On the Superiority of Data-Driven Combined Forecasts Based on Deep Learning Models

Abstract*—Forecasting financial asset prices and quantifying associated risks are critical challenges in computational finance. This paper presents an optimal forecast combination framework by integrating advanced time series models and machine learning techniques to enhance prediction accuracy and risk assessment. A novel data-driven risk measure ($DDRisk_{new}$) based on price differences is introduced, and $sign$ correlation is used to capture risk and mitigate the limitations of traditional risk metrics. Our approach incorporates state-of-the-art forecasting models, including ARIMA, Neural Network Autoregression, Long Short-Term Memory, XGBoost, and Random Forest, alongside two combination methods: equally weighted averages (FComp_SA) and data-driven optimal weighted averages (FComp_Weighted). Experimental results conducted on a dataset of ten sector-diverse stocks during the highly volatile COVID-19 period demonstrate the superior performance of FComp_Weighted in minimizing forecasting errors across multiple metrics (RMSE, MAE, MAPE). Moreover, the RMSE of asset price and $DDRisk_{new}$ forecasts using FComp_Weighted models is always lower than the RMSE obtained using the FComp_SA model. This study underscores the importance of combining forecasts and provides a robust, computationally efficient framework for predictive modeling in financial markets. These findings have implications for algorithmic trading, portfolio optimization, and risk management, paving the way for future applications in broader domains like energy and cryptocurrency markets.

***Index Terms*—Forecasting, Risk Measure, Neural Networks, Time Series, Machine Learning, Forecast Combinations**

I. Introduction

A constant in life is that we are always concerned with predicting or forecasting the future, whether it is determining if the roads will be safe tomorrow or if our investments will be appreciated. The prediction and forecasting of stock prices have become vital components of statistics, with forecasting methods continually evolving to provide an edge that creators can leverage to increase profits or advance research. The primary motivations for forecasting include generating profits for a company, identifying trends in a time series, gaining market insights, and understanding which financial assets are the most volatile (and thus, the riskiest). When estimating the risk of financial assets, a popular risk measure is volatility, defined as the standard deviation of log returns or simple returns, obtained by first modeling the variance. However, this measure is not highly efficient due to its higher asymptotic variance and the requirement of normality assumptions. Therefore, there is a need to derive new efficient estimators to capture the risk. In this paper, we propose a new risk measure based on price differences, and in our extensive preliminary study, we have observed and verified that the price differences are stationary with statistical tests during the study period.

Time series forecasting is a field of study that involves predicting future values of a variable over time based on its past values. In many real-world scenarios, data is collected at regular intervals, such as hourly, daily, monthly, or yearly, forming a sequence of observations ordered by time. This sequential nature of the data makes it a time series, and forecasting techniques aim to understand and predict the patterns inherent in these sequences. Forecasts of a given time series can be obtained using popular time series benchmark models such as the mean (average) model, naive model, seasonal naive model, and drift model. Even though they are simple and easy to implement with less computational burden, their performances are limited in practice as they fail to capture the complicated correlation structures in real-life data [1]. Thus, to obtain more accurate forecasts, more advanced time models have been taken into account [2].

The non-seasonal autoregressive integrated moving average (ARIMA) model is a very popular time series model in the literature, which has been extensively investigated in stock price predictions [3] and demand forecasting [4] extensively. Nevertheless, the efficacy of ARIMA models is constrained since they overlook the seasonality inherent in data and are incapable of integrating additional factors that influence the target time series. Seasonal ARIMA models can be used to model a wide range of seasonal data and to include other relevant information along with information from past data [5].

Deep learning models are becoming popular in time series forecasting due to their accuracy and capability of handling more complex real-world problems [6]. The feed-forward neural network is a popular way to approximate a given multivariate nonlinear function and a neural network can approximate any continuous function and translate inputs to desired output signals. One of the popular neural networks specifically designed to handle time series data is neural network autoregressive (NNAR) [5]. Also, it can be modified to consider some feature values [7], [8]. Even though NNAR is a very capable neural network for forecasting time series data, it only fits a neural network with a single hidden layer. Long short-term memory (LSTM) is another popular recurrent neural network model that is well-suited for sequential data. It captures the log term dependencies and is highly effective in accurate forecasting [9], [10]. A recently introduced deep learning model, extreme gradient boosting (XGBoost), demonstrated remarkable success in forecasting by combining multiple weak learners to enhance generalization and reduce overfitting [11]. An ensemble learning method based on decision trees known as random forest (RF) is another model for time series data, and in recent research work, they have been investigated with time series data in different fields [12]. Determining the number of hidden layers and neurons in the hidden layer that work optimally for all the tasks is not possible. It depends on various factors such as the complexity of the problem and data and the availability of computational resources. It is believed that increasing the number of layers and neurons can improve model performance and forecast accuracy. However, this can also lead to overfitting. Too many hidden layers and neurons may result in a well fit for training data and it may fail to generalize for unseen data. Also, more layers and neurons increase the complexity of the model and reduce interpretability.

Apart from the individual models, combine forecasts from individual models after another layer of improvement in forecast accuracy. Modern literature on forecast combinations started to grow with the work of [13], and since then, different approaches to obtain both point forecasts and probabilistic forecasts have been introduced and investigated. Among the different point forecast combinations, the simplest and most popular combination scheme is the equally weighted average/simple arithmetic average [14]. The median and mode of the forecasts are also considered simple ways of obtaining points forecasts [15]. While equal-weight combinations are simple to implement and perform well in many situations, the optimal weight approach offers significant advantages.

This study introduces a novel forecast combination framework integrating traditional time series and machine learning models to improve forecast accuracy. A new data-driven risk measure for asset price, $DDRisk_{new}$, is proposed considering price differences and `sign` correlation. The study uses state-of-the-art forecasting models (ARIMA, NNNAR, LSTM, XGBoost, and RF) and combined individual model forecasts through two approaches: simple averaging (FComp_SA) and an optimally weighted method (FComp_Weighted). Experimental results demonstrate that FComp_Weighted consistently outperforms individual models and simple averaging in reducing forecast errors with a dataset spanning the highly volatile COVID-19 period.

The remainder of the paper is organized as follows: Section II provides details of statistical and deep learning models, optimal combined forecasts, and a novel risk measure for asset prices. Section III describes experiments and results. Finally, Section IV provides concluding remarks.

II. Methodology

The methodology section of this paper provides a detailed description of statistical and deep learning models and optimal combined forecasts. Moreover, we propose a new data-driven risk measure to assess the risk associated with asset prices.

In this study, two traditional statistical time series models, Drift and ARIMA, are used. When the forecasts are simply the values of the last observation, the method is known as the Naive method. Drift, a variation of the Naive model, allows the forecasts to increase or decrease over time. A moving average (MA) model is a type of regression-like model that uses past forecast errors. When the forecast is obtained using a linear combination of past values of the variable, it is known as an autoregression (AR) model. By combining differencing with a moving average and an autoregression model, a non-seasonal ARIMA model can be obtained. More details about Drift and ARIMA can be found in [5].

