Ensemble Learning Models for Large-Scale Time Series Forecasting in Supply Chain

Minjuan Zhang and Chase Q. Wu
Department of Data Science
New Jersey Institute of Technology
Newark, NJ 07102, USA
Email: {mz339,chase.wu}@njit.edu

Aiqin Hou
School of Information Science and Technology
Northwest University
Xi'an, Shaanxi 710127, China
Email: houaiqin@nwu.edu.cn

***Abstract*—Machine learning techniques have gained significant traction in supply chain forecasting, driven by the increasing availability of data assets. These techniques offer opportunities to optimize management processes, reduce operational costs, and enhance decision-making for enterprise success. However, conventional statistical approaches dominating time series forecasting, such as the Autoregressive-moving-average model (ARMA), dynamic regression, and unobserved component models (UCMs), suffer from limitations in model accuracy and performance. They struggle to handle batch processing, large-scale big data, uncertainty-induced disruptions, and the synchronization of demand and supply scenarios. To address these challenges, we propose a class of ensemble techniques that combine neural networks with baseline models. Firstly, we conduct classification and segmentation by leveraging feature engineering on signal components, such as spikes and anomalies as outlier skews, to capture the complexity of combined scenarios in categorical data hierarchies and identify patterns for ensemble forecasting. Subsequently, we employ an ensemble model equipped with time series pattern sensors to automatically discern signal components, encompassing seasonality, promotions, trends, and intermittent or discontinued activities. We evaluate the performance of eight commonly-used model categories, and our proposed ensemble modeling approaches exhibit substantial improvements in accuracy compared to individual baseline models and other univariate time series algorithms.**

***Index Terms*—Ensemble models, time series forecasting, large-scale data, supply chain, neural networks, stacking techniques**

I. Introduction

The advent of the digitalization revolution has ushered in a new era for enterprises, propelling them towards Industry 4.0 [1]. This transformation has permeated all facets of the supply chain [2], encompassing procurement, manufacturing, engineering, and customer management. Against the backdrop of complex decision-making scenarios, encompassing considerations of globalization versus localization [3], rapidly evolving technologies, and increasingly demanding customers, demand forecasting [4] within the corporate supply chain plays a pivotal role in satisfying customer requirements and gaining a competitive edge.

Demand forecasting represents the primary source of variance and uncertainty in integrated business planning (IBP), a process integral to strategic management systems. Its overarching aim is to identify improvement opportunities and define actionable steps involving all stakeholders. Enhancing demand forecasting yields a multiplier effect as it permeates the IBP process, influencing nearly every component within the supply chain. Even minor enhancements in forecasting capabilities can exert significant impact on revenue, costs, profitability, customer satisfaction, and working capital, surpassing the influence of other supply-oriented or non-supply-oriented elements. IBP, in general, entails managing vast quantities of disconnected data, rendering it one of the most structurally complex processes in business operations. By leveraging classification and segmentation techniques, the efficiency of this process can be augmented, leading to cost reduction, expedited predictions, and informed decision-making. Consequently, the supply chain management team can allocate more time to value-adding activities.

A. Ensemble Models vs Traditional Time Series Forecasting

Supply chain forecasting plays a critical role in enabling organizations to optimize their management processes, reduce operational costs, and make informed decisions for achieving success in the dynamic business landscape. With the increasing availability of data assets, machine learning techniques have emerged as a powerful tool in supply chain forecasting. These techniques offer promising opportunities to overcome the limitations of conventional statistical approaches, such as the Autoregressive-moving-average model (ARMA), dynamic regression, and unobserved component models (UCMs). While these traditional methods have been widely used, they often fall short in terms of model accuracy and performance, particularly when confronted with challenges like batch processing, handling large-scale big data, managing uncertainty-induced disruptions, and effectively synchronizing demand and supply scenarios.

To address the aforementioned challenges, we propose a novel class of ensemble techniques that combine neural networks with baseline models. Our approach leverages the power of machine learning to enhance supply chain forecasting accuracy and performance. The first step involves conducting classification and segmentation by employing feature engineering on signal components. This enables us to capture the inherent complexity of combined scenarios present in categorical data hierarchies and identify relevant patterns for ensemble

forecasting. By effectively analyzing and distinguishing signal components, including spikes, anomalies, and outlier skews, we can gain deeper insights into the underlying patterns and dynamics of the supply chain.

Furthermore, we employ an ensemble model that is equipped with specialized time series pattern sensors. These sensors enable the automatic differentiation of various signal components, encompassing seasonality, promotions, trends, and intermittent or discontinued activities. By utilizing these sensors, we can effectively capture and utilize the valuable information embedded in the time series data, thereby enhancing the accuracy and reliability of the forecasting process.

To assess the effectiveness of our proposed ensemble modeling approaches, we evaluate their performance against eight commonly-used model categories. The results demonstrate substantial improvements in accuracy when compared to individual baseline models and other univariate time series algorithms. These findings underscore the potential of our ensemble techniques in revolutionizing supply chain forecasting and enabling organizations to make more accurate and informed decisions, ultimately leading to enhanced operational efficiency and competitiveness.

