Advances in Consumer Research
https://acr-journal.com/
Volume-3 | Issue-1 | Jan 2026
Original Researcher Article

Enhanced Forecasting of Long Term Index Fund Prices Using A Hybrid Model

Venkata Vara Prasad[1*], Suresh Jagannadhan[2], Yashasvee Vijaykumar[3], Karthik Vijayakumar[4],Srinivas Gumparthi[5]
[1*,2,,43]Sri Sivasubrmaniya Nadar College of Engineering, [5]Mahalakshmi Tech Campus

ABSTRACT
Forecasting the stock market is a paramount challenge owing to its volatility and sensitivity and hence, precise prediction becomes essential for investors. This study solves the problem of index fund forecasting as well as anomaly detection, a twin problem that is mostly ignored in classical models. The significance is in facilitating improved risk management and investment choices. Hence creating a gap in comprehensive financial modeling.
Inspired by the necessity of a strong system incorporating these features, this research presents a new architecture that unites anomaly detection, forecasting. The process starts with preprocessing index fund data, then detects anomalies in a 2 layered anomaly detection module. The trend forecasting engine utilizes Stacked GRU and Holt-Winters models. Tests on Nifty Auto and Nifty Bank data sets yielded unique anomaly scores of 14.28 and 16.23 respectively out of 50, demonstrating the capability of the model to distinguish index behaviors. An MAPE score of 0.687 was observed on Nifty Pharma forecast while using Random Forest to combine predictions within the forecast engine.
Keywords: forecast; index fund; hybrid; model; deep learning.

1. INTRODUCTION:

Background

The financial market operates in a complex and dynamic environment, influenced by various macroeconomic factors, investor sentiment, and global events. Predicting market trends, particularly the price movements of index funds, requires an understanding of both historical and real-time data patterns. Traditionally, statistical models have been employed to forecast prices based on past data trends. However, with the evolution of machine learning, there has been a shift towards integrating advanced computational methods, providing opportunities to enhance prediction capabilities in the financial sector.

Motivation

In recent years, the financial market has witnessed significant volatility and unpredictability, prompting investors to seek reliable methods for predicting movements of index funds. With the increasing complexity of financial instruments and the vast amounts of data available, traditional forecasting methods often fall short. This thesis aims to explore innovative approaches to improve forecast accuracy by using a combination of models within its forecasting engine and also perform anomaly detection to ultimately help investors make informed decisions and enhance portfolio management strategies.

Problem Definition

Despite the advancements in predictive modeling, many existing approaches either rely solely on historical price data or fail to integrate diverse datasets effectively. This research seeks to address the following problem: How can a hybrid model that combines different forecasting techniques improve the accuracy of index fund price forecast? By developing and evaluating a hybrid model that integrates traditional statistical models with deep learning models, this study aims to contribute to the existing body of knowledge in financial forecasting and provide practical insights for investors.

This research also aims to provide the end user with a confidence score for an index fund , this is provided by the anomaly detection module. Which applies a wide range of rules , conditions to check for possible anomalies.

2. LITERATURE SURVEY

Papers related to the problem identified

In the domain of stock market prediction, various approaches and models have been explored to improve forecasting accuracy and handle the complexities of long-term predictions. This literature survey reviews several key studies, highlighting their unique features, contributions, and limitations.

Chahua´n-Jime´nez et al. [4] investigates multiple models, including LSTM, GRU, CNN, ANN, and RNN, for stock market index forecasting. The study emphasizes the potential of ensemble techniques for future research. It highlights the challenges associated with LSTM in long-term forecasting, particularly its complexity due to the three-gate architecture, which results in numerous parameters and increased computational costs. This complexity can adversely affect cumulative returns, especially at critical points.

Kumar et al. [9] offer a comprehensive review of stock market prediction using machine learning and statistical techniques. The paper suggests leveraging historical stock data and compares various models, providing insights into their usage frequency and effectiveness. This systematic review helps understand the evolution of predictive techniques and their applicability.

Lu and Xu [10] present the TRNN model, a custom time-series recurrent neural network, which closely resembles the original RNN but incorporates time series compression and moving average with a sliding window. Their approach aims to enhance prediction efficiency, highlighting the potential benefits of using a tailored model for time-series data.

Billah et al. [3] compare various moving average methods, such as SMA and EMA, with LSTM models. The study finds that while LSTM performs well in the short term, SMA and EMA methods are more effective in the long term. This comparative analysis underscores the strengths and limitations of different forecasting techniques over varying time horizons.

Md et al. [13] introduce a novel optimization approach for stock price forecasting using a multi-layered sequential LSTM (MLSLSTM). They compare MLSLSTM with LSTM, CNN, MLP, and RNN models, demonstrating that MLSLSTM provides superior metrics. This study highlights the advantages of multi-layered approaches in enhancing forecasting performance.

Akhter Mohiuddin Rather [1] explores a new LSTM-based Deep Learning Model for Stock Prediction and a Predictive Optimization Model for portfolio selection.The model uses a novel autoregressive moving pointer model (AMPM) implemented on LSTM-DNN for stock price prediction, and a predictive portfolio model (PPM) using Shannon entropy for portfolio optimization. Experiments on NIFTY-50 stock data show the proposed model outperforms standard predictive models and portfolio optimization models, with RMSE values for LSTM-DNN significantly lower than MLP (case in point, 2.09 vs 9.10 for Stock 1).

M.Mohan et al. [11] proposes a stock market prediction system using three main techniques: Holt-Winters Triple Exponential Smoothing, Recurrent Neural Networks and a Modified Recommendation System.The model uses a real-time dataset of 15 stocks from different sectors, predicting stock prices for the next quarter based on 3 years of historical data (750 rows per security).The system achieved an average RMSE value of less than 50 in many cases when forecasting closing prices for Q1 2018 (01-01-2018 to 31-03-2018) using data from 01/01/2015 to 31/12/2017. It emphasises the downside in using conventional techniques such as news-based feed systems and proposes a hybrid model integrating a recommendation system.