A. Deep Learning Models

Recent advancements in deep learning have introduced more sophisticated architectures capable of capturing complex temporal dependencies and sequence patterns. These models offer enhanced capabilities for handling non-linearities and long-term dependencies in data. In this study, NNAR, LSTM, XGBoost/XGB, and RF models are used to obtain forecasts of price and a new risk measure. Details of these models are provided below.

Neural Network Autoregression (NNAR) Model: NNAR is a specialized variant of neural networks designed for time series forecasting. The model predicts future values based on past values. The notation $NNAR(p,P,k)_m$ indicates an NNAR model with a single hidden layer. Here, p is the number of lagged inputs, P is the AR order of the seasonal part, k is the number of nodes in the hidden layer, and m is the seasonal period. In NNAR, if we are predicting the value y_t at time t, the inputs might be the values $y_{t-1}, y_{t-2}, \ldots, y_{t-p}$ at $t-1, t-2, \ldots, t-p$, respectively. The R function `nnetar` fits a single-layer $NNAR(p,P,k)_m$ model and if not specified by the user, p and P are automatically selected (according to the Akaike Information Criterion (AIC)). In most cases, the user will define the number of neurons in the hidden layer. In contrast to other feed-forward, the number of neurons in the hidden layer is equal to $k = (p + P + 1)/2$ (rounded to the nearest integer) in `nnetar`. Moreover, `nnetar` runs 20 iterations and takes an average to give more accurate forecasts.

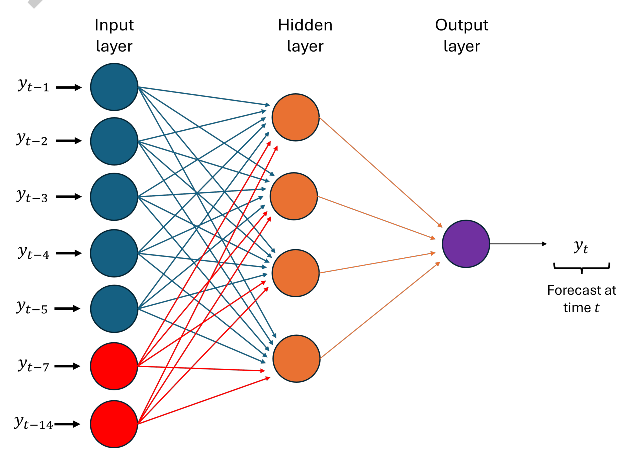

Fig. 1. Schematic diagram of $NNAR(5,2,4)_7$ Model.

Figure 1 denotes a $NNAR(5,2,4)_7$ model ($p = 5$, $P = 2$, $k = 4$, and $m = 7$). The input to each node in the hidden layer is combined using a weighted linear combination:

$$z_j = b_j + \sum_{i=1}^{5} w_{i,j} y_{t-i} + \sum_{i=1}^{2} w_{i,j} y_{t-7i},$$

where z_j is the j^{th} node in the hidden layer, $w_{i,j}$ are weight assigned using a learning algorithm for i^{th} inputs for the j^{th} node. Then, the results are modified using a non-linear function, $s(z) = \frac{1}{1+e^{-z}}$ to obtain output. The number of weights (b_j and $w_{i,j}$) to be estimated is 32 $(4 + 7 \times 4)$.

Long Short-Term Memory (LSTM) Model: LSTM is a specialized architecture of recurrent neural networks designed to address the vanishing gradient problem using memory cells [16]. LSTMs achieve this through the use of gates (input gate, forget gate, and output gate) that control the flow of information within the network. The input gate decides what new information to store in the cell's memory, and the forget gate decides what information to discard. The output gate determines the information to output from the cell. Thus, LSTM is well-suited for sequential data due to its ability to capture long-term dependencies through a memory cell mechanism, making it highly effective in time series forecasting. In this study, LSTM models are trained using the `keras` package, and two LSTM models are considered for the numerical analysis: LSTM1 (a single hidden layer) and LSTM2 (two hidden layers). The number of neurons in the hidden layer(s) is equal to the number of neurons in the NNAR model (denoted as k in NNAR). Details of the hyperparameters (optimizer, loss function, epochs, and batch size) are provided in the experimental results section. Additionally, the data frames are modified to include lagged values (denoted as p in NNAR) of the target series (price/new risk) when generating forecasts.

Extreme Gradient Boosting (XGBoost) Model: XGBoost is a powerful machine learning algorithm known for its efficiency, speed, and accuracy. It belongs to the family of boosting algorithms, which are ensemble learning techniques that combine the predictions of multiple weak learners (models that perform only slightly better than random guessing). Here in this study, we implement XGBoost using the R package `xgboost` [17] and it is referred to as XGB. The details of the hyperparameters (booster, objective, learning rate, maximum depth of trees, subsample ratio of training data, subsample ratio of columns, and number of rounds) of the XGB models are provided in section III.

Random Forest (RF) Model: RF is an ensemble learning method that combines multiple decision trees to achieve improved predictive performance and reduce overfitting [18]. Each tree in the forest is built on a bootstrapped sample of the training data, and at each node, a random subset of features is considered for splitting. This randomness introduces diversity among the trees, enhancing generalization. The model aggregates the predictions from all trees using majority voting (classification) or averaging (regression). RF is implemented using the R package `randomForest` and except for the hyperparameter, the number of simple models, or the number of decision trees, that are combined to create the final prediction (number of trees grown), the others are set to default values in this study.

B. Forecast Combination

An alternative approach to obtaining superior forecasts is through forecast combinations. The objective is to combine forecasts from different models to minimize large errors, especially when it is uncertain which method provides the most accurate forecasts. In this study, we have used simple and optimally weighted average forecast combination models, and the details of these models are provided below.

Let for the target variable y, $\hat{y}_{k,t+h|t}$ is the h-step ahead forecast $(h = 1, 2, \ldots, H)$ using model k $(k = 1, 2, \ldots, K)$ given the information up to time t is available. Then the forecast combination based on the optimal weight (FComb_Weighted) can be derived as:

$$\hat{y}^c_{t+h|t} = \sum_{k=1}^{K} w_k \; \hat{y}_{k,t+h|t},$$

where w_k $\left(\sum_{k=1}^{K} w_k = 1\right)$ are the weights associated with the individual models that need to be optimized. The simple average/equally weighted average forecast combination (FComb_SA) is a special case of FComb_Weighted when $w_k = 1/K$. In this work, forecasts from two possible individual models are used to obtain optimal combined forecasts (FComb_Weighted). Let w_1 and w_2 be the weights associated with forecasts using individual model 1 and model 2, respectively, and $w_1 + w_2 = 1$. For simplicity, we can write, $w_1 = w$ and $w_2 = (1 - w)$, and then the FComb_Weighted can be written as

$$\hat{y}^c_{t+h|t} \leftarrow w \; \hat{y}_{1,t+h|t} + (1-w) \; \hat{y}_{2,t+h|t}, h = 1, 2, \ldots, H,$$

where w is the weight to be optimized. In literature, different techniques have been used in weight optimization. In [19], random weights generated from a uniform distribution are used for the weighted weight-moving average of demands, and optimal weights are determined by minimizing the one-step ahead forecast error sum of squares of the demand.

