daily adjusted closing price
- Volume: The daily trading volume
- Log Return: The daily Log return
- Vol: Realized volatility, measured by the standard deviation of the past 30 days.
- MACD: Moving average convergence/divergence indicator
- CCI: The commodity channel index
- BOLL_MID: The Bollinger band mid-rail
- MA5: 5-day moving average

WSAEs-LSTM Architecture

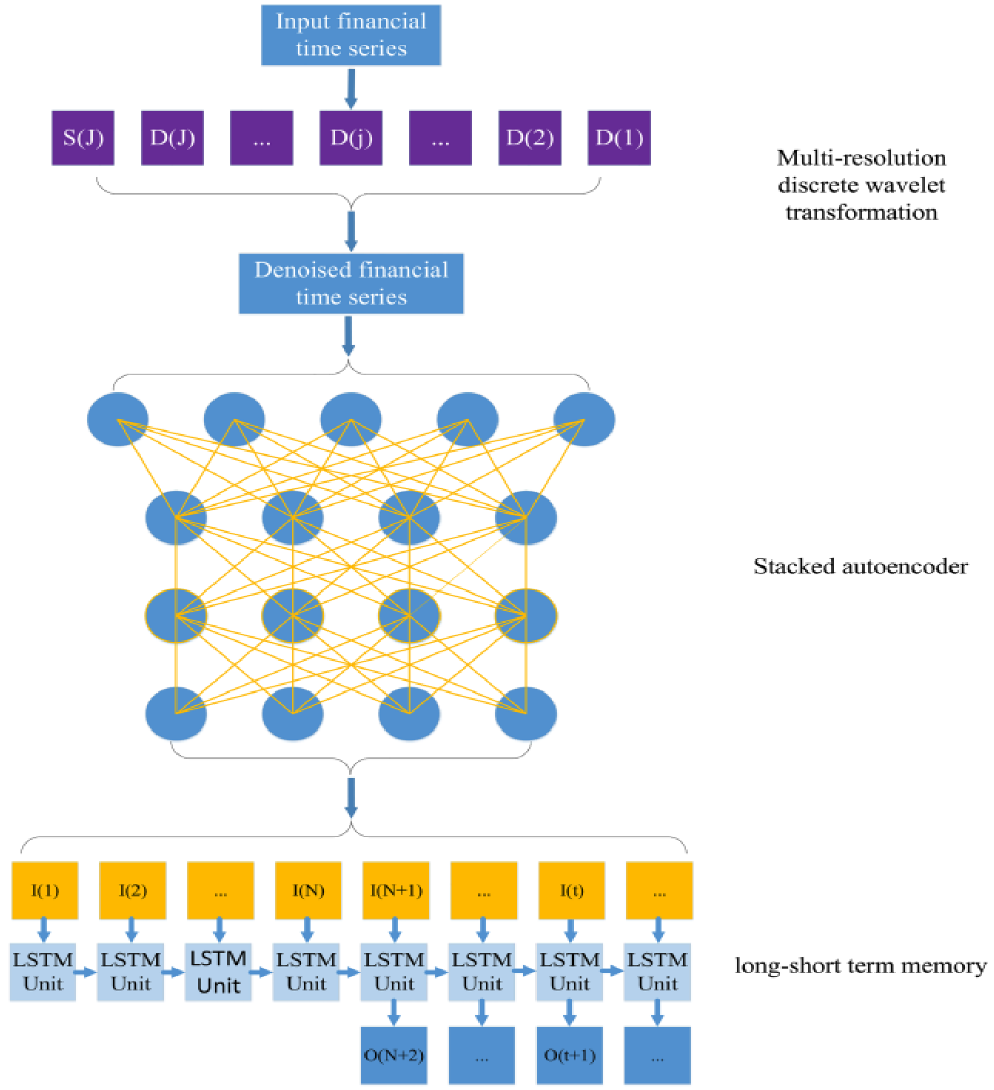
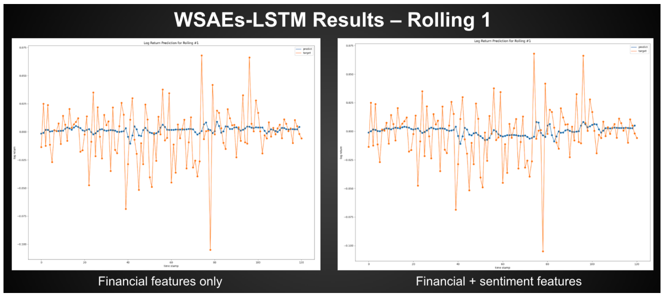


Figure 2: SAEs-LSTM Architecture.

