MetaLearner

The Generative AI Forecasting System

RAFAEL FERMÍN

KEVIN GOOD

JOSE LAMA

LIM TING HUI

Table of Contents

TABLE OF CONTENTS **2**

1. PROBLEM STATEMENT **3**

2. FORECASTING ARCHITECTURE **3**

3. DATA ENGINEERING **4**

3.1 Time series Features 4
3.2 Technical Indicators and Macro Trends 4
3.3 Natural Language Processing 5
3.4 Attention Layer on Data 6
3.5 Feature Expansion and Elimination 6

4. TRADITIONAL TIME SERIES FORECASTING **6**

4.1 SARIMAX 6
4.2 Long-short Term Memory 6
4.3 Prophet 6

5. TRADITIONAL REGRESSION FORECASTING **6**

5.1 DLinear 7
5.2 LightGBM 7
5.3 Elastic Net 7

6. FORECASTING WITH COVARIANCE MATRICES **7**

7. FORECASTING WITH TRANSFORMERS **7**

7.1 FEDformer 7

8. ADAPTION WITH CONFORMAL PREDICTION **8**

8.1 Quantile Regression 8
8.2 Conformal Quantile Regression 8

9. COMBINING FORECASTING SYSTEMS **9**

9.1 Equally Weighted Forecasting 9
9.2 Optimization 9
9.3 Bootstrapping 9

10. EXPLAINABLE AI **9**

10.1 Feature Importance 10

11. FORECASTING AT SCALE **10**

11.1 Time series features decomposition 10
11.2 Predictability based on History 10

12. APPLICATION **11**

12.1 Linear Programming 11
12.2 Monte Carlo Simulation 11

13. GENERATIVE AI **11**

14. CONCLUSION **12**

APPENDIX A. TIME SERIES FEATURES **15**

Table A1.1 Time Series Features Employed 15
Table A1.2 Time Series Features Employed (cont.) 16
Table A1.3 Time Series Features Employed (cont.) 17

1. Problem Statement

In the domain of time-series data analysis, numerous factors can influence the accuracy of predictions and the difficulty in deriving meaningful insights. These factors include but are not limited to fluctuating data trends, risk of data inconsistency, potential disruptions in the data generation process, and anomalies in data. Furthermore, as time-series data analysis is at the heart of many competitive industries, it is intrinsically an ever-evolving field. Therefore, it is crucial to have an adaptive system, given that practitioners in the field are continually refining their methods.

To help mitigate some of the challenges associated with time-series data analysis, our team has designed a self-adaptive forecasting framework. This system is designed to assist analysts, researchers, and decision-makers in conducting informed interpretations, drawing accurate conclusions, and enhancing the visibility of the associated risks and uncertainties.

For instance, by forecasting trends and shifts in the data, analysts can manage their strategies based on the predicted volatility and optimize their resource allocation. Additionally, potential disruptions such as anomalies and inconsistencies in the data could potentially be preempted as well, providing analysts with a competitive edge in managing their data analysis strategies.

2. Forecasting Architecture

The team has proposed a forecasting platform inspired by various state-of-the-art technologies, ideas, and mathematical models. To improve accuracy, feature engineering and model selection by a suitable set of features is critical to large-scale forecasting success. These features of time series are useful in selecting an optimal subset of forecasting models from all individual models without the need to run all possible combinations of individual models.

With the proposed architecture, users will only be required to communicate the information they wish to learn through the Chat User Interface (UI) as shown in Figure 1. The team has shortlisted various commodities enclosed in Appendix A that could be forecasted with the proposed framework.

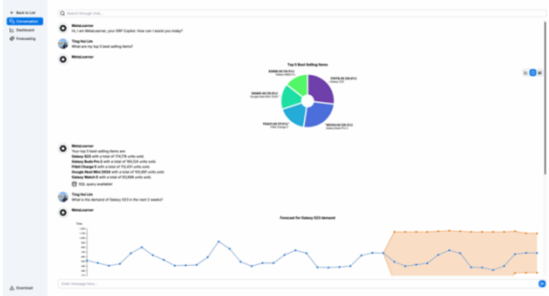

Figure 1. Chat User Interface

Through user-provided data series, our proposed framework decomposes the movements into various time series features, encodes significant events, captures cyclical patterns, and integrates macro trends such as key economic indicators within the time series. The framework also incorporates sentiment analysis and topic modeling of the underlying subject matter and macroeconomic environment as additional features. Details on the data engineering processes are encapsulated in Section 3.

Next, the framework would shortlist the available forecasting methods, which have been proven effective in various time series forecasting domains. Details on the forecasting methods are reviewed in Sections 4, 5, 6, and 7.