Weight Optimization: The optimization algorithm can be used to select the optimal weight, in which the optimization criterion such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) or Root Mean Squared Error (RMSE) will be used. In this work, the optimal weight (w^*) associated with individual model 1 is obtained by minimizing the one-step ahead forecast error sum of squares (FESS),

$$\text{FESS} = \sum_{h=1}^{H} (y_{t+h} - \hat{y}^c_{t+h|t})^2,$$

where y_{t+h} is the observed value of y at time $t+h$. Some other techniques such as genetic algorithms can be used to determine the optimal weight and future studies may focus on these techniques.

Once the optimal weight for model 1 is determined, the optimal weight for model 2 can be calculated as $1 - w^*$.

Finally, the FComb_Weighted forecasts for the unobserved period are given by

$$\hat{y}^c_{t+h|t} = w^* \ \hat{y}_{1,t+h|t} + (1 - w^*) \ \hat{y}_{2,t+h|t}, \ h = 1, 2, \dots, H.$$

In practice, the observed period of the target variable can be identified as the training period. The steps for finding optimal weights and obtaining optimal combined forecasts are summarized in Algorithm 1.

Algorithm 1 Optimal Data-Driven Combined Forecasts

Require: Data: Observed value of target variable y_t at time t, and set of K individual forecasting models

Generate forecast of target variable, $\hat{y}_{k,t+h|t}$ at time $t+h, h = 1, 2, \dots, H$ using $k = 1, 2, \dots, K$ individual models.

for each pair of combination of individual models **do**

Generate R sequential values $w_r, r = 1, 2, \dots, R$, where $w_r = w$ is equally spaced in $(0, 1)$, such that $w_r = \frac{r}{R+1}$, for $r = 1, 2, \dots, R$.

  for $r \leftarrow 1, \dots, R$ **do**

    for $h \leftarrow 1, \dots, H$ **do**

    $w = w_r$

    $\hat{y}^c_{r,t+h|t} \leftarrow w \ \hat{y}_{1,t+h|t} + (1 - w) \ \hat{y}_{2,t+h|t}$

    end for

  $\text{FESS}_r \leftarrow \sum_{h=1}^{H} (y_{t+h} - \hat{y}^c_{r,t+h|t})^2$

  end for

Obtain optimal weight

  $w^*_r = w^* \leftarrow \arg\min_{w_r} FESS_r, \ r = 1, 2, \dots, R$

Compute the optimal combined forecast

  $\hat{y}^c_{t+h|t} = w^* \ \hat{y}_{1,t+h|t} + (1 - w^*) \ \hat{y}_{2,t+h|t}, \ h = 1, 2, \dots, H.$

Calculate RMSE for each combination of individual models

  $\text{RMSE} = \sqrt{\frac{\sum_{h=1}^{H} (y_{t+h} - \hat{y}^c_{t+h|t})^2}{H}}$

end for

Choose the best pair of combination that minimize the RMSE as the optimal forecast combination and corresponding $\hat{y}^c_{t+h|t}$ as optimal forecasts.

C. A New Risk Measure ($DDRisk_{new}$)

In finance, the stock prices, P_t at time t, are modeled as a geometric Brownian motion. Thus, it is a continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion with drift. The calculation of returns is fundamental to estimating risk in finance. Log returns, $r_t = \ln(P_t) - \ln(P_{t-1})$, are widely used due to their time additivity, compatibility with continuous-time stochastic models, and better approximation to normality, which simplifies statistical analysis. Simple returns, $R_t = (P_t - P_{t-1})/P_{t-1}$, on the other hand, are intuitive for measuring percentage changes in price and are often used for assessing portfolio performance. While log returns are preferred for modeling and volatility estimation of individual assets, simple returns are commonly applied when analyzing the risk or returns of portfolios.

Volatility, denoted as the standard deviation of log returns, serves as the primary metric in the majority of financial analyses. Typically log returns are stationary, and when returns are normally distributed, it is an adequate risk measure. In traditional approaches (generalized autoregressive conditional heteroskedasticity (GARCH) models), volatility is calculated by first modeling the conditional variance of log returns and then taking the square root of conditional variance forecasts. Research has revealed that log returns often deviate from the normal distribution, with the majority exhibiting a t-distribution characterized by heavy tails in most cases [20]. Moreover, we can show that the asymptotic variance of the sample standard deviation depends on excess kurtosis. For heavy-tailed distributions, excess kurtosis is very large leading to very large asymptotic variance. Based on the theory of combined estimating functions, [20] proposed a direct estimate for volatility.

For any random variable Y with cumulative distribution function (CDF) $F()$, mean μ and finite variance σ^2, one can define the sign correlation as

$$\rho = Corr(Y - \mu, sgn(Y - \mu)) = \frac{E|Y - \mu|}{2\sigma\sqrt{F(\mu)(1 - F(\mu))}},$$

where $F(\mu)$ is the CDF evaluated at the mean μ. For symmetric distribution with finite variance, ρ turns out to be $E|Y - \mu|/\sigma$.

In this study, we select stocks from 10 diverse sectors based on market capitalization. Based on our empirical analysis, it has been observed that the fitted ARIMA models for all stock prices have first differences, indicating that the price difference ($d_t = P_t - P_{t-1}$) is a stationary series. Table I shows the results of 10 stocks (one stock from each sector). Moreover, we have used available statistical tests, such as the Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test to assess the stationarity of time series data. While the ADF test is designed to detect the presence of a unit root, the KPSS test checks for stationarity around a deterministic trend. These tests complement each other, and using both to draw robust conclusions about the stationarity of time series data is common. Thus, we use the price difference (because of the stationary behaviors) to define a new risk measure that we call Data-Driven New Risk ($DDRisk_{new}$) as

$$DDRisk_{new} = \frac{|d - E[d]|}{\rho},$$

where $\rho = Corr(d - E(d), sgn(d - E(d)))$. When obtaining forecasts of the new risk measure using different models, it can be improved with combined forecasts. The steps for obtaining estimates of new risk measures and optimal combined forecasts are summarized in Algorithm 2.

D. Forecast Accuracy

In order to evaluate the accuracy of the forecasting models, we use three commonly employed performance metrics that provide insight into the prediction errors and the overall performance of the models. These are MAE $=$

Algorithm 2 Data-Driven Risk Measure and Combined Forecasts

Require: Data: Adjusted closing price of stocks P_t at time t, and set of K individual forecasting models.

1: Calculate the price difference, $d_t \leftarrow P_t - P_{t-1}$
2: Calculate sign correlation, $\hat{\rho} = Corr(d_t - \bar{d}, sign(d_t - \bar{d}))$
3: Calculate new risk measure, $D_t \leftarrow \frac{|d_t - \bar{d}|}{\hat{\rho}}$
4: Follow Algorithm 1 using new risk measure D_t to select the optimal forecast combination and corresponding $\hat{D}^c_{t+h|t}$ as optimal risk forecasts.

$\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$, RMSE $= \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$, and MAPE $= \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100$.