C R Karthik et al [6] compares Deep Neural Network (DNN) and Long Short-Term Memory (LSTM) models for predicting daily variance of the NIFTY IT index.The models use 10 years of historical data from the Bombay Stock Exchange (BSE) for the NIFTYIT index, which includes data from 10 major IT companies.

DNN and LSTM performed well, but LSTM outperformed DNN. DNN achieved a minimum loss of 9.65e-5 and maximum of 0.2265. LSTM achieved a minimum loss of 8.23e-5 and maximum of 0.0003974. LSTM showed better accuracy and lower loss metrics compared to DNN.

TabNet, introduced by Arik and Pfister [16], is a deep learning architecture for tabular data that utilizes sequential attention to dynamically select relevant features at each decision step. Unlike traditional models, it employs an interpretable attention mechanism to enhance learning efficiency and generalization. The model is composed of shared and independent feature transformers, enabling both representation learning and direct decision-making.

TabM, proposed by Gorishniy et al. [15], is an MLP-based deep learning model for tabular data that integrates structured enhancements for improved performance. It uses GELU activation for smoother gradient flow and dropout regularization to prevent overfitting. Additionally, it employs an ensemble of eight sub-models with input/output scaling and bias adjustments to stabilize learning, making it highly effective for time-series regression tasks like stock price forecasting.

Papers related to the problem identified

•Chahua´n-Jime´nez et al. [4] : Long-term LSTM predictions at cumulative turning points tend to perform poorly due to the absence of an adaptive cross-entropy loss function. The model utilizes S&P 500 data from Yahoo Finance, relying on only the seven basic attributes.

Solution: Using a Stacked GRU which does not have this defect shown by LSTM. Performing extensive feature engineering to include more features.

•Billah et al. [3] :Ability of LSTM to focus on hidden patterns which gives it better predictions in short term but SMA, EMA are better in long term Ability of LSTM to focus on hidden patterns gives it better predictions in the short term, but SMA and EMA are better in the long term.

Solution: Using a combination of Stacked GRU with Holt-Winters model to improve long-term predictions while keeping short-term fluctuations in mind.

3. PROPOSED METHODOLOGY

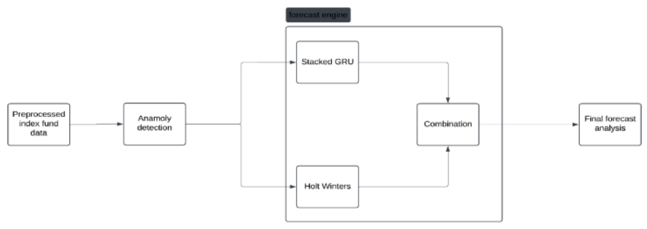
System Overview

The architecture design as described pictorially begins with data preprocessing, where the raw data is cleaned and prepared for analysis. Following this, novel features like moving averages, P/E ratios, and Bollinger bands are created to enrich the dataset, making it more informative for predictive modeling.

Following this the data is fed into the Anomaly Detection module which provides a score out of 50 for the input index fund. The higher the score the greater the anomalous nature of the fund.

Following this the data is fed into 2 models i.e the Stacked GRU and Holt Winters model , both of which train on the historical data and provide us with a forecast

Finally the individual forecasts are combined using various methods.



Description of each module

Preprocessed Index Fund Data

Data preprocessing is a critical first step in any machine learning workflow. In this project, we dealt with various data quality issues. This included handling missing values (NaNs) by removing records with significant gaps. We also reversed the order of the dataset to align with the chronological sequence of stock prices, which is crucial for time series analysis. Furthermore, we decoded all attribute types to ensure that they are appropriately formatted for processing. Special attention was given to anomalous events, such as the 2008 financial crisis and the 2020 COVID pandemic.

The preprocessed index fund data includes data for several index funds sourced from nse.org. These include nifty 50 , nifty Pharma, nifty IT, nifty Auto etc. These cover a wide range of major industries.

We add several financial indicator aid in developing models. These include Simple Moving Average over a 10 day period, Exponential Moving Average ofer 12, 26 days, Moving Average Convergence Divergence (MACD), Signal Line, Bollinger Bands, sine and cosine of the day of the year.

MACD identifies the direction and momentum of the index fund. It gives indications on whether we are in a bullish or bearish market. Signal Line gives us buy and sell signals for the index fund. When the price of the fund reaches the Upper Band it indicates overbought conditions and when the price of the index fund reaches the Lower Band it indicates an oversold condition. The width of the band being high indicates high volatility and the width reduces indicates low volatility.

Anamoly Detection

This module ingests the preprocessed index fund data and analyses it for potential future anamolies. A

possible anamoly includes a situation of a crash. It provides a score out of 50 for each index fund.

The anomaly detection module consists of two submodules: Rule-Based Anomaly Detection and Unsupervised Statistical Anomaly Detection. Together, these submodules help identify unusual patterns in stock prices before proceeding with forecasting.

Rule-Based Anomaly Detection

This submodule applies traditional statistical methods to detect anomalies in stock price movements. It uses three different techniques:

•Bollinger Bands: Identifies anomalies by checking if stock prices fall outside the upper or lower Bollinger Bands, which are calculated using a 20-day moving average and standard deviation.

•Z-Score Method: Measures how far the stock price deviates from its mean in terms of standard deviations, marking extreme deviations as anomalies.

•Interquartile Range (IQR) Method: Defines outliers as prices that fall outside the normal range determined by the first and third quartiles.

The final rule-based anomaly score is computed by aggregating the anomalies detected by these three methods. This score is then scaled to a 0-20 range to provide an overall anomaly measure for the dataset.

Unsupervised Statistical Anomaly Detection

This submodule leverages machine learning-based unsupervised anomaly detection techniques to identify outliers without requiring labeled data. The following methods are applied:

•Local Outlier Factor (LOF): Detects anomalies by comparing the local density of a point with its neighbors, identifying instances with significantly lower density.

•Histogram-Based Outlier Score (HBOS): Uses distribution-based modeling to assign anomaly scores based on deviations from expected statistical patterns.

•Isolation Forest: A tree-based ensemble method that isolates anomalies by partitioning the dataset recursively.