As the real world is full of uncertainty, there has been significantly increased interest in band prediction compared to point-wise prediction, as band prediction provides users with a range prediction of where the actual forecast may lie. Band prediction allows users to make informed risk management decisions based on the range provided rather than merely having a point prediction. Details on integrating band prediction in our forecasts are expanded in Section 8.

Ensemble is taking multiple forecasting models and combining them via a statistical method to obtain a superior performance compared to each forecasting model individually. Our team reviewed both the stacking and bootstrapping ensemble methods in Section 9, which each has its advantage of combining models.

Possessing the ability to fathom and trust a forecast that impacts monetary gain is vital. Hence, our team leveraged feature importance to demystify the forecasts provided by the framework. Details on interactions and usage of feature importance are continued in Section 10.

Users can leverage the provided forecasts for decision-making through linear or stochastic programming. This allows the decision-makers to fathom the various uncertainties and how each decision may have a downstream impact.

Lastly, the forecasting architecture is constantly reviewed and improved through Generative Artificial Intelligence (AI) and a team of data scientists. Details on the continuous improvement process are detailed in Section 13.

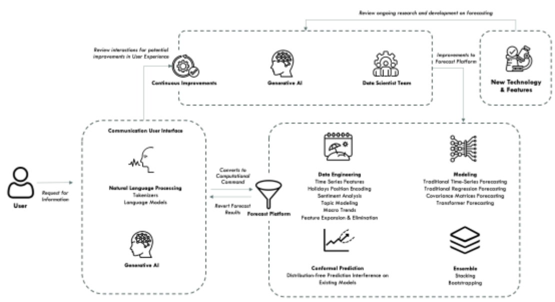

Figure 2. Forecasting Architecture

3. Data Engineering

3.1 Time series Features

Unlike traditional regression data, time series data are subjected to components like trends, seasonality, and cycles. The longer the dataset, the more apparent the time series features will exhibit. Hence, leveraging the behavior exhibited by time series data allows the team to perform forecasts with superior performance.

In the time-series context, we always see trends, seasonality, and cycles. There has been empirical evidence that trend-based insights has generated positive insights throughout various macro environments, including high and low-interest rate regimes, high and low inflation periods, war and peace, recession, and boom [1].

We have computed 68 time series features across 14 major categories. The 14 major categories are listed in Table 1. The full details of the time series features used are in Table B1 under Appendix B.

Table 1. Major categories of time series features

Time Series Major Categories	
ACF PACF Features	BOCP Detector
CUSUM Detector	Holt Parameters
Holt-Winters Parameters	Level Shift Features
Nowcasting	Outlier Detector
Robust Statistics Detector	Seasonality
Special AC	Statistics
STL Features	Trend Detector

3.2 Technical Indicators and Macro Trends

3.2.1 Technical Indicators

Technical analysis is a mainstream investment analysis for active traders in the current environment [2]. Technical indicators attempt to model the short-term price movements of commodities based on heuristic or pattern-based calculations [3].

In financial-based context, we have incorporated various technical indicators as features for our ML models. The indicators we have employed are shown in Table 2.

Table 2. Technical Indicators

Technical Indicators Employed
Welles Wilder's Directional Movement Index
Aroon Indicator
Chande Momentum Oscillator
MACD Oscillator
Relative Strength Index
Stochastic Oscillator / Stochastic Momentum Index
Volatility
Volume-Weighed Average Price

3.2.2 Macro Trends

Macro trends undeniably affect the movements of various time-series. To model the change in the macro environment, we have introduced factors such as unemployment rates, continuing claims, OBFR, SOFR, and Money Supplies as features for the downstream ML models.

3.3 Natural Language Processing

Our team leveraged Bidirectional Encoder Representations from Transformers (BERT) in our Natural Language Processing (NLP) tasks. BERT is a family of Transformers released by Google in 2019 and works by pretraining deep bidirectional representations from unlabelled text [4,5]. To use BERT, users have to train the final layer with the tasks intended, which spawns various BERT models like FinBERT for financial sentiment analysis and BERTopic for topic modeling [6,7]. BERT allows users to introduce a final layer of training for context-specific NLP tasks, which provides a degree of flexibility in leveraging BERT.

To obtain sentiment from natural text, we leverage FinBERT, the BERT model that was further trained on TRC2-financial and Financial PhraseBank. We can retrieve a more accurate representation of the sentiment as the financial context allows FinBERT to accurately decipher the intention behind the sentences more precisely than other language models [8].

With the recent advancement in computing power, Transformer models such as ChatGPT by OpenAI and Pathways Language Model (PaLM) by Google perform exceptionally well in the NLP domain [9,10]. Hence, they are potential model considerations if users have access to high computing power.