III. Experimental Results

In this work, the adjusted closing prices were downloaded from Yahoo Finance for the analysis. According to Yahoo Finance, the stock market comprises 11 sectors (Technology, Financial Services, Consumer Cyclical, Healthcare, Industrials, Communication Services, Consumer Defensive, Energy, Real Estate, Basic Materials, and Utilities), each with distinct characteristics and features. Each sector includes a grouping of industries represented by all companies in those industries that trade on the stock market. The study considered 10 stocks (Apple Inc. (AAPL), Berkshire Hathaway Inc. (BRK-B), Amazon.com, Inc. (AMZN), Eli Lilly and Company (LLY), GE Aerospace (GE), Walmart Inc. (WMT), Texas Gulf Energy, Incorporated (TXGE), Prologis, Inc. (PLD), Linde plc (LIN), NextEra Energy, Inc. (NEE)) covering each sector for analysis (omit communication services sector). Stocks are chosen with the highest market capital. The study period spans from March 11, 2020, to May 5, 2023, and is chosen based on the global outbreak of the coronavirus disease 2019 (COVID-19) pandemic, as identified by the World Health Organization (WHO). This period is characterized by significant volatility for most financial assets, and the study specifically examines data within this volatile timeframe to compare the performance of the underlying models. Moreover, we have investigated the idea of structural breakpoints for stock prices in our preliminary analysis. Following structural breakpoints, we have chosen the study period without significant structural breaks. In future studies, analysis will be conducted for a longer study span with structural breakpoints, and it is not in the scope of this study. While stocks from each sector were investigated, not all were included in the experimental results. The experimental results section consists of two subsections. In the first subsection, the study investigates price forecasts, and forecasts of the new risk measure are investigated in the second subsection.

A. Price Forecasts

Daily adjusted closing prices of the 10 selected stocks are considered in price forecasts. Initially, the ARIMA models were fitted for all stocks, and results are shown in Table I. It can be seen that the ARIMA models for all stock prices have first differences, indicating that the stock prices are non-stationary during the study period and the price difference ($d_t = P_t - P_{t-1}$) makes the series a stationary series. Moreover, since $P = 0$ in all NNAR models, non-seasonal models are suggested for stock prices.

TABLE I
Model Parameters of $ARIMA(p, d, q)$ and $NNAR(p, P, k)$ for Stock Price

Stock	$ARIMA(p, d, q)$	$NNAR(p, P, k)$
Apple Inc. (AAPL)	(2,1,3)	(1,0,1)
Berkshire Hathaway Inc (BRK-B)	(3,1,3)	(8,0,4)
Amazon.com Inc. (AMZN)	(1,1,1)	(8,0,4)
Eli Lilly and Company (LLY)	(4,1,0)	(3,0,2)
GE Aerospace (GE)	(1,1,1)	(1,0,1)
Walmart Inc. (WMT)	(4,1,0)	(5,0,3)
Texas Gulf Energy (TXGE)	(4,1,1)	(4,0,2)
Prologis, Inc (PLD)	(2,1,2)	(8,0,4)
Linde plc (LIN)	(2,1,2)	(1,0,1)
NextEra Energy Inc. (NEE)	(1,1,1)	(8,0,4)

Next, the study investigates asset price forecasts using different models. Price forecasts are important because they can be used directly in pair trading applications and trading strategies based on Bollinger Bands [21], enabling traders to identify profitable opportunities by exploiting pricing inefficiencies between assets. Accurate forecasts also play a crucial role in portfolio optimization, risk management, and strategic investment planning by helping investors anticipate market movements. Furthermore, they are integral to algorithmic trading systems [22], [23], which rely on predictive models to execute trades with precision and speed. Effective forecasting minimizes financial risk, enhances decision-making, and provides a competitive edge in increasingly volatile and complex financial markets. Thus, applications of price forecasts will be further investigated in future work with the novel optimal data-driven combined forecasts approach.

The study chooses five stocks (AAPL, LLY, WMT, PLD, and NEE) from five major sectors (technology, healthcare, consumer defensive, real estate, and utilities) for further analysis. However, the results can easily be extended to other stocks in other sectors.

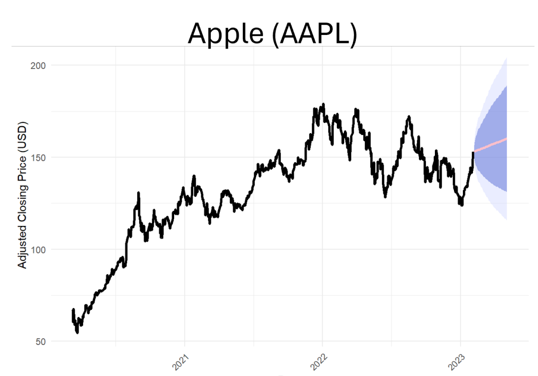

Fig. 2. Interval Forecasts of Apple Adjusted Closing Price using Drift Model.

Figure 2 shows the 80% (dark blue) and 95% (light blue) Bootstrap confidence intervals for price forecasts of AAPL. When the normality assumption for the innovation residuals is not valid, bootstrapping can be used as an alternative approach. This approach assumes that the residuals are uncorrelated with almost constant variance. Once we have the H forecasts $\hat{y}_{t+1}, \ldots, \hat{y}_{t+H}$, assuming future errors and past errors are similar, H errors are sampled from the collection of errors from the training data. Point forecasts added with sampled errors consist of a sequence of simulated future observations. By doing this repeatedly for s times, s realizations of the future observations of the target y_t are obtained. Then, percentiles (e.g., the 2.5th percentile and the 97.5th percentile for a 95% prediction interval (PI)) of the simulated future observations at each prediction horizon are calculated to compute the PIs of the target variable. The resulting PIs are called bootstrap PIs. Bootstrap intervals are important when forecasting stock prices because they provide a robust way to quantify the uncertainty associated with the forecast.

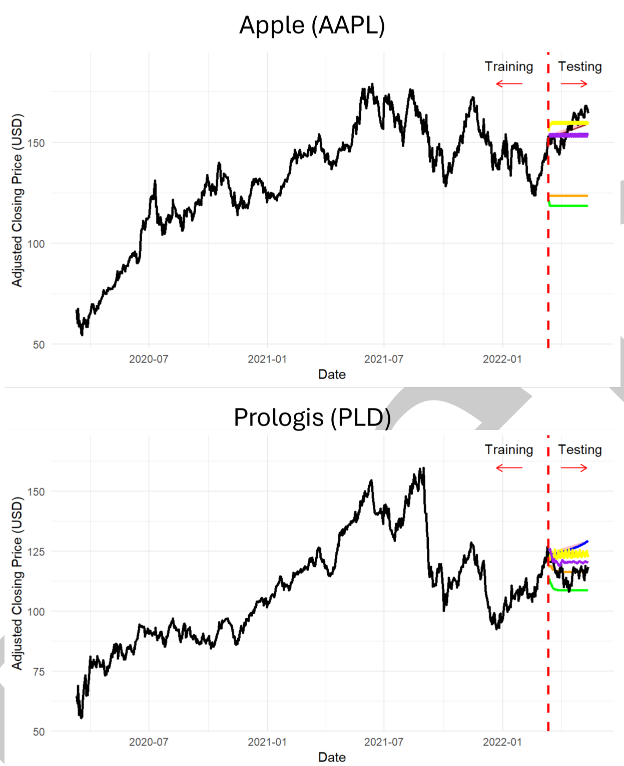

Fig. 3. Adjusted Closing Price Forecasts of AAPL (top) and PLD (bottom) using Drift (Pink), ARIMA (Brown), NNAR (Blue), LSTM1 (Green), XGB (Purple), LSTM2 (Orange), and RF (Yellow).