The anomaly scores from these models are aggregated and normalized to a 0-30 scale using an exponential function. This ensures that the final statistical anomaly score reflects the severity of detected anomalies proportionally.

Overall Anomaly Score

The final anomaly detection score is determined by combining the outputs of both submodules. This ensures that anomalies detected using traditional statistical techniques are complemented by advanced machine learning models, leading to a more robust identification process. The resulting score provides a quantifiable measure of anomaly severity, which can be used as a pre-processing step before stock price forecasting.

Stacked Gated Recurrent Unit (GRU)

A stacked GRU is a part of the forecast engine and it is one of the models which contributes to the forecasting process.

It consists of 2 GRU layers , each with 256 units. The model is designed for stock price forecasting by capturing temporal dependencies in sequential stock marketdata. The GRU layer processes historical stock prices to learn patterns over time, with stacked layers improving the model's ability to capture complex trends.The first layer captures long term trends while the second layer captures short term trends. A dropout layer prevents overfitting by randomly deactivating neurons, ensuring better generalization. The fully connected layer maps the learned temporal features to a single output, predicting the next stock price. The initial hidden state provides a starting point for the GRU, ensuring consistent training. During inference, the model takes a sequence of past stock prices and predicts the next closing price, which can be fed recursively to generate multi-step forecasts.

The Gated Recurrent Unit (GRU) consists of two main gates: the Reset Gate and the Update Gate, which control the flow of information.

Reset Gate

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{3.1}$$

Update Gate

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{3.2}$$

Candidate Hidden State

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{3.3}$$

Final Hidden State Update

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{3.4}$$

Activation Functions

Sigmoid activation function:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3.5}$$

Tanh activation function:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3.6}$$

Holt Winters Triple Exponential Smoothning

The Holt-Winters Exponential Smoothing model is a time series forecasting method that extends exponential smoothing to account for trend and seasonality in the data.

The Holt-Winters model consists of three main components:

Level (Base Value) : Represents the baseline or the smoothed value of the time series at a given point in time.

Trend : Captures the upward or downward movement in the data over time. It can be additive (constant change per unit time) or multiplicative (percentage change per unit time).

Seasonality : Accounts for repeating patterns in the data over a fixed period, such as daily, weekly, or monthly fluctuations in stock prices. It can also be additive or multiplicative.

The model applies a smoothning process which provides more weightage to more recent datapoints. It also predicts the trend. One key distinguishing feature of this model is its ability to see seasonal cycles in the time series data. Once the model is trained it can be used to extrapolate future values

The model is tuned to consider only business days for and forward filling is used to fill in values on weekends and holidays

The model is configured with an additive trend and additive seasonality, making it suitable for stock prices that exhibit linear growth patterns and seasonal fluctuations of a constant magnitude. This prevents explosion of values which is common in cases of exponential growth

To determine the optimal seasonal period, the Autocorrelation Function (ACF) is utilized. The ACF measures how the stock's closing price at time t is correlated with past values at different lags. The seasonal period is chosen as the first lag where autocorrelation exceeds 0.5, ensuring that the model captures periodic price movements. If no strong seasonal pattern is found, a default period of 21 business days (approximately one month) is used. The model is then optimized using automatic parameter tuning, and a bias correction is applied to improve forecasting accuracy. The predictions are generated for 60 business days and aligned with actual trading days to maintain consistency with stock market behavior.

Autocorrelation Function (ACF) and Its Formula

The Autocorrelation Function (ACF) measures the correlation between a time series and its lagged values. It helps identify repeating patterns, trends, and seasonality within the data. The ACF value for a given lag k quantifies how similar the time series is to itself after shifting by k time steps.

Formula for ACF The ACF at lag k is computed as:

$$\rho_k = \frac{\sum_{t=1}^{N-k}(y_t - \bar{y})(y_{t+k} - \bar{y})}{\sum_{t=1}^{N}(y_t - \bar{y})^2} \tag{3.7}$$

where:

- ρ_k is the autocorrelation coefficient at lag k.
- y_t represents the time series value at time t.
- $\bar{y}$ is the mean of the time series.
- N is the total number of observations.

Usage in Seasonal Period Detection In the implementation of the Holt-Winters model, ACF is used to determine the optimal seasonal period by computing autocorrelation values for multiple lags. The highest significant peak in the ACF plot indicates the dominant seasonality in the data, which is then used as the seasonal period for the model.

Combination

To combine the predictions from our forecasting models we propose to use the following methods in an exploratory manner

Weighted Averaging : This method assigns different weights to each model's predictions based on performance metrics.

Meta Learning (Regression Stacking) : A meta-model is trained using the predictions from multiple models as inputs to improve overall accuracy.

Quantile Blending : Combines predictions by merging specific quantiles from multiple models, enhancing robustness against market fluctuations and extreme values.

Min Max Selector : This combination method selects between multiple forecasted values at each point in time by looking at which forecasted value has the least error (RMSE, MAPE) at that time. It helps us switch between models based on their performance over time periods.

Miscellaneous explored models

TabNet

TabNet is a deep learning architecture designed specifically for tabular data, utilizing sequential attention to select relevant features at each decision step. Unlike traditional tree-based models, TabNet leverages an interpretable attention mechanism to focus on different subsets of features dynamically, improving learning efficiency and generalization. The model is composed of independent and shared feature transformers, enabling both representation learning and direct decision-making.

TabM

TabM is a neural network-based architecture designed for tabular data modeling, incorporating deep learning techniques with structured enhancements for improved performance. It employs a multi-layer perceptron (MLP) as its core network, consisting of multiple hidden layers with an increased number of neurons to capture complex relationships within the data. The model utilizes the GELU activation function, which provides smoother activation transitions, improving gradient flow and learning stability. Dropout regularization is applied to prevent overfitting and enhance generalization.

To further improve predictive performance, TabM integrates an efficient ensemble mechanism that stabilizes learning through multiple model variations. This ensemble consists of eight sub-models, leveraging input and output scaling techniques along with bias adjustments to refine predictions. The ensemble approach ensures robustness against noise and enhances the model's ability to generalize across varying market conditions. With its structured architecture and ensembling capabilities, TabM serves as a powerful deep learning model for time-series regression tasks like stock price forecasting.