3.3.1 Attention on News

Similar to the ideation of transformers, we have implemented an attention layer on the news that was converted into features. Unfortunately, not all news is relevant, which ultimately reflects the movements of the time-series. Hence, it is crucial to sieve irrelevant news to reduce noise and improve the feature relevancy to the downstream machine learning models. The attention layer is modeled by the mathematical function below:

$$\mu_{sentiment} = \min\left(\frac{e^{\frac{a_x}{a_{max}}}}{e^{0.8}}, 1\right) \qquad (1)$$

Where $\mu_{sentiment}$ is the average sentiment of the time-series within a specific period, a_x is the attention of the time-series within the specified period, and a_{max} is the peak attention of the time-series within the specific period.

3.3.2 Topic Modeling

Topic modeling is an NLP technique used to identify and extract main topics from a collection of texts and documents through unsupervised learning. The application of transformer-based models is advantageous as transformers have the ability to capture contextual information from the text.

Through topic modeling, the team can quickly identify opinions and sentiments around topics and the emergence of new topics of interest.

With the recent advancement of transformer models such as GPT-4 and LLaMa, topic modeling will continue to gain significance in the forecasting domain.

3.4 Attention Layer on Data

The inception of the concept of attention has inspired many applications in the domain of machine learning [11]. The paper published by Amazon's data scientist team has highlighted the boost of performance when they have incorporated a decoder-encoder attention layer for context alignment on time series data [12]. Essentially, they have introduced position encoding of holidays and a feedback decoder on the volatility of previous forecasts by the model. Hence, we can leverage the novel concepts that Amazon's data scientist team has proposed in the forecasting framework to boost performance.

3.5 Feature Expansion and Elimination

As the relationship between each feature and the target variable may not have a fixed period difference, our team took inspiration from ACF-PCF analysis. We expanded each feature by lagging each feature to a magnitude of 9 periods.

We employed three independent feature elimination methods with a predefined threshold level to ensure that the ML models' performance is not affected by the introduction of a significant increase in features. If the features created have less significance than the threshold level, the feature will be removed from the final model training.

4. Traditional Time Series Forecasting

As time series data commonly exhibit trends and patterns, many mathematical models leverage the trends and patterns to make an informed forecast of the future.

4.1 SARIMAX

Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) is one of the most popular models in the time series domain. Inspired by the ARIMA model, SARIMAX incorporates the seasonal component on top of the ARIMA base. SARIMAX has shown promising results in various domains, such as demand forecasting [13,14].

4.2 Long-short Term Memory

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that uses a memory cell to store and access information from previous inputs [15]. With these memory cells, LSTM can handle non-stationaries data access long-term dependencies in the data, making LSTM ideal in the time series forecasting domain [16].

4.3 Prophet

Prophet is a time series forecasting tool developed by Meta, which is a decomposable time series model with three components: trend, seasonality, and holidays [17]. When Prophet was introduced, the holidays component was one of the most critical advancements in the domain of time series as it allows the model to account for special events that may impact the data, such as holidays or significant events. Prophet also integrated the Bayesian model, efficiently handling outliers and missing data. The holiday component has inspired various time series modeling techniques, such as MQTransformer.

5. Traditional Regression Forecasting

Traditional regression forecasting frameworks have been popular in time series forecasting as time series are influenced by various factors such as seasonal variations, economic conditions, and market trends. Hence, it is crucial to introduce external data sources to complement the trends and patterns exhibited by the data. The team has noted that many top performers in the renowned M-Series Time series Competition leverage simple regression models.

5.1 DLinear

DLinear is an approach proposed by a group of PhDs from the Chinese University of Hong Kong when they doubted the validity of applying transformers-based models in long-term time series forecasting [18]. They integrated the concept of decomposition scheme in FEDformer, which split the data into a trend component and a seasonal component. Next, they apply a one-layer linear model to each of the components and sum up the predictions of each component to obtain the final prediction. Surprisingly, the simple approach has outperformed the state-of-the-art transformer models in the time series forecasting domain.

5.2 LightGBM

LightGBM is a gradient-boosting framework developed by Microsoft. Unlike most gradient boosting frameworks, LightGBM grows leaf-wise rather than depth-wise [19]. It is capable of handling a large amount of data with relatively little computing power required. It is extremely popular in hackathons as it performs well within the time constraint. However, due to its nature of growing leaf-wise, it is prone to over-fitting compared to other gradient-boosting frameworks. LightGBM's overfitting issue could be combated by limiting the level it could grow.

5.3 Elastic Net

Elastic Net is a linear regression model targeted to solve the sensitivity of linear models with large coefficients. It penalizes the model based on the sum of the absolute coefficient values (L1) and the sum of the squared coefficient values (L2) [20]. By balancing the penalization on both L1 and L2, Elastic Net can reduce model's sensitivity more than linear models that only focus on one area of penalization.

6. Forecasting with Covariance Matrices

Covariance is widely used in the time-series forecasting due to strong correlation among various time-series.