Data from March 11, 2020, to February 5, 2023, are considered training data, and the last three months of data are used as test data. We employ a range of forecasting models, including Drift, ARIMA, NNAR, LSTM (single-layer and multi-layer), XGBoost (XGB), and RF to forecast asset prices. To ensure fair comparisons and reproducibility, the number of neurons in the hidden layer of neural network-based models (LSTM1 and LSTM2) is determined by the optimal configuration suggested by the NNAR model. All LSTM models are trained using 20 epochs, 30 batch size, mean square error loss function, and the Adam optimizer to ensure consistent training conditions across experiments. The XGB model is fitted with the "gbtree" booster, "reg:squarederror" objective, a learning rate of 0.1, 6 maximum depth of trees, 0.8 subsample ratio of training data, 0.8 subsample ratio of columns, and 100 number of rounds. For the RF model, we use 500 trees, with the number of variables tried at each split determined automatically by the `randomForest` package. This approach guarantees reproducible results by standardizing configurations across models, thus providing a robust benchmark for model evaluation. For assets AAPL and PLD observed adjusted closing prices (black line) during the training and testing period along with forecasts using Drift, ARIMA, NNAR, LSTM1, XGB, LSTM2, and RF are shown in Figure 3. The vertical red dashed line marks the division between the training and testing datasets, highlighting the models' ability to forecast trends beyond the observed data in the training set. It can be observed from the figure that LSTM1 and LSTM2 underestimate the price forecasts for AAPL and fail to capture the increasing trend during the testing period. In contrast, both models (LSTM1 and LSTM2) successfully capture the decreasing trend in PLD prices during the testing period. Except for XGB, all other models forecast an increase in the price.

In Table II, we report RMSE for forecasts from individual models: Drift, ARIMA, NNAR, LSTM1, XGB, LSTM2, and RF, alongside two forecast combination approaches: FComp_SA and FComp_Weighted. The FComp_SA reports the best RMSE for each asset using a simple average combination of forecasts (equal-weighted model) and FComp_Weighted reflects the best RMSE achieved using a weighted average combination of forecasts (optimal weighted model). For both FComp_SA and FComp_Weighted, the best-performing combinations for each asset are derived from all possible combinations of the pairs of individual models, and all possible combinations are provided in Table III. Moreover, MAE and MAPE for price forecasts from individual models, FComp_SA, and FComp_Weighted are reported in Table VI and VII in the Appendix. These tables provide additional insight into the performance of the models across different error metrics. For instance, while RMSE captures the quadratic mean of forecast errors, MAE focuses on the absolute magnitude of errors, and MAPE normalizes these errors relative to the actual values, offering a percentage-based interpretation. This side-by-side comparison highlights the variability in forecast accuracy across metrics and the relative consistency of combination models, particularly FComp_Weighted, in minimizing these errors.

The bold numbers in Table II highlight the minimum RMSE for each asset. A comparison of the individual models shows that the best-performing model varies across assets, underscoring the heterogeneity in model performance. For AAPL, LLY, and WMT, the Drift and ARIMA models show superior performance. For NEE, even though Drift is not the best model, its performance is very close to that of the NNAR model.

TABLE II
RMSE OF PRICE FORECAST COMPARISON OF INDIVIDUAL AND BEST COMBINATION MODELS

Model	AAPL	LLY	WMT	PLD	NEE
Drift	5.254	**24.934**	**1.606**	11.168	1.921
ARIMA	5.289	24.964	1.994	8.601	2.401
NNAR	7.250	29.592	2.430	10.484	1.775
LSTM1	38.564	110.038	3.516	7.588	2.219
XGB	7.608	29.780	2.127	5.511	4.394
LSTM2	33.579	114.018	3.239	3.075	2.170
RF	7.596	29.582	2.130	8.259	1.782
FComp_SA	5.259	24.948	1.791	3.082	**1.666**
FComp_Weighted	**5.253**	**24.934**	1.609	**2.856**	**1.666**

The LSTM2 model is the best model for PLD price forecasts, while Drift has the highest RMSE for PLD in contrast to other stocks. When investigating combination models, under FComp_SA, the best model combinations are as follows: for AAPL, LLY, and WMT, Drift and ARIMA; for PLD, LSTM1 and XGB; and for NEE, NNAR and RF. The same combinations are observed for FComp_Weighted. However, RMSE values with FComp_Weighted are always smaller than or equal to those with FComp_SA for all stocks. Thus, when comparing forecast combination approaches, FComp_Weighted generally demonstrates a slight advantage over FComp_SA in terms of RMSE, indicating that a weighted average often provides a more accurate forecast than a simple average. Among all individual and combination models, except for WMT, the FComp_Weighted model has the lowest RMSE, indicating its superior performance. For WMT, the best model for price forecasting is Drift, with an RMSE of 1.606, while the RMSE for WMT with the FComp_Weighted model is 1.609. Thus, the improvement for WMT is only marginal. For AAPL, the FComp_Weighted model demonstrates the lowest RMSE of 5.253, marginally outperforming the Drift model with an RMSE of 5.254. Similarly, for LLY, the FComp_Weighted and Drift models have the same RMSE of 24.934, showing no significant difference between these two models for this stock. In the case of PLD, the FComp_Weighted model achieves the lowest RMSE of 2.856, followed by the LSTM2 and FComp_SA models with an RMSE of 3.075, and 3.082, respectively. In the PLD case, the improvement of RMSE with FComp_Weighted is significant compared to other assets and it was only possible due to optimal weight assignments on forecasts. For NEE, the FComp_Weighted model and the FComp_SA model are the top performers, both yielding an RMSE of 1.666. This result highlights the comparable accuracy of FComp_Weighted and FComp_SA for this stock. Overall, the FComp_Weighted model consistently achieves or ties for the lowest RMSE across all stocks, further solidifying its effectiveness as a robust forecasting approach.

Table III reports the RMSE for AAPL and PLD price forecasts using all possible combinations under FComp_SA and FComp_Weighted models. As observed in Table II, the FComp_Weighted model consistently achieves or ties for the lowest RMSE across all stocks. While Table II only reports the best RMSE under FComp_SA and FComp_Weighted models, it is also important to investigate other cases to obtain optimal weights for combined forecasts, and Table III provides several valuable insights in this regard. The RMSE_2 column represents the RMSE using FComp_Weighted models, which is always lower than the RMSE obtained using the FComp_SA model. Thus, even if the user does not opt for the best combined model (e.g., Drift.ARIMA for AAPL), they will still benefit from improved RMSE. For example, the RMSE for the combined forecast with LSTM1.RF for AAPL is 18.787 under FComp_SA, while it is significantly reduced to 6.966 under FComp_Weighted. This improvement is attributed to the optimal weighting in FComp_Weighted. In this case, between LSTM1 and RF, the best individual model is RF. Consequently, more weight $(1 - 0.07 = 0.93)$ is assigned to RF price forecasts, leading to a superior combined forecast.