Traditional time series forecasting models utilize a disturbance filter and potentially incorporate one or more inputs to characterize the behavior of a time series based on its lagged values. The first-order autoregressive (AR) model, a fundamental time series model, offers a simple explanation of time series behavior using initial values. The model can be represented as:

$$y_t = \phi y_{t-1} + a_t + c, \tag{1}$$

where y_t denotes the time series data at index t, ϕ represents the first-order autoregressive parameter, a_t signifies the randomized factor with zero mean and standardized variance σ^2, and c represents a constant term. It is important to note that a_s and a_t are uncorrelated for any $s \neq t$, indicating a white noise process. When the magnitude of ϕ is less than one, the series exhibits stationarity. In the case of a stationary time series, previous values exert an exponentially diminishing influence on the current value. The lags associated with a time series model can be of considerable complexity.

In time series analysis, it is common for one or more deterministic and/or stochastic input variables, referred to as regressor, exogenous, or explanatory variables, to have an influence on the observed time series. This influence can either be static or dynamic, meaning it may remain constant over time or change over time. To gain a deeper understanding of these inputs and their impact, a model with a time-varying mean can be employed. The model takes the following form:

$$y_t = \mu_t + \psi(B)a_t, \tag{2}$$

where μ represents the mean of the series, B corresponds to the backshift operator (such that $By_t = y_{t-1}$), $\psi(B)$ denotes the disturbance filter of either limited or infinite order, and μ_t is a time-varying constant that describes the influence of the inputs on the time series at each point in time. If the term μ_t is not affected by lagged input values, the model is commonly referred to as a regression with time series errors. Conversely, if the term μ_t varies based on lagged input values, the model is often referred to as a dynamic regression model.

This research focuses on a specific aspect of time series modeling and does not aim to address the broader issue of a time series model with stochastic inputs, although interventions can be considered as deterministic inputs. In the context of this study, we categorize the inputs into two types based on their influence on the time series. Inputs that exert a static influence are referred to as regressor variables, while those with a dynamic influence are termed dynamic regression variables or transfer function inputs. Furthermore, it is important to note that the time series model may incorporate various transformations, such as logarithmic, square root, logistic, or Box-Cox transformations, to enhance its representation and analysis.

B. Big Data and ETL Technologies

In recent years, the use of big data technologies centered around parallel computing has gained significant attention from enterprises across various industries for processing large-scale forecasting data. Big data, as defined by Gartner, refers to information assets characterized by high volume, velocity, and variety, requiring innovative and cost-effective methods of information processing to gain valuable insights and facilitate decision-making [5]. This demand for analyzing massive amounts of data has paved the way for a digital revolution in demand forecasting.

To address the challenges associated with big data forecasting, we propose an extract, transform, and load (ETL) process coupled with a MapReduce/Hadoop solution that leverages time series signal components to differentiate between various factors such as level, trend, seasonality, cycle, exogenous, and irregular components. Our approach begins by modeling classifications and segmentations of large-scale data as an automated ETL dataset generator, relying on the component features anticipated by traditional statistical models prior to the training and validation processes. Additionally, we introduce an ensemble model that enables more efficient time series forecasting while maintaining a high level of accuracy compared to baseline models and other time series approaches. Moreover, the dynamic ETL process acts as a labeling function for the supervised machine learning model and as a weight categorization function for the proposed ensemble learning approach based on stacking.

The main technical contributions of this paper are summarized as follows:

- We are among the pioneers in exploring the application of ensemble learning techniques and big data processing systems for large-scale demand forecasting in the supply chain domain.
- We utilize a set of time series components to represent the features of supervised machine learning labels extracted from the data source. The ETL process and MapReduce solution in this study incorporate various category inputs

that partially align with existing traditional time series models. This includes discovering distinctive short-term, long-term, seasonal, low-volume, retired, and intermittent patterns, as well as identifying groups of better-performing models for each distinctive parameter.
- To further validate our approach, we create a mixed ensemble model utilizing these time series parameters. The experimental results demonstrate that our proposed ensemble learning outperforms single-agent learning, with the integrated ensemble model achieving the highest classification accuracy among all compared models.

The remainder of this study is organized as follows. Section II provides an overview of related work. Sections III introduces the prediction framework and the method for intelligent integration, respectively. Section IV describes the MapReduce-based data processing method and the proposed ensemble learning techniques. The experimental setup, results, and analysis are presented in Section V. Finally, Section VI concludes our work and outlines potential future directions.

II. Related work

A. ARIMA Models Trained with Voting Techniques

Autoregressive integrated moving average (ARIMA) models have been widely used for time series forecasting over the past three decades [6]. These models incorporate time series integration to achieve stationarity and offer a diverse range of prediction intervals. However, evaluating