In the same vein, covariance can be used to forecast the movement of differing time-series relative to each other. Additionally, covariance forecasting has also recently picked up momentum in the forecasting domain, where new covariance methods and models have been proposed to forecast uncertainties in the world [21].

The covariance of two time series data is computed below:

$$cov_{A,B} = \frac{\sum(R_A - \overline{R_A})\,(R_B - \overline{R_B})}{N-1} \tag{2}$$

Where R is the movements of the target variable within the specified period.

7. Forecasting with Transformers

Since the success of Transformers in NLP, there has been massive interest in applying transformer models in long-term time series due to their ability to capture complex dependencies between different time points in a series. There have been various Transformers models proposed for time series, such as Informer, Autoformer, Pyraformer, and FEDformer, each achieving astonishing results during their time of publication.

7.1 FEDformer

FEDformer is introduced to address the inability of existing transformer models to capture overall trend by incorporating time series seasonality, trend, and residual into the model [22]. Additionally, the researchers also introduced a Fourier analysis on the time series frequency domain for the model to access the global properties of time series. With the two ideas combined, it emerged as the best transformer model for long-term time series forecasting [22,23].

8. Adaption with Conformal Prediction

Unlike most popular forecasting approaches, Conformal Prediction (CP) allows us to construct distribution-free predictions within a finite sample [24]. Hence, it guarantees to contain the ground truth within a specified probability, as shown in Figure 3 [25].

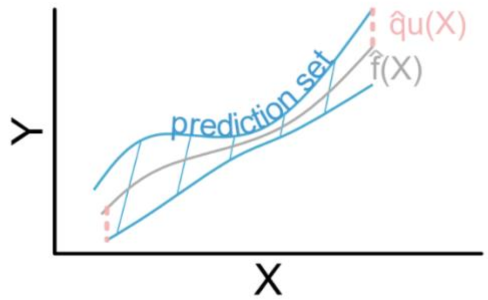

Figure 3. Conformal Regression Prediction [25]

The function of CP is modeled in Eq. 3 below:

$$1-\alpha \leq P\{y\epsilon\tau(X_{n+1})\} \leq C \tag{3}$$

Where α refers to the confidence level, P refers to the prediction set, and C refers to the calibration set.

8.1 Quantile Regression

Differing from most mean-based predictors, Quantile Regression (QR) is a statistical method that allows users to understand the non-linear relationship of target variables to the feature variables.

The formula for QR is given in Eq. 4 below:

$$Q_\tau(y) = B_0(\tau) + B_1x_1(\tau) + \cdots + B_1x_1(\tau) \tag{4}$$

Where τ is the quantile specified, B_n is the weight for each variable, and x_n is the feature variables [26]. It is also governed by a loss function as given in Eq. 5.

$$L_\tau = max\begin{cases} y_\tau - f(x)_\tau \\ y_{1-\tau} - f(x)_{1-\tau} \end{cases} \tag{5}$$

Where L is loss is the specified quantile [27]. With QR, we can model the uncertainties in our models' prediction and adjust our decisions based on the volatility the models exhibit.

8.2 Conformal Quantile Regression

Despite promising theories behind QR and CP, they have inherent disadvantages to be utilized compared to other forecasting methods. QR suffers inaccuracy of its extreme quantiles prediction on new data, and CP tends to provide static bands throughout the whole interval.

Conformal Quantile Regression (CQR) is a technique that combines statistical efficiency in QR while leveraging on the distribution-free coverage in CP proposed by a group of Stanford scholars [28].

CQR is adaptative, indicating that the width of the prediction band is highly correlated to the confidence of the prediction. This would reduce the disadvantages of QR and CP if they were to be used individually.

CQR requires three sets of data, the training data, the calibration data, and the testing data. To demonstrate CQR, we first train any QR on the training data. Then, we use QR to conduct interval predictions on the calibration dataset, which we could leverage on the results to compute the conformity score based on the principles of CP. Lastly, we could combine these two approaches by performing the prediction on the testing data and subtracting one end of the QR extremities band from the conformity score.

The formula for the conformity score is given in Eq. 6:

$$CS_i = \max(|\, y_i - f(x_i)|) \tag{6}$$

Where CS is the conformity score, y is the target variable, and f(x) is the prediction on the calibration set.

In the real world, knowing the probability of target variable distribution within the regression band that guarantees ground truth within the user-specified confidence level allows users to perform data-driven decisions with an appropriate risk management framework.

9. Combining Forecasting Systems

Ensembling different forecasting systems (FS) often utilize the weightage system, modeled by Eq. 7 below:

$$FS = W_1 FS_1 + \cdots + W_N FS_N, \quad s.t. \sum_{i=0}^{N} W_i = 1 \tag{7}$$

Where W is the weight of each FS, and N is the number of FS.