Experimental Results

Dataset Description

The dataset utilized for this study is comprised of historical index fund price data pertaining to most popular indices belonging to the National Stock Exchange(NSE).These indices include the Nifty 50,Nifty IT, Nifty Auto, Nifty Metal, Nifty Pharma, Nifty FMCG(Fast Moving Consumer Goods), Nifty finserv, Nifty Bank, Nifty realty.Each index mentioned contains equity allotment for top companies in each industry.

This dataset encompasses the Open(opening price for a day), Close(closing price for the day), High(highest price reached by index in a day), Low(lowest price reached by the index in a day) along with the full date in which these parameters were noted.

The data spans an extensive period from 1990 to 2024. And thereby providing a comprehensive view of the market trends during this timeframe which can be used for prediction.

The dataset incorporates a variety of attributes that are critical for analyzing the performance of the index fund, which include:

•Date: This attribute signifies the specific date of the recorded data point, serving as a temporal marker for all subsequent price information.

•Open Price: This denotes the price at which the index opened for trading on a given day. The open price is essential as it reflects the initial market sentiment and conditions at the start of the trading session.

•Close Price: This indicates the price at which the index closed at the end of trading for that day. The close price is crucial for determining daily price fluctuations and is often used as a benchmark for evaluating the index's performance over time.

•High Price: This refers to the highest price reached during the trading day. It provides insight into the peak trading activity and market enthusiasm throughout the session.

•Low Price: Conversely, this indicates the lowest price reached during the trading day, which can help in assessing the level of volatility and the extent of downward price movements.

The dataset is cleaned and preprocessed to remove any anomalies or missing values, ensuring that the model is trained on high-quality data. The final dataset provides a robust basis for analysis and prediction.

Performance Analysis

Stacked GRU

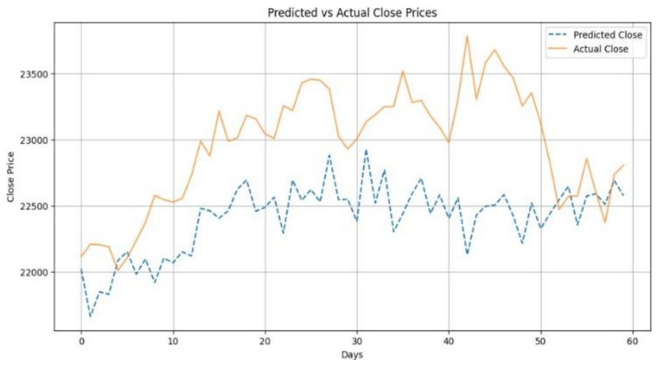


FIGURE 4.1: Stacked GRU forecast

Metric	Value
RMSE	652.386
MAPE (%)	2.42
RMSPE (%)	2.80

TABLE 4.1: Forecasting Errors for Stacked GRU Model

Forecast explained

The model consists of multiple GRU layers(2) with 256 hidden units, which allows it to capture both short-term and long-term dependencies in stock price movements. GRUs are well-suited for financial time series because they efficiently retain past information while avoiding vanishing gradient issues common in deep RNNs. Presence of Dropout prevents overfitting

Instead of using traditional MSELoss, the model employs Log-Cosh Loss, which is less sensitive to outliers. This is crucial for stock market prediction, where large spikes or dips could distort loss calculations if squared errors (MSE) were used. This function behaves like MSE for small errors and behaves like MAE for larger errorsas shown in fig 4.1 and table 4.1.

The model utilizes a Step Learning Rate Scheduler, which reduces the learning rate by half every 50 epochs. This helps stabilize training by allowing larger updates initially and smaller updates later, leading to a smoother convergence. With a patience counter (20 epochs), training stops early if validation loss does not improve for 20 consecutive epochs. This avoids excessive training that could lead to poor generalization.

A small Gaussian noise component (2% of the predicted value) is added to each prediction.This prevents the forecast from becoming an unrealistic, overly smooth curve by introducing some randomness, which financial markets inherently exhibit.This also nudges the model to actively forecast different values. Without this the rolling window will soon be filled with similar values.

Despite crashes like the crash of 2008 , 2020 the market not only recovers from the fall but also exceeds its baseline to set new highs. This is a recurring pattern. There is a bias in the market to favour higher values. Mispredicting higher values is more costly than mispredicting lower values on the longer run. Thus we use temperature scaling to the forecast which scales predictions by a small factor to favour higher values

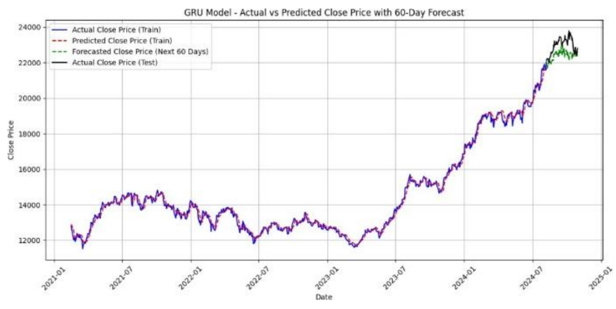


FIGURE 4.2: Stacked GRU training + forecast

The fig 4.2 shows us how well the model predicted and forecasted. The model has successfully forecasting a curved trend towards end of 2024 (shown in green).

Holt Winters

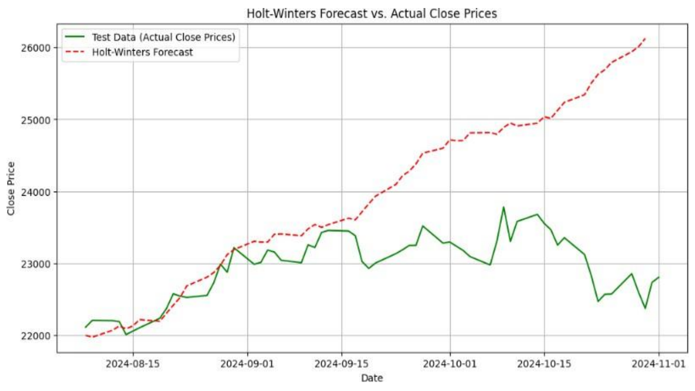


FIGURE 4.3: Holt Winters forecast

Metric	Value
RMSE	1433.500
MAPE (%)	4.37
RMSPE (%)	6.27

TABLE 4.2: Holt Winters metrics

Inference : Fig 4.3 and table 4.2shows the holt winters model forecasts the actual trend correctly for upto 30 days. After this the forecast starts deviating. This is because of the fundamental nature of the holt winters model which tries to fit a recurring seasonal curve onto the datapoints.The current train duration is from 2021 and extends into 2024. This is exclude anamolous datapoints which occured during the 2008 financial crisis and during the 2020 COVID pandemic.