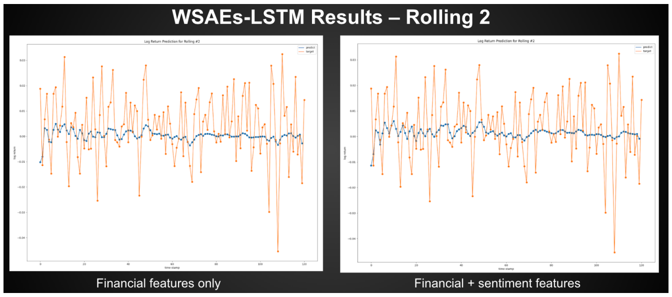
WSAEs-LSTM Model Results



WSAEs-LSTM Results – Rolling 1

Financial features only

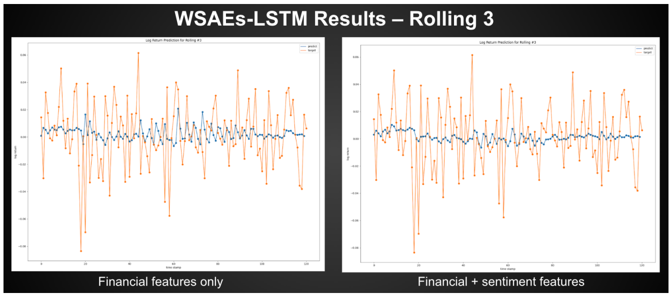
Financial + sentiment features



WSAEs-LSTM Results – Rolling 2

Financial features only

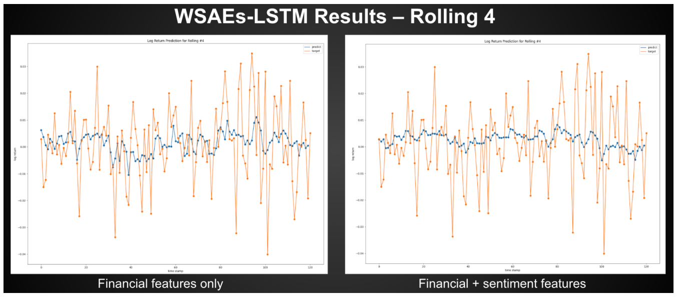
Financial + sentiment features



WSAEs-LSTM Results – Rolling 3

Financial features only

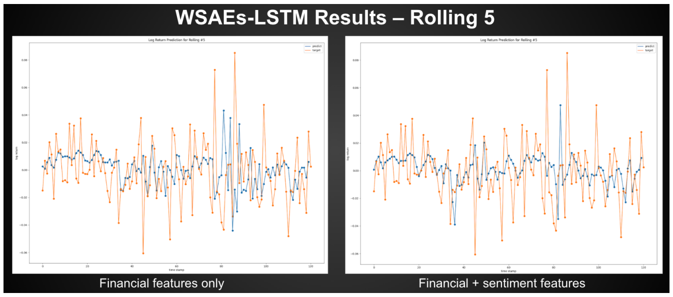
Financial + sentiment features



WSAEs-LSTM Results – Rolling 4

Financial features only

Financial + sentiment features



WSAEs-LSTM Results – Rolling 5

Financial features only

Financial + sentiment features

SAEs Losses

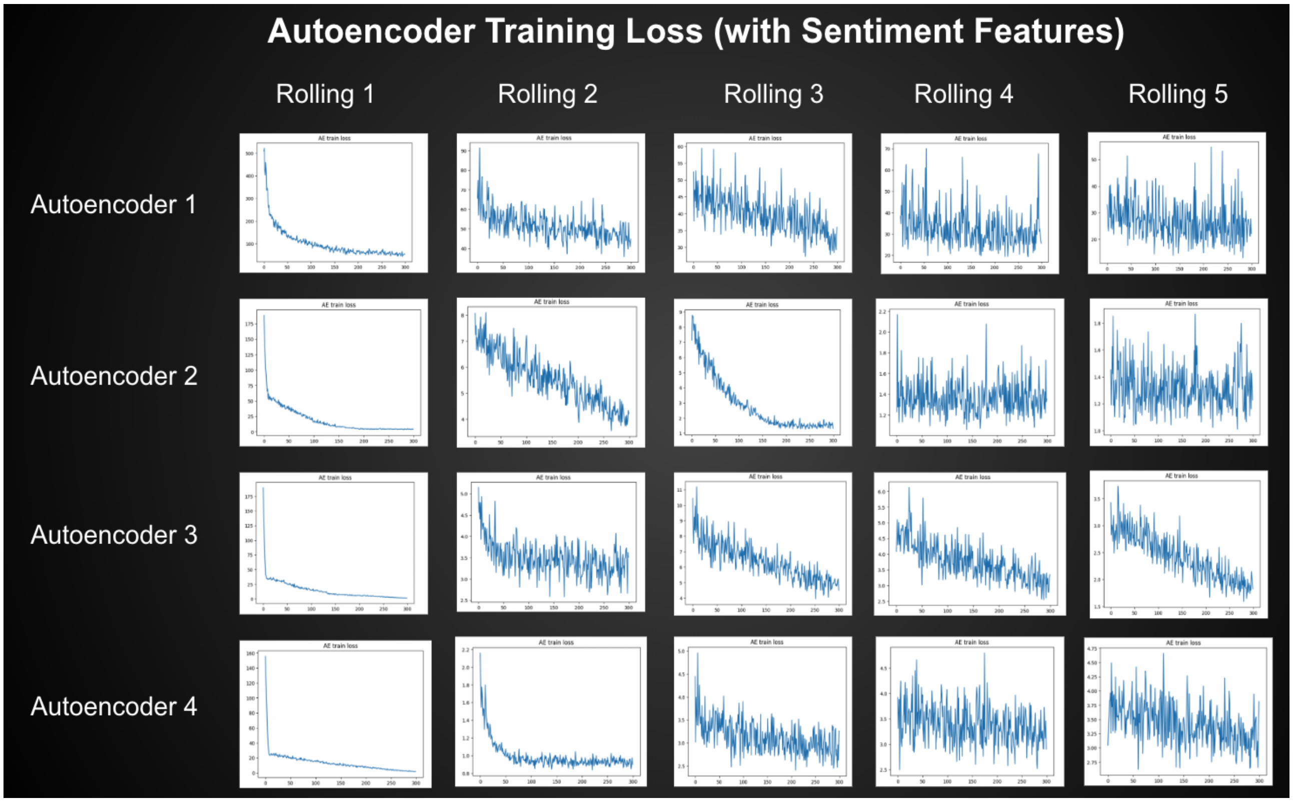


Figure 4: SAEs Loss.