9.1 Equally Weighted Forecasting

As its name suggests, an equally weighted forecast system (EWFS) assigns a static $W_{\frac{1}{N}}$ to Eq 7, which assumes equal weightage of all FS. This approach is often used by researchers if there is no additional information that could be gained to enhance the performance of the FS.

In theory, the performance of EWFS is subjected to Eq. 8:

$$P_{Worst} \geq P_{EWFS} \geq P_{Best} \tag{8}$$

Where the performance of the EWFS is better than the worst FS and worse than the best FS ensembled.

9.2 Optimization

The optimization approach attempts to calculate the optimal weightage of each FS to produce an ensembled FS that grants superior performance compared to all FS ensembled based on past performance. It has shown tremendous success in time series domains like demand forecasting and logistic application. However, optimization should be applied with caution in forecasting as optimizing FS produces an ensembled FS that may be better or worse than each FS utilized [29].

9.3 Bootstrapping

To produce an FS that is better than EWFS yet limits the downside of fully optimized FS, we leveraged the concept of bootstrapping, which combines EWFS and optimized FS.

Our bootstrapped forecasting system (BSFS) computes the optimized weightage for each FS in a fixed period N. With the optimized weightage in each period N, we combine those weights through a factor of M, which is the number of periods within the predictive horizon. BSFS is modeled by Eq. 9 below:

$$BSFS = \frac{1}{M} \sum_{i=0}^{M} W_1 FS_1 + \cdots + W_N FS_N \tag{9}$$

Through bootstrapping, we can increase the performance of our ensemble FS while preventing over-optimization. The less volatility the optimized weight of each FS in each period, the closer the BSFS models optimized FS.

10. Explainable AI

Model interpretability is gaining traction as the adoption of Machine Learning in various applications increases. Understanding how the machine derives the forecast allows users to explain outcomes to stakeholders. Furthermore, experienced traders could also exercise their judgment to determine if the outcome is reasonable. These positive interactions and reinforcement enable the forecasting framework to continuously learn and improve performance.

10.1 Feature Importance

Feature importance is the concept of determining how important a feature is to influence the prediction of the model relative to other features present. To represent the ensemble model, the team will leverage the linear-based models and the tree-based models' feature importance calculation method via a weighted approach. The formula for the feature importance weightage for a stacked model is depicted in Eq. 10 below:

$$W_{M1}F_{1M1} + W_{M\ldots}F_{\ldots M\ldots} + W_{Mi}F_{jMi} = 1 \qquad (10)$$

Where Mi is the i-th model in the stacked model, and Fj is the j-th feature in the i-th model. To present the most significant features to the users, the team has employed Eq. 11 below:

$$F_j = \sum_{1}^{i} W_{Mi}F_{jMi} \qquad (11)$$

By employing Eq. 10 and Eq. 11, users can quickly fathom what are the features that influenced the model as depicted in Figure 4.

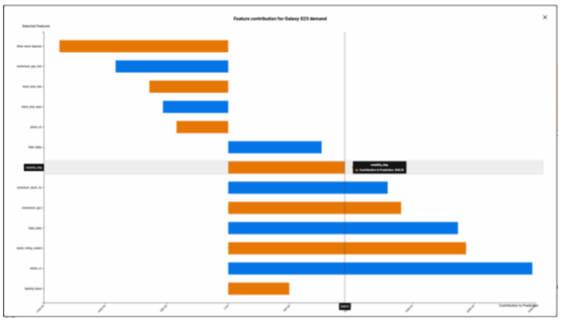

Figure 4. Explaining factors affecting the model prediction

11. Forecasting At Scale

Computing power is one of the significant constraints in the domain of forecasting. Depending on the size of the time series and the modeling process for the forecast, training a time series forecast workflow from scratch could take up to days. Hence, our team proposed a method that took inspiration from recommendation engines.

11.1 Time series features decomposition

Similar to how Amazon recommends products based on users' demographic and behavior, our team decided to treat each time series as a "customer" and profile them based on their time series characteristics [30]. Every time a time series is trained, the time series features based on Table 1, optimal model stack, and parameters are stored in a data repository. Once enough models for diverse time series data are trained and stored in the data repository, we can create a meta learner like Random Forest to predict the optimal model stacks and parameters for new time series based on the decomposed time series features. This would allow users to conduct large-scale forecasting with a close-to-optimal result at a fraction of the original computing cost.

11.2 Predictability based on History

It is evident that the data repository would not have all the time series in the world. Hence, the team has introduced the concept of predictability to prevent users from obtaining subpar results from forecasting at scale. Predictability is essentially the F-beta score where we strike a balance between the precision and recall of this classification task. The formula of F-beta is modeled in Eq.12 below:

$$F_\beta = \frac{(1+\beta^2) * (precision * recall)}{(\beta^2 * precision + recall)} \qquad (12)$$

Where β is the weight assigned to recall relative to precision, precision is the probability of true positive within predicted positive, and recall is the probability of true positive among all positive cases—having a higher β results in higher weightage in the recall.