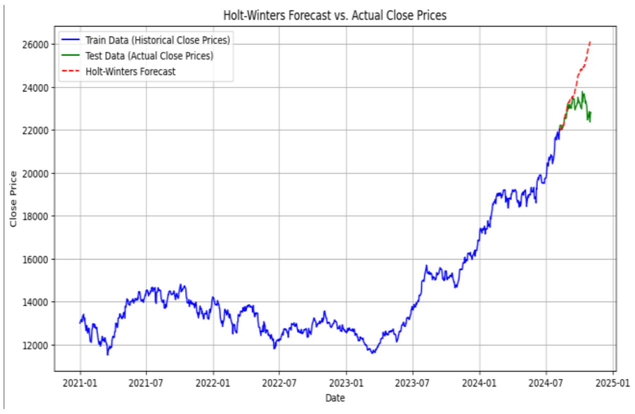
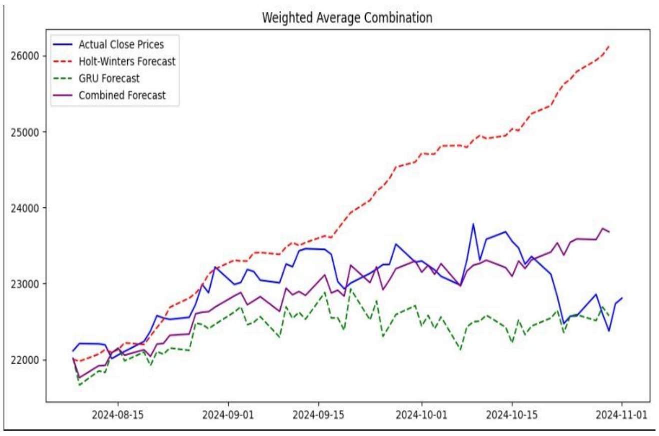


FIGURE 4.4: Holt Winters train + forecast

Fig 4.4 shows us the forecast along with training data . Main focus is on how accurate the model is for the first 30 days of forecast.

Combination of Forecasts

The methods proposed in the previous module for combining forecasts are experimented with and their results along with inference are presented.



Weighted Averaging

FIGURE 4.5: Weighted average combination

Metric	Value
RMSE	460.925
MAPE (%)	1.57
RMSPE (%)	0.02

TABLE 4.3: Weighted average combination metrics

Inference : The fig 4.5 and table 4.3 shows the Weighted Averaging involves assigning a weight that is inverse to the RMSE value of that model, This provides a steady forecast that is good over the forecasting duration.

Min Max Selector

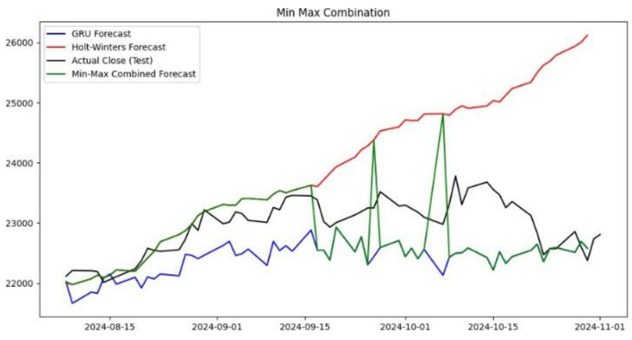


FIGURE 4.6: Min max combination

Metric	Value
RMSE	511.642
MAPE (%)	1.645
RMSPE (%)	0.022

TABLE 4.4: Min max combination metrics

Inference : Fig 4.6 and table 4.4 shows the min max selector selects that model's prediction which has lesser absolute error at each time instant. However this method leads to a lot of sudden jerks in the prediction and not ideal for forecasting condition

Quantile Blending

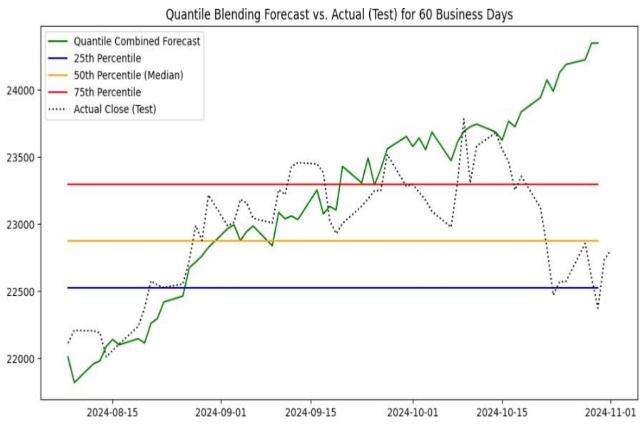


FIGURE 4.7: Quatile blending combination

Metric	Value
RMSE	635.947
MAPE (%)	1.89
RMSPE (%)	0.02

TABLE 4.5: Quantile blending combination metrics

Inference : Fig 4.7 and table 4.5 shows the methods blends the quantile values for 25th, 50th and 75th percentiles based on weights allotted to each model based on their inverse RMSE.These quantile values of both models give us confidence intervals for our forecast. Combined forecast is generated taking median of predictions.