TABLE III
RMSE OF ALL POSSIBLE COMBINATIONS UNDER FCOMP_SA AND FCOMP_WEIGHTED MODELS FOR AAPL AND PLD PRICE FORECASTS

	AAPL			PLD		
Model	RMSE_1	RMSE_2	w^*	RMSE_1	RMSE_2	w^*
Drift.ARIMA	5.259	**5.253**	0.85	9.862	8.625	0.01
Drift.NNAR	6.107	5.267	0.99	10.824	10.491	0.01
Drift.LSTM1	19.861	5.275	0.99	3.722	3.195	0.39
Drift.XGB	6.266	5.269	0.99	8.242	5.562	0.01
Drift.LSTM2	17.459	5.273	0.99	6.497	3.102	0.01
Drift.RF	6.285	5.271	0.99	9.663	8.286	0.01
ARIMA.NNAR	6.197	5.306	0.99	9.512	8.618	0.99
ARIMA.LSTM1	20.209	5.361	0.99	3.121	3.073	0.46
ARIMA.XGB	6.359	5.308	0.99	7.009	5.539	0.01
ARIMA.LSTM2	17.801	5.352	0.99	5.327	3.094	0.01
ARIMA.RF	6.204	5.301	0.99	8.405	8.262	0.01
NNAR.LSTM1	21.454	7.393	0.99	3.535	3.178	0.41
NNAR.XGB	7.423	7.254	0.99	7.898	5.555	0.01
NNAR.LSTM2	19.061	7.371	0.99	6.171	3.099	0.01
NNAR.RF	6.835	6.821	0.58	9.317	8.279	0.01
LSTM1.XGB	21.584	7.748	0.01	3.082	**2.856**	0.40
LSTM1.LSTM2	36.068	33.628	0.01	4.335	2.994	0.09
LSTM1.RF	18.787	6.966	0.07	3.225	3.205	0.53
XGB.LSTM2	19.195	7.726	0.99	3.951	3.082	0.01
XGB.RF	6.983	6.983	0.50	6.824	5.535	0.99

RMSE_1 - RMSE using FComp_SA model, RMSE_2 - RMSE using FComp_Weighted model, and w^* optimal weight associated with Model 1 forecasts under FComp_Weighted model

Figure 4 visualizes the optimal weights for AAPL that minimize the forecast error sum of squares (FESS). This optimal weight differs from the equal weighting (0.5) used in FComp_SA highlighting the robustness of the forecast combination approach. While individual models occasionally achieve the lowest RMSE for specific assets, the combination models, particularly FComp_Weighted, provide consistently competitive forecasts across all assets. This underscores the value of using multiple models to enhance forecast accuracy and mitigate the risk of relying on a single model's performance.

B. Risk Forecasts

In this subsection, we further discuss the findings from the empirical results of our experimental study, deriving the new risk measure along with the benefits of using optimal combined forecasts. Preliminary results show that the price difference series ($d_t = P_t - P_{t-1}$) for underlying assets appear

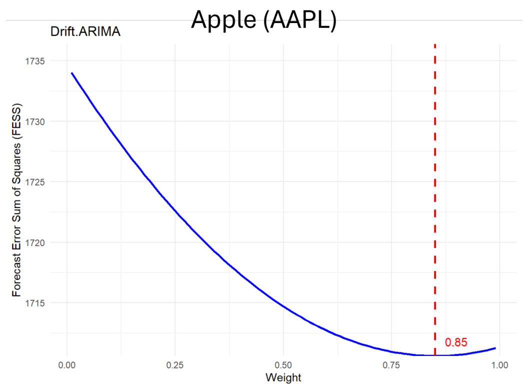

Fig. 4. Optimal weight (w^*) based on FESS of Price for best FComp_Weighted Model.

relatively stationary. For generalization of the results we have chosen 110 stocks (10 stocks from each sector considered) and both the stationarity tests; the Augmented Dickey-Fuller (ADF) and Kwiatkowski-Phillips-Schmidt-Shin (KPSS), are carried out. The reported P-values are less than the level of significance (0.05) indicating that we do not have statistical evidence supporting non-stationarity in the price difference series.

Figure 5 shows the point forecasts of the new risk measure ($DDRisk_{new}$) for AAPL and PLD during the testing period (the same test period as in the price forecast) using individual models. The black line in the figure represents the observed risk during the testing period. Forecasts using the LSTM1 and LSTM2 models are relatively similar and even for both assets. However, NNAR, XGB, and RF successfully capture the variations during the testing period. Moreover, for AAPL, NNAR and XGB appear to overestimate the risk during the latter part of the testing period. For PLD, except for XGB, the other models barely capture the high risks during the middle of the testing period.

Forecast accuracy (RMSE) for $DDRisk_{new}$ with individual, FComp_SA, and FComp_Weighted models are reported in Table IV. The same training period as for price data is also used for $DDRisk_{new}$, and the minimum RMSE for each stock is highlighted with bold numbers. Among all models, the FComp_Weighted model consistently achieves or ties for the lowest RMSE across all stocks, demonstrating its superior performance. For AAPL, the RMSE of the FComp_Weighted model is 1.564, which is only marginally lower than the RMSE of the FComp_SA model (1.568), highlighting the slight advantage of using weighted combinations. Similarly, for LLY, the FComp_Weighted model achieves the lowest RMSE of 5.985, outperforming the FComp_SA model and all individual models. For WMT, the FComp_Weighted model also achieves the minimum RMSE of 0.323, closely followed by the RMSE of the FComp_SA model (0.330). This highlights that while both combination models perform well, the weighted approach yields a marginally better forecast. For PLD, the FComp_Weighted model has an RMSE of

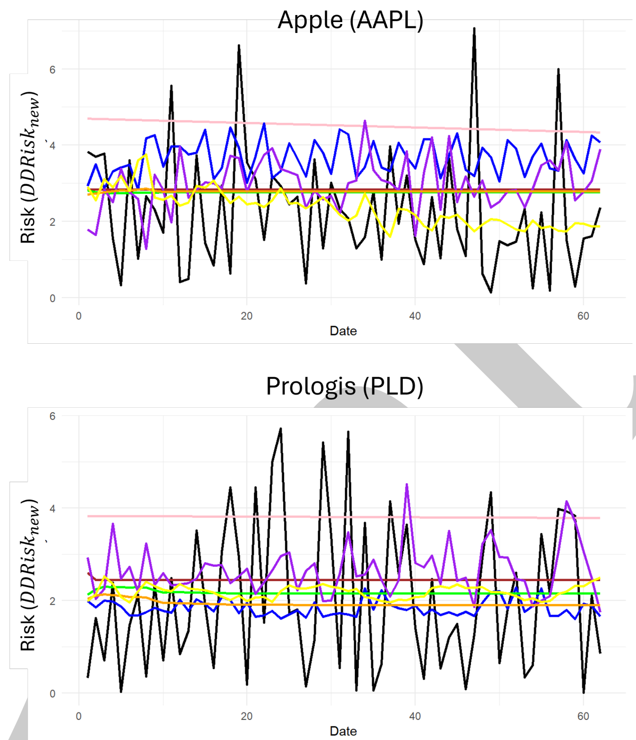

Fig. 5. $DDRisk_{new}$ forecasts of AAPL (top) and PLD (bottom) using Drift (Pink), ARIMA (Brown), NNAR (Blue), LSTM1 (Green), XGB (Purple), LSTM2 (Orange), and RF (Yellow) Models.