Additionally, predictability also highly depends on the number of diverse time series trained in the data repository. Figure 5 denotes a general depiction of the performance of the meta-learner based on the number of diverse data trained. Within the figure, there is an arbitrary inflection point, where the performance of the meta-learner will increase exponentially past the point. Hence, as the usage of this forecasting workflow increases, the performance of the meta-learner increase exponentially.

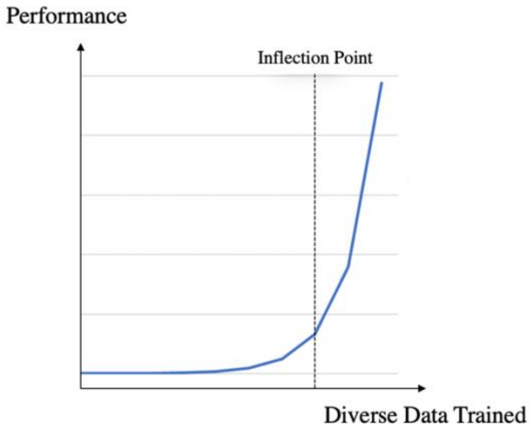

Figure 5. General Performance of Metalearner

12. Application

After obtaining the forecast from our workflow, users should have a sense of where the target variable will lie within the specified confidence interval. With that information, users can model the parameters and integrate the necessary risk management framework based on the prediction obtained. Linear Programming and Monte Carlo simulation are some popular methods to model the optimal approaches users should take to maximize gains, minimize risks, or strike a balance of both.

12.1 Linear Programming

Linear programming is an optimization technique that users use to solve problems with linear constraints [31]. When applied to the time series domain, it can be used to model decisions that have to be made based on the forecasted values.

For instance, in the context of portfolio management, users can model the decisions (the position of each asset) to strike a balance on the Sharpe Ratio. An example of linear programming is modeled in Eq. 13:

$$Max: SR = \frac{E_P - RFR}{\sigma_P} \quad \text{where } E_P = \sum_{i=0}^{N} \hat{R}_i W_i \tag{13}$$

Where SR is Sharpe Ratio, E_P is expected return of the portfolio, RFR is the risk-free rate, and σ_P is the standard deviation of the portfolio, $\hat{R}$ is the forecasted return of the asset, and W is the weightage of the assets in the portfolio. However, there are uncertainties in the world, and decision-makers must incorporate the concept of uncertainties into their decision-making process. To combat this, users can create scenarios like the bottom 5%, median, and top 95% scenarios and set up multiple linear optimization models to account for this uncertainty. Alternatively, users can turn to stochastic programming like Monte Carlo simulation to model all the uncertainties in the problem.

12.2 Monte Carlo Simulation

Monte Carlo simulation is a stochastic method that models uncertainties by generating random samples from various probability distributions. For instance, we could generate random samples on the forecasted price movements and model the final potential outcomes based on the samples. Based on the probability distribution of the outcome, we can create decisions to maximize gains or to assert certain risk management to protect against low-probability high-impact scenarios.

13. Generative AI

To ensure that the forecasting framework remains competitive in the future, the team proposed a scalable improvement method for the forecasting architecture to continuously improve.

When there are new technologies or features published in academia, the papers and the idea will be conveyed to the Generative AI by the team of data scientists. The Generative AI will review the existing architecture and propose if the new technologies and features can be applied to the existing architecture for improvements. The team of data scientists will review the proposals generated by the Generative AI and implement if the suggestion is valid and effective. This method allows the team to digest existing information at a highly efficient pace.

Furthermore, through the use of Generative AI, the machine may potentially draw connections across different papers and suggest a brand-new approach that may improve the architecture significantly. Hence, by leveraging Generative AI, the forecasting architecture is constantly under scrutiny and challenged.

14. Conclusion

The team has proposed a framework that incorporated various state-of-the-art technologies, ideas, and mathematical models to gain insights into the uncertain future. With these insights, researchers, analysts, and decision-makers can formulate strategic plans to optimize outcomes or refine their predictive strategies. Furthermore, the team recognizes the challenges of forecasting at scale. Hence, the team has suggested a feature-based approach that breaks the time-series down into features and proposes the ideal configuration of models and parameters to conduct the forecast at a fraction of the costs imposed by the traditional framework. With the scalability, this forecasting framework would be applied to various time-series that require insights into the future.

References

[1] B. Hurst, Y. H. Ooi, and L. H. Pedersen, “A century of evidence on trend-following investing,” The Journal of Portfolio Management, vol. 44, no. 1, pp. 15–29, 2017.