Autoformer Architecture

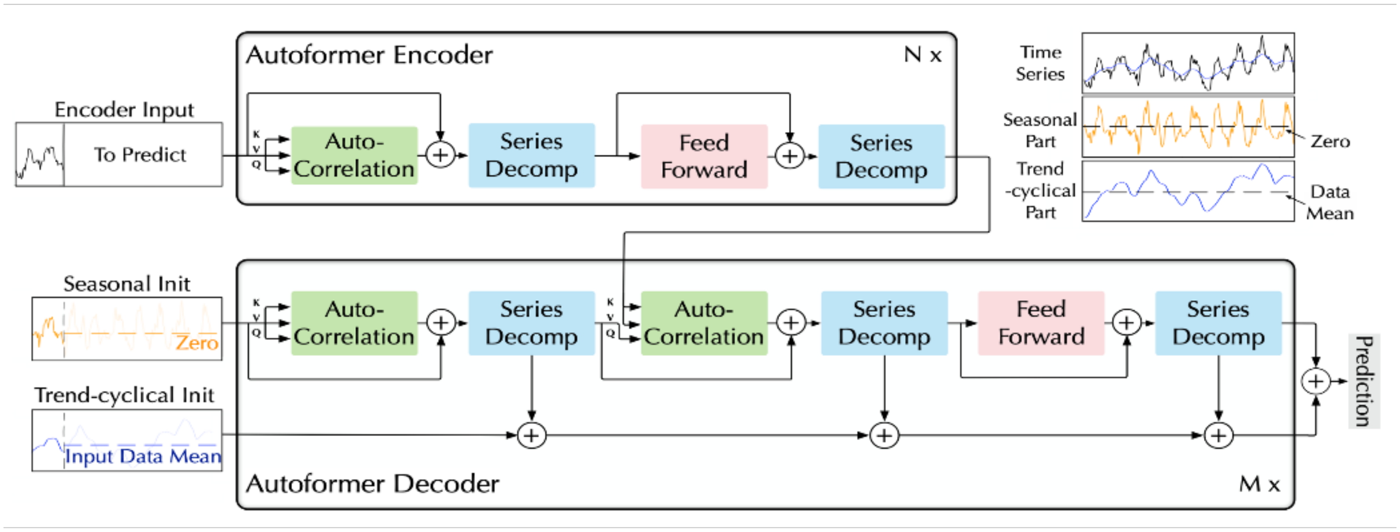
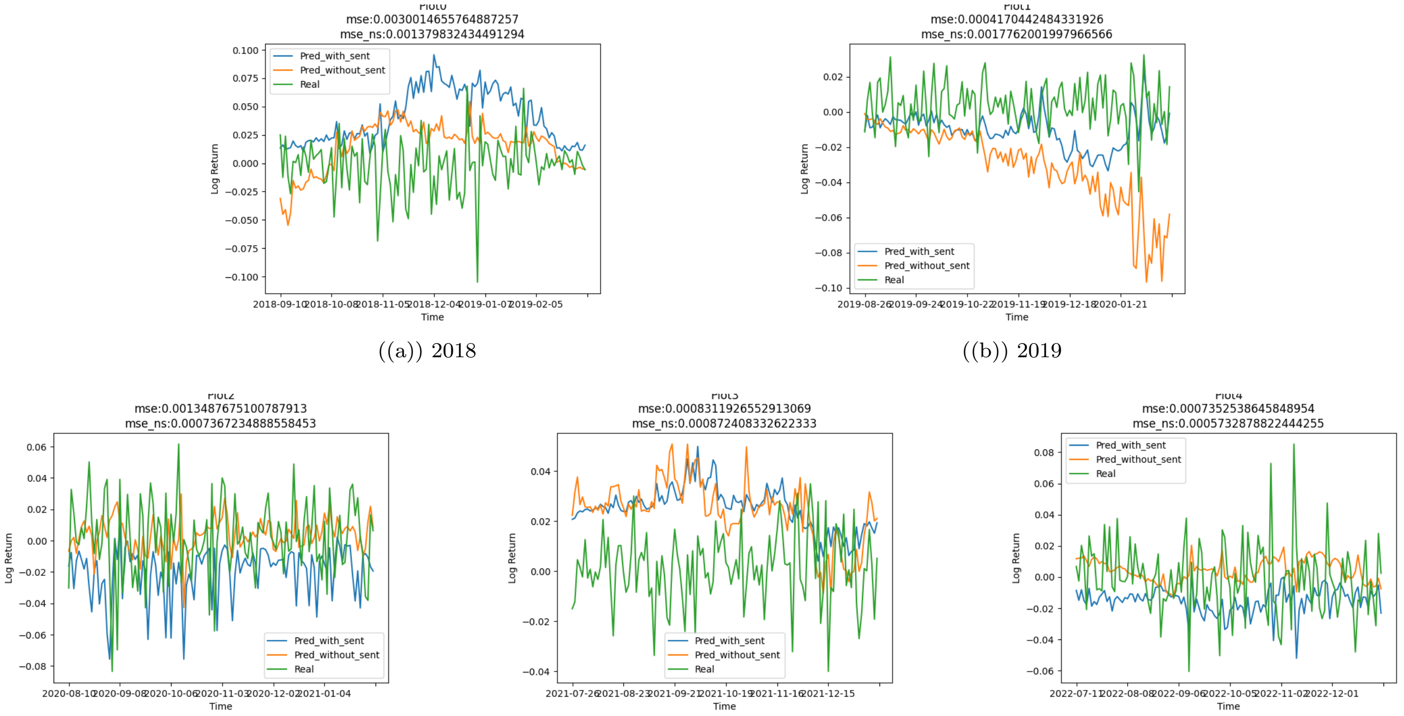


Figure 5: Autoformer Architecture.

Baseline Model Results



((a)) 2018

((b)) 2019

((c)) 2020

((d)) 2021

((e)) 2022

Figure 6: LSTM Results