1.566, again slightly better than the FComp_SA model (1.573) and the best individual model, RF (1.616). For NEE, both FComp_SA and FComp_Weighted models tie for the lowest RMSE of 0.865, which is better than any individual model. Overall, these results underscore the importance of using forecast combination models, particularly the weighted approach, as they consistently provide the most accurate forecasts. The marginal improvements observed with FComp_Weighted over FComp_SA further justify the preference for the weighted approach, as it ensures more precise risk forecasts across all stocks.

TABLE IV
FORECAST ACCURACY (RMSE) OF $DDRisk_{new}$

Model	AAPL	LLY	WMT	PLD	NEE
Drift	2.747	7.922	0.431	2.367	0.893
ARIMA	1.650	6.017	0.383	1.637	1.005
NNAR	2.189	6.184	0.348	1.606	0.911
LSTM1	1.621	6.152	0.363	1.596	0.925
XGB	1.850	6.533	0.385	1.693	0.899
LSTM2	1.633	6.166	0.369	1.602	0.912
RF	1.576	6.672	0.324	1.616	0.876
FComp_SA	1.568	6.023	0.330	1.573	**0.865**
FComp_Weighted	**1.564**	**5.985**	**0.323**	**1.566**	**0.865**

Similar to the price forecast, the RMSE for AAPL and PLD risk forecasts using all possible combinations under the FComp_SA and FComp_Weighted models is investigated and reported in Table V. For any combination of the individual models, the RMSE with FComp_Weighted is always smaller compared to the FComp_SA model, emphasizing the superiority of the new approach.

TABLE V
RMSE OF ALL POSSIBLE COMBINATIONS UNDER FCOMP_SA AND FCOMP_WEIGHTED MODELS FOR AAPL AND PLD $DDRisk_{new}$ FORECASTS

	AAPL			PLD		
Model	$RMSE_1$	$RMSE_2$	w^*	$RMSE_1$	$RMSE_2$	w^*
Drift.ARIMA	2.104	1.656	0.01	1.919	1.640	0.01
Drift.NNAR	2.436	2.193	0.01	1.765	1.590	0.11
Drift.LSTM1	2.077	1.626	0.01	1.846	1.597	0.01
Drift.XGB	2.186	1.853	0.01	1.958	1.696	0.01
Drift.LSTM2	2.087	1.639	0.01	1.791	1.597	0.07
Drift.RF	1.950	1.577	0.01	1.856	1.617	0.01
ARIMA.NNAR	1.879	1.653	0.99	1.590	1.588	0.38
ARIMA.LSTM1	1.635	1.621	0.01	1.610	1.596	0.01
ARIMA.XGB	1.720	1.650	0.99	1.640	1.632	0.78
ARIMA.LSTM2	1.641	1.633	0.01	1.599	1.595	0.29
ARIMA.RF	1.577	1.568	0.24	1.619	1.616	0.13
NNAR.LSTM1	1.857	1.624	0.01	1.590	1.589	0.39
NNAR.XGB	1.965	1.851	0.01	1.573	**1.566**	0.64
NNAR.LSTM2	1.865	1.636	0.01	1.601	1.600	0.37
NNAR.RF	1.754	1.577	0.01	1.597	1.596	0.59
LSTM1.XGB	1.704	1.622	0.99	1.604	1.589	0.80
LSTM1.LSTM2	1.627	1.621	0.99	1.595	1.594	0.67
LSTM1.RF	1.568	**1.564**	0.32	1.605	1.596	0.99
XGB.LSTM2	1.710	1.634	0.01	1.585	1.576	0.32
XGB.RF	1.629	1.572	0.11	1.614	1.605	0.26

$RMSE_1$ - RMSE using FComp_SA model, $RMSE_2$ - RMSE using FComp_Weighted model, and w^* optimal weight associated with Model 1 forecasts under FComp_Weighted model

Figure 6 presents the optimal weight (w^*) derived based on FESS for the best FComp_Weighted model, similar to Figure 4 in the price forecast section. The observed consistency in forecast accuracy across both price and risk measures reinforces the effectiveness of the forecast combination approach, particularly the weighted methodology, in minimizing errors and providing reliable forecasts.

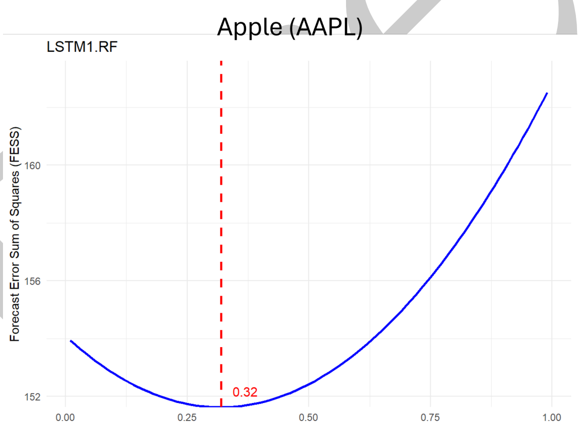

Fig. 6. Optimal weight (w^*) based on FESS of $DDRisk_{new}$ for best FComp_Weighted Model.

IV. CONCLUSIONS

This study demonstrates the transformative potential of combining machine learning techniques with traditional forecasting models to enhance financial asset price prediction and risk assessment. By introducing a novel risk measure ($DDRisk_{new}$) and using models such as LSTM, NNAR, Random Forest, and XGBoost, along with optimal forecast combination strategies, the proposed framework achieves superior predictive accuracy across diverse financial assets. The findings highlight the efficacy of weighted forecast combinations (FComp_Weighted) in minimizing forecast errors (MAE, RMSE, and MAPE). The research underscores the versatility and robustness of machine learning in handling the complexities of financial data while offering computationally efficient and scalable solutions. These results have practical implications for algorithmic trading, portfolio optimization, and strategic investment planning. Future work can explore the integration of probabilistic forecast combinations and extend the framework to emerging domains such as renewable energy and cryptocurrency markets, further broadening its impact.