[2] L. Yin and Q. Yang, “Predicting the oil prices: Do technical indicators help?,” Energy Economics, vol. 56, pp. 338–350, 2016.

[3] J. Chen, “Technical indicator definition,” Investopedia, 29-Jun-2022. [Online]. Available: https://www.investopedia.com/terms/t/technicalindicator.asp.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, “Attention Is All You Need,” Jun. 2017.

[5] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding,” 2019.

[6] D. Aract, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models,” Jun. 2019.

[7] M. Grootendorst, “Topic Modeling with BERT,” *Medium*, 06-Oct-2020. [Online]. Available: https://towardsdatascience.com/topic -modeling-with-bert-779f7db187e6.

[8] D. Aract, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models,” Jun. 2019.

[9] OpenAI, “CHATGPT: Optimizing language models for dialogue,” OpenAI, 30-Nov-2022. [Online]. Available: https://openai.com/blog/chatgpt/.

[10] A. Chowdhery and S. Narang, “Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance,” Google Research, 04-Apr-2022. [Online]. Available: https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html.

[11] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention is all you need. arXiv preprint arXiv:1706.03762.

[12] C. Eisenach, Y. Patel, and D. Madeka, “MQTransformer: Multi-Horizon Forecasts with Context Dependent and Feedback-Aware Attention,” Jan. 2022.

[13] Y. Wang, J. Wang, G. Zhao, and Y. Dong, “Application of residual modification approach in seasonal Arima for electricity demand forecasting: A case study of China,” Energy Policy, vol. 48, pp. 284–294, 2012.

[14] S. V. Kumar and L. Vanajakshi, “Short-term traffic flow prediction using seasonal ARIMA model with limited input data,” European Transport Research Review, vol. 7, no. 3, 2015.

[15] A. Pulver and S. Lyu, “LSTM with working memory,” 2017 International Joint Conference on Neural Networks (IJCNN), 2017.

[16] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A comparison of Arima and LSTM in forecasting time series,” 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018.

[17] Facebook Open Source, “Prophet Documentation,” Prophet, 05-Mar-2017. [Online]. Available: https://facebook.github.io/prophet/docs/quick_start.html#python-api.

[18] A. Zeng1, Q. Xu, L. Zhang, and M. Chen, “Are Transformers Effective for Time Series Forecasting?,” International Digital Economy Academy (IDEA), Aug. 2022.

[19] Microsoft, “LightGBM Documentation,” Welcome to LightGBM's documentation! - LightGBM 3.3.2 documentation, 24-Apr-2017. [Online]. Available: https://lightgbm.readthedocs.io/en/v3.3.2/.

[20] J. Brownlee, “How to develop elastic net regression models in Python,” MachineLearningMastery.com, 11-Jun-2020. [Online]. Available: https://machinelearningmastery.com/elastic-net-regression-in-python/.

[21] C. Trucíos, J. H. Mazzeu, M. Hallin, L. K. Hotta, P. L. Valls Pereira, and M. Zevallos, “Forecasting conditional covariance matrices in high-dimensional time series: A general dynamic factor approach,” Journal of Business & Economic Statistics, pp. 1–13, 2021.

[22] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, “FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting,” Jun. 2022.

[23] P. Lara-Benítez, L. G. Ledesma, M. Carranza-García, and J. M. Luna-Romera, “Evaluation of the Transformer Architecture for Univariate Time Series Forecasting,” Advances in Artificial Intelligence, pp. 106–115, Sep. 2021.

[24] S. Tadakaluru, “What is quantile regression? Introduction to quantile regression,” 22-Mar-2022. [Online]. Available: https://www.mygreatlearning.com/blog/what-is-quantile-regression/.

[25] A. Angelopoulos and S. Bates, “A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification,” Jul. 2021.

[26] S. Tadakaluru, “What is quantile regression? Introduction to quantile regression,” 22-Mar-2022. [Online]. Available: https://www.mygreatlearning.com/blog/what-is-quantile-regression/.

[27] C. Garcia, “Quantile Regression,” *GitHub*, 06-Jul-2021. [Online]. Available: https://github.com/cgarciae/quantile-regression.

[28] Y. Romano, E. Patterson, and E. Candes, “Conformalized Quantile Regression,” 2019.

[29] R. Carver, *Systematic trading: A unique new method for designing trading and investing systems*. Petersfield, Hampshire: Harriman House Ltd., 2015.

[30] A. Krysik, “Amazon's product recommendation system in 2021: How does the algorithm of the ecommerce giant work?,” Recostream, 14-Oct-2021. [Online]. Available: https://recostream.com/blog/amazon-recommendation-system.

[31] W. Chung, “Applying large-scale linear programming in business analytics,” 2015 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), 2015.