Regression Stacking

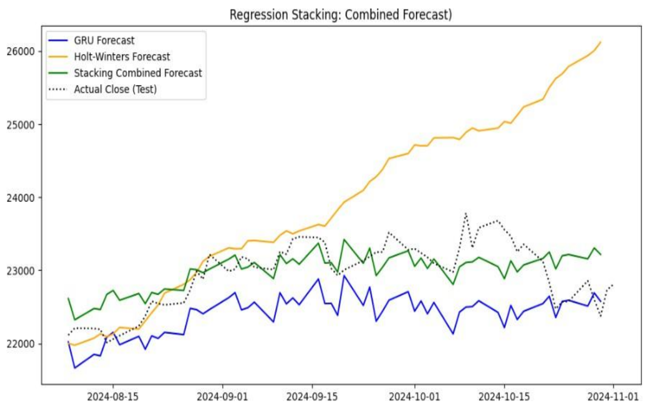


FIGURE 4.8: Regression Stacking with Linear Regression

Metric	Value

RMSE	398.46
MAPE (%)	1.415
RMSPE (%)	0.017

TABLE 4.6: metrics for Regression Stacking with Linear Regression

Inference : Fig 4.8 and table 4.6 shows the linear regression model is fit on the forecasts of models to provide the final forecast. A simple model like linear regression ensures combined forecasts are good and at the same time ensures overfitting does not happen.

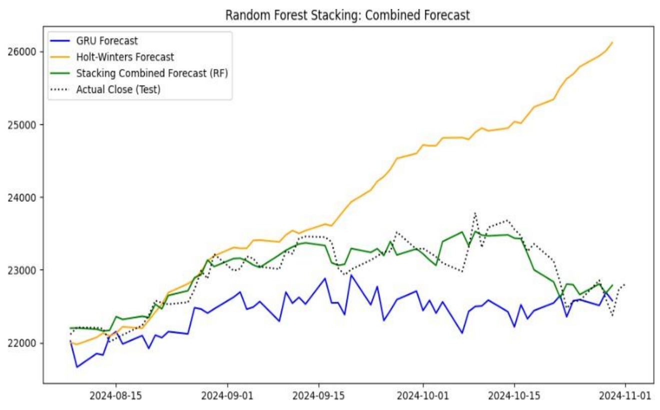


FIGURE 4.9: Regression Stacking with Random Forest

Metric	Value
RMSE	190.948
MAPE (%)	0.687
RMSPE (%)	0.008

TABLE 4.7: Metrics for Regression Stacking with Random Forest

Inference : Fig 4.9 and table 4.7 shows the Random Forest Model is fit on the forecasts of both models to improve predictions. This method provides exceptionally good metrics however there is the risk of overfitting.

Anomaly Detection

Category	Anomaly Score (out of 50)
nifty auto	14.288
nifty it	15.960
nifty bank	16.235

TABLE 4.8: Anomaly Scores for each fund

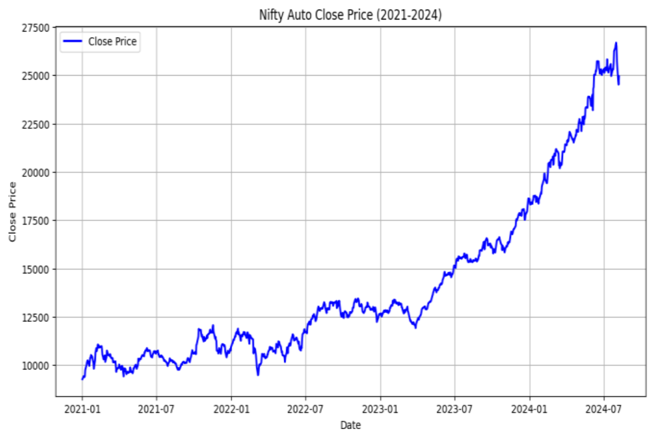


FIGURE 4.10: Nifty Auto Close prices

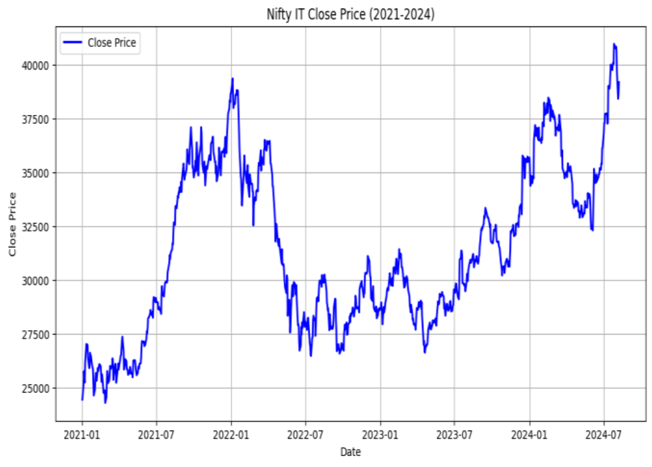


FIGURE 4.11: Nifty IT Close prices

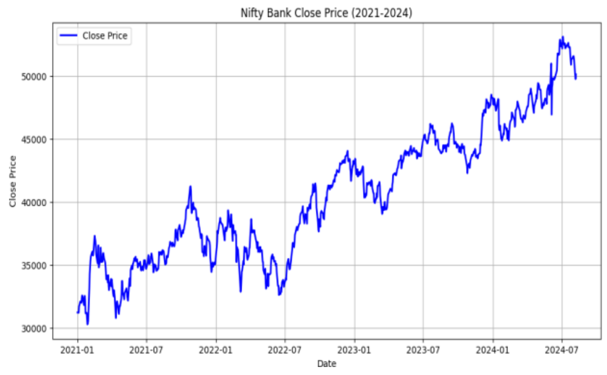


FIGURE 4.12: Nifty Bank Close prices

Inference:

The final anomaly score is computed by combining the outputs of both modules. The Bank index fund (16.235) has the highest score, likely due to its sharp fluctuations, which lead to more frequent anomalies across both modules. The IT index fund (15.960) follows closely, showing similar volatility patterns but slightly fewer extreme deviations. The Auto index fund (14.288) has the lowest score, as its steady upward trend results in fewer detected anomalies fig 4.10 fig 4.11 ,fig 4.12.

This methodology effectively captures price behavior variations, allowing for a data-driven understanding of index fund anomalies over time.