REFERENCES

[1] Pala, Z., Ünlük, İ. H., & Yaldız, E. (2019). Forecasting of electromagnetic radiation time series: An empirical comparative approach. The Applied Computational Electromagnetics Society Journal (ACES), 1238-1241.
[2] Darwish, A., Rahhal, Y., & Jafar, A. (2020). A comparative study on predicting influenza outbreaks using different feature spaces: application of influenza-like illness data from Early Warning Alert and Response System in Syria. BMC research notes, 13(1), 33.
[3] WAHYUDI, S. T. (2017). The ARIMA Model for the Indonesia Stock Price. International Journal of Economics & Management, 11.
[4] Al-Musaylh, M. S., Deo, R. C., Adamowski, J. F., & Li, Y. (2018). Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia. Advanced Engineering Informatics, 35, 1-16.
[5] Hyndman, R. J. (2018). Forecasting: principles and practice. OTexts.
[6] Pala, Z., & Atici, R. (2019). Forecasting sunspot time series using deep learning methods. Solar Physics, 294(5), 50.
[7] Chukwueloka, E. H., & Nwosu, A. O. (2023). Modelling and Prediction of Rainfall in the North-Central Region of Nigeria Using ARIMA and NNETAR Model. In Climate Change Impacts on Nigeria: Environment and Sustainable Development (pp. 91-114). Cham: Springer International Publishing.
[8] Martínez, F., Charte, F., Rivera, A. J., & Frías, M. P. (2019, May). Automatic time series forecasting with GRNN: A comparison with other models. In International Work-Conference on Artificial Neural Networks (pp. 198-209). Cham: Springer International Publishing.
[9] Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019, December). The performance of LSTM and BiLSTM in forecasting time series. In 2019 IEEE International conference on big data (Big Data) (pp. 3285-3292). IEEE.
[10] Moldovan, D., Anghel, I., Cioara, T., & Salomie, I. (2019, October). Time series features extraction versus lstm for manufacturing processes performance prediction. In 2019 international conference on speech technology and human-computer dialogue (SpeD) (pp. 1-10). IEEE.
[11] Cheng, X., Liu, N., Guo, L., Xu, Z., & Zhang, T. (2020, July). Blocking bug prediction based on XGBoost with enhanced features. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 902-911). IEEE.
[12] Joiner, D., Vezeau, A., Wong, A., Hains, G., & Khmelevsky, Y. (2022, November). Algorithmic trading and short-term forecast for financial time series with machine learning models; state of the art and perspectives. In 2022 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE) (pp. 1-9). IEEE.
[13] Bates, J. M., & Granger, C. W. (1969). The combination of forecasts. Journal of the operational research society, 20(4), 451-468.
[14] Lichtendahl Jr, K. C., & Winkler, R. L. (2020). Why do some combinations perform better than others?. International Journal of Forecasting, 36(1), 142-149.
[15] Jose, V. R. R., & Winkler, R. L. (2008). Simple robust averages of forecasts: Some empirical results. International journal of forecasting, 24(1), 163-169.

[16] Schmidhuber, J., & Hochreiter, S. (1997). Long short-term memory. Neural Comput, 9(8), 1735-1780.

[17] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[18] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[19] Hoque, M. E., Thavaneswaran, A., Appadoo, S. S., Thulasiram, R. K., & Banitalebi, B. (2021, July). A novel dynamic demand forecasting model for resilient supply chains using machine learning. In 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 218-227). IEEE.

[20] Thavaneswaran, A., Paseka, A., & Frank, J. (2020). Generalized value at risk forecasting. Communications in Statistics-Theory and Methods, 49(20), 4988-4995.

[21] Liang, Y., Thavaneswaran, A., Paseka, A., Qiao, W., Ghahramani, M., & Bowala, S. (2022, June). A novel optimal profit resilient filter pairs trading strategy for cryptocurrencies. In 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1274-1279). IEEE.

[22] Liang, Y., Thavaneswaran, A., Yu, N., Hoque, M. E., & Thulasiram, R. K. (2020, July). Dynamic data science applications in optimal profit algorithmic trading. In 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC) (pp. 1314-1319). IEEE.

[23] Thavaneswaran, A., Liang, Y., Zhu, Z., & Thulasiram, R. K. (2020, July). Novel data-driven fuzzy algorithmic volatility forecasting models with applications to algorithmic trading. In 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (pp. 1-8). IEEE.

APPENDIX

In Tables VI and VII, we further explore the performance of individual and combination models through the metrics of mean absolute error (MAE) and mean absolute percentage error (MAPE). These tables reveal similar patterns consistent with the RMSE findings, where combination models, particularly FComp_Weighted, demonstrate superior performance across most assets. For example, in terms of MAE, FComp_Weighted achieves the lowest error for AAPL, PLD, and NEE, while individual models such as Drift and ARIMA perform well for LLY and WMT. Similarly, MAPE values reinforce the effectiveness of FComp_Weighted in providing accurate forecasts by minimizing relative errors. The inclusion of MAE and MAPE alongside RMSE provides a comprehensive assessment of model performance, highlighting the advantages of forecast combination methods in reducing forecast errors across diverse metrics and assets.

TABLE VI
FORECAST ACCURACY (MAE) OF PRICE

Model	AAPL	LLY	WMT	PLD	NEE
Drift	4.470	20.186	1.374	10.414	1.570
ARIMA	4.517	20.154	1.603	7.908	1.953
NNAR	6.143	22.218	1.903	9.697	1.458
LSTM1	37.909	105.732	3.099	7.053	1.811
XGB	6.452	22.579	1.688	4.710	3.913
LSTM2	32.835	110.327	2.760	2.339	1.754
RF	6.538	22.608	1.680	7.456	1.385
FComp_SA	4.481	20.170	1.482	2.404	**1.331**
FComp_Weighted	**4.468**	**20.109**	**1.376**	**2.289**	**1.331**

TABLE VII
FORECAST ACCURACY (MAPE) OF PRICE

Model	AAPL	LLY	WMT	PLD	NEE
Drift	2.871	5.746	**2.875**	9.072	2.175
ARIMA	2.883	5.728	3.329	6.902	2.687
NNAR	3.866	6.175	3.937	8.450	2.026
LSTM1	24.073	30.288	6.461	6.011	2.501
XGB	4.059	6.292	3.499	4.127	5.375
LSTM2	20.825	31.644	5.739	2.042	2.473
RF	4.264	6.315	3.483	6.511	1.947
FComp_SA	2.869	5.737	3.089	2.094	**1.862**
FComp_Weighted	**2.867**	**5.695**	2.879	**1.991**	**1.862**

Similar to price forecast, the forecast accuracy of $DDRisk_{new}$ with MAE and MAPE are reported in Table VIII and IX.

TABLE VIII
FORECAST ACCURACY (MAE) OF $DDRisk_{new}$

Model	AAPL	LLY	WMT	PLD	NEE
Drift	2.524	6.948	0.377	2.051	**0.569**
ARIMA	1.358	4.179	0.333	1.413	0.840
NNAR	1.905	3.784	0.292	**1.297**	0.705
LSTM1	1.327	3.719	0.312	1.357	0.736
XGB	1.561	4.098	0.320	1.427	0.656
LSTM2	1.341	**3.673**	0.319	1.333	0.710
RF	**1.192**	4.017	0.267	1.382	0.606
FComp_SA	1.232	3.696	0.272	1.314	0.578
FComp_Weighted	1.194	3.674	**0.266**	1.308	0.584

TABLE IX
FORECAST ACCURACY (MAPE) OF $DDRisk_{new}$

Model	AAPL	LLY	WMT	PLD	NEE
Drift	339.845	442.980	222.716	1320.945	485.522
ARIMA	191.287	222.101	189.746	835.773	714.334
NNAR	263.115	157.553	162.193	**628.612**	620.970
LSTM1	184.768	154.755	175.695	740.812	652.544
XGB	205.909	160.534	157.706	953.610	**407.462**
LSTM2	187.094	148.053	180.712	659.753	627.157
RF	**145.112**	**135.673**	**136.492**	754.676	513.458
FComp_SA	163.777	143.808	143.388	644.147	443.858
FComp_Weighted	146.093	148.145	136.839	648.205	445.406