Appendix A. Time Series Features

Table A1.1 Time Series Features Employed

Category	Variable	Definition
ACF PACF Features	y_acf1	First ACF value of the original series
	y_acf5	Sum of squares of first 5 ACF values of original series
	diff1y_acf1	First ACF value of the differenced series
	diff1y_acf5	Sum of squares of first 5 ACF values of differenced series
	diff2y_acf1	First ACF value of the twice-differenced series
	diff2y_acf5	Sum of squares of first 5 ACF values of twice-differenced series
	y_pacf5	Sum of squares of first 5 PACF values of original series
	diff1y_pacf5	Sum of squares of first 5 PACF values of differenced series
	diff2y_pacf5	Sum of squares of first 5 PACF values of twice-differenced series
	seas_acf1	Autocorrelation coefficient at the first seasonal lag
	seas_pacf1	Patial Autocorrelation coefficient at the first seasonal lag
BOCP Detector	bocp_num	Number of changepoints detected by BOCP Detector
	bocp_conf_max	Max value of the confidence of the changepoints detected
	bocp_conf_mean	Mean value of the confidence of the changepoints detected
CUSUM Detector	cusum_num	Number of changepoints, either 0 or 1
	cusum_conf	Confidence of the changepoint detected, 0 if not changepoint
	cusum_cp_index	Index or position of the changepoint detected within the time series
	cusum_delta	Delta of the mean levels before and after the changepoint
	cusum_llr	Log-likelihood ratio of changepoint
	cusum_regression_detected	Boolean - whether regression is detected by CUSUM
	cusum_stable_changepoint	Boolean - whether changepoint is stable
	cusum_p_value	P-value of changepoint

Table A1.2 Time Series Features Employed (cont.)

Category	Variable	Definition
Holt Parameters	holt_alpha	Level parameter of the Holt model
	holt_beta	Trend parameter of the Holt model
Holt-Winters Parameters	hw_alpha	Level parameter of a fitted Holt-Winter's model
	hw_beta	Trend parameter of a fitted Holt-Winter's model
	hw_gamma	Seasonal parameter of a fitted Holt-Winter's model
Level Shift Features	level_shift_idx	Location of the maximum mean value difference, between two consecutive sliding windows
	level_shift_size	Size of the maximum mean value difference, between two consecutive sliding windows
Nowcasting	nowcast_roc	Indicating return comparing to step n back
	nowcast_ma	Indicating moving average in the past n steps
	nowcast_mom	Indicating momentum
	nowcast_lag	Indicating lagged value at the past n steps
	nowcast_macd	Moving average convergence divergence is a trend-following momentum indicator that shows the relationship between two moving averages of a security's price
	nowcast_macdsign	MACD combined with EWMA with a decay window of 9 and minimum periods of 8
	nowcast_macddiff	MACD minus MACDsign
Outlier Detector	outlier_num	Number of outliers
Robust Statistics Detector	robust_num	Number changepoints detected by the Robust Stat Detector
	robust_metric_mean	Mean of the Metric values from the Robust Stat Detector
Seasonality	seasonal_period	Detected seasonality period
	trend_mag	Slope acquired via fitting simple linear regression model on the trend component as trend magnitude
	seasonality_mag	Difference between the 95 percentile and 5 percentile of the seasonal component as the seasonality magnitude
	residual_std	Standard deviation of the residual component
Special AC	firstmin_ac	The time of first minimum in the autocorrelation function
	firstzero_ac	The time of first zero crossing the autocorrelation function

Table A1.3 Time Series Features Employed (cont.)

Category	Variable	Definition
Statistics	length	Length of the time series array
	mean	Average of the time series array
	var	Variance of the time series array
	entropy	Getting normalized Shannon entropy of power spectral density
	lumpiness	Calculating the variance of the chunk-wise variances of time series
	stability	Calculating the variance of chunk-wise means of time series
	flat_spots	Maximum run-lengths across equally-sized segments of time series
	hurst	Hurst exponent measures the amount that a series deviates from a random walk
	std1st_der	Standard deviation of the first derivative
	crossing_points	The number of times a time series crosses the median line
	binarize_mean	Convert time series array into a binarized version
	unitroot_kpss	Test a null hypothesis that an observable time series is stationary around a deterministic trend using KPSS
	heterogeneity	Engle's Lagrange Multiplier test to check for the existence of Autoregressive Conditional Heteroscedasticity (ARCH)
	histogram_mode	Measures the mode of the data vector using histograms with a given number of bins
	linearity	R square from a fitted linear regression
STL Features	trend_strength	Strength of trend
	seasonality_strength	Strength of seasonality
	spikiness	Variance of the leave-one-out variances of the remaining component
	peak	Location of peak
	trough	Location of trough
Trend Detector	trend_num	Number of trends detected by the Kats Trend Detector
	trend_num_increasing	Number of increasing trends
	trend_avg_abs_tau	Mean of the absolute values of Taus of the trends detected