The anomaly detection module is designed to capture anomalies in a holistic manner by combining two complementary approaches: rule-based statistical methods and unsupervised machine learning models. Each component plays a unique role in identifying different types of anomalies, ensuring that no significant deviations in stock prices go unnoticed.

1.Rule-Based Statistical Methods (Module 1): This module applies predefined statistical thresholds to detect anomalies based on historical price movements. It focuses on fundamental statistical patterns and deviations, making it useful for detecting expected but extreme movements.

Bollinger Bands: Identifies short-term extreme fluctuations by marking data points where prices move beyond the upper or lower band. This method captures anomalies caused by sudden spikes or drops.

Z-Score Method: Detects global outliers by checking if a price deviates significantly (beyond 3 standard deviations) from the mean. It helps in identifying sharp breakouts from normal price distributions.

Interquartile Range (IQR): Captures persistent but unusual price behavior by flagging values outside the typical range. It is particularly effective in catching slow-forming anomalies where prices remain high or low for extended periods.

Strength of Module 1:

Uses well-established statistical thresholds to detect basic anomalies in price movements. Works well for identifying large, distinct deviations that are expected in time series data. Helps in early anomaly detection using standard financial metrics.

2.Unsupervised Machine Learning Models (Module 2) :

This module applies advanced outlier detection techniques that do not rely on predefined thresholds. Instead, these models learn from the data to detect hidden, complex, and multi-dimensional anomalies.

Local Outlier Factor (LOF): Detects local anomalies by comparing the density of data points in small regions. It is useful for catching anomalies that may not be extreme globally but stand out in a specific context. Example: A sudden price drop in an otherwise stable trend for a sector, which might not be detected by statistical methods.

Histogram-based Outlier Score (HBOS): Identifies global anomalies by looking at the overall frequency distribution of price movements. It assumes that common price movements follow a predictable pattern and flags deviations from it.

Example: A stock price suddenly moving into a range that has historically been rare.

Isolation Forest: Specializes in detecting more complex, non-linear anomalies by isolating rare patterns in price behavior using decision trees. It captures subtle anomalies that other methods may miss. Example: A gradual shift in price trends that does not immediately appear as an outlier but indicates potential manipulation or structural change.

Strength of Module 2:

Identifies anomalies that are non-obvious, local, or evolving over time. Works well in multi-dimensional space, considering not just the closing price but also moving averages, oscillators, and seasonality patterns. Adapts to changing market conditions, unlike rule-based methods that rely on fixed statistical thresholds.

How Both Modules Work Together :

By integrating Module 1 (Rule-Based) and Module 2 (Unsupervised Models), the system captures anomalies from both a statistical and a structural perspective:

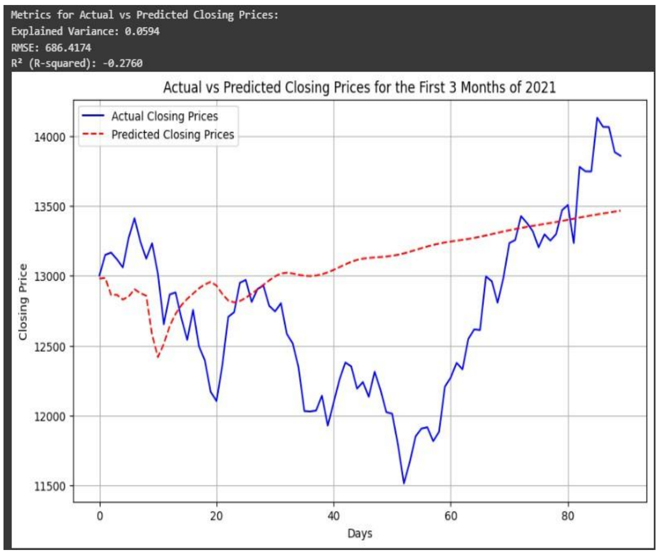
Module 1 detects broad, known deviations based on historical price distributions (e.g., Bollinger breakouts, large Z-score deviations).

Module 2 identifies hidden, local, and evolving anomalies (e.g., sudden density shifts detected by LOF, rare global movements found by HBOS, or complex patterns isolated by Isolation Forest).

The final anomaly score combines these results, balancing explainability (Module 1) with robustness (Module 2) to provide a comprehensive anomaly detection framework. This dual approach ensures that anomalies aren't just captured, but also meaningfully interpreted, making it useful for detecting potential market shifts, economic events, or sector-specific risks.

Results of Miscellaneous Models

Tabnet



```
Metrics for Actual vs Predicted Closing Prices:
Explained Variance: 0.0594
RMSE: 686.4174
R² (R-squared): -0.2760
```

FIGURE 4.13: Tabnet forecast

Thefig 4.13 shows TabNet model as shown above is able to forecast the trend for the first 10 to 15 days after which the predictions start to slowly increase almost like a straight line.

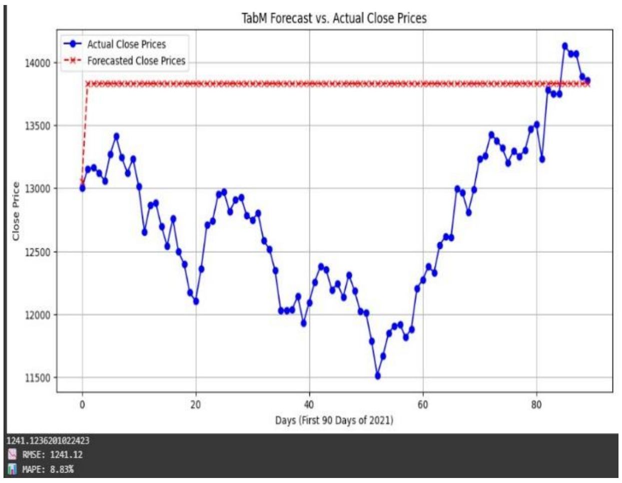
Reasons for Poor Forecast

TabNet due to its architecture is not most suitable for time series data as a Sequence to Sequence model. Despite feature engineering to include sine, cosine of the day of the year to include cyclic nature and including moving average features, the model is not able to infer patterns properly

High model complexity can also be a contributor to poor predictions as TabNet is a relatively complex architecture with attention-based feature selection and multiple decision steps. However, stock price movements are often highly stochastic (random), and an overly complex model can struggle to generalize well, leading to overfitting on historical data but failing on future unseen trends. Features like Number of decision steps, Feature reuse penalty, Independent layers per step, Shared layers across steps, Momentum factor for feature selection are some of the many parameters that need to be set during model initialisation which add to model complexity.