Illustration of Autoformer's Autocorrelation Mechanism

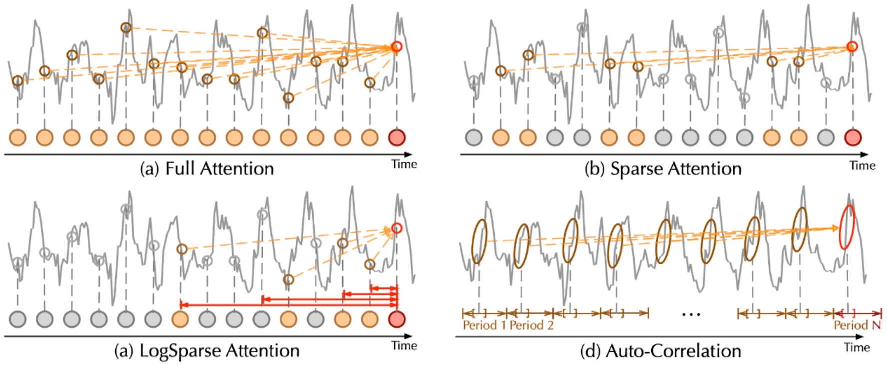


Figure 7: Illustration of the period-wise autocorrelation structure (d) in Autoformer, in constrast to the point-wise attention structures (a), (b) and (c)