TabNet is a discriminative model, meaning it learns from existing data patterns. However, stock prices require forecasting beyond known data, which demands good extrapolation.

In conclusion even though Tabnet excels in tasks like dealing with tabular data it is not suitable for stock forecasting despite extensive feature engineering.



```
1241.1236201022423
RMSE: 1241.12
MAPE: 8.83%
```

TabM FIGURE 4.14 TabM

Reasons for Poor Forecast

Fig 4.14 TabM has an architecture based on MLP hence it treats each input independently, rather than modeling the sequential dependencies in stock prices. Unlike GRUs, LSTMs, or Transformer-based models, it lacks memory mechanisms to capture trends and momentum.

TabM does not learn feature importance dynamically. TabM uses a MLP based structure, where all features contribute equally based on learned weights.A good model should dynamically adjust feature importance over time. TabNet uses an attention-based feature selection mechanism, meaning it selects relevant features dynamically at each step.This allows better adaptability to changes in stock market conditions.

The ensemble combines multiple weak models to create a more stable prediction.This works well for classification and structured data but fails for stock prices, where volatility and trends matter.Each ensemble member learns slightly different patterns.When predictions are combined, the final output becomes an average of different models.This results in flat and unresponsive predictions as seen in your our graph.

REFERENCES

[1] A. M. Rather, "LSTM-based Deep Learning Model for Stock Prediction and Predictive Optimization Model," EURO Journal on Decision Processes, vol. 9, p. 100001, 2021, doi: 10.1016/j.ejdp.2021.100001.

[16] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, pp. 6679–6687, May 2021, doi: 10.1609/aaai.v35i8.16826.

[2] B. A. Abdelfattah, S. M. Darwish, and S. M. Elkaffas, "Enhancing the Prediction of Stock Market Movement Using Neutrosophic-Logic-Based Sentiment Analysis," Journal of Theoretical and Applied Electronic Commerce Research, vol. 19, no. 1, pp. 116–134, Jan. 2024, doi: 10.3390/jtaer19010007.

[3] M. M. Billah, A. Sultana, F. Bhuiyan, and M. G. Kaosar, "Stock price prediction: comparison of different moving average techniques using deep learning model," Neural Computing and Applications, vol. 36, no. 11, pp. 5861–5871, Jan. 2024, doi: 10.1007/s00521-023-09369-0.

[4] K. Chahuán-Jiménez, "Neural Network-Based Predictive Models for Stock Market Index Forecasting," Journal of Risk and Financial Management, vol. 17, no. 6, p. 242, Jun. 2024, doi: 10.3390/jrfm17060242.

[5] C. Wang, J. Ren, H. Liang, J. Gong, and B. Wang, "Conducting stock market index prediction via the localized spatial–temporal convolutional network," Computers and Electrical Engineering, vol. 108, p. 108687, May 2023, doi: 10.1016/j.compeleceng.2023.108687.

[6] C. R. Karthik, Raghunandan, B. Ashwath Rao, and N. V. Subba Reddy, "Forecasting variance of NiftyIT index with RNN and DNN," Journal of Physics: Conference Series, vol. 2161, no. 1, p. 012005, Jan. 2022, doi: 10.1088/1742-6596/2161/1/012005.

[7] B. Gülmez, "Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm," Expert Systems with Applications, vol. 227, p. 120346, Oct. 2023, doi: 10.1016/j.eswa.2023.120346.
[8] J. Wang, J. Liu, and W. Jiang, "An enhanced interval-valued decomposition integration model for stock price prediction based on comprehensive feature extraction and optimized deep learning," Expert Systems with Applications, vol. 243, p. 122891, Jun. 2024, doi: 10.1016/j.eswa.2023.122891.
[9] D. Kumar, P. K. Sarangi, and R. Verma, "A systematic review of stock market prediction using machine learning and statistical techniques," Materials Today: Proceedings, vol. 49, pp. 3187–3191, 2022, doi: 10.1016/j.matpr.2020.11.399.
[10] M. Lu and X. Xu, "Trnn: An Efficient Stock Price Prediction Neural Network Model Based on Recurrent Neural Network and Price-Volume Time Series," 2023, doi: 10.2139/ssrn.4481659.
[11] M. Mohan, P. C. Kishore Raja, P. Velmurugan, and A. Kulothungan, "Holt-Winters Algorithm to Predict the Stock Value Using Recurrent Neural Network," Intelligent Automation &amp; Soft Computing, vol. 35, no. 1, pp. 1151–1163, 2023, doi: 10.32604/iasc.2023.026255.
[12] A. Moghar and M. Hamiche, "Stock Market Prediction Using LSTM Recurrent Neural Network," Procedia Computer Science, vol. 170, pp. 1168–1173, 2020, doi: 10.1016/j.procs.2020.03.049.
[13] A. Q. Md et al., "Novel optimization approach for stock price forecasting using multi-layered sequential LSTM," Applied Soft Computing, vol. 134, p. 109830, Feb. 2023, doi: 10.1016/j.asoc.2022.109830.
[14] K. Rekha and M. Sabu, "A cooperative deep learning model for stock market prediction using deep autoencoder and sentiment analysis," PeerJ Computer Science, vol. 8, p. e1158, Nov. 2022, doi: 10.7717/peerj-cs.1158.
[15] Gorishniy, Y., Kotelnikov, A., & Babenko, A. (2024). Tabm: Advancing tabular deep learning with parameter-efficient ensembling. arXiv preprint arXiv:2410.24210.
[16] S. Ö. Arik and T. Pfister, "TabNet: Attentive Interpretable Tabular Learning," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 8, pp. 6679–6687, May 2021, doi: 10.1609/aaai.v35i8.16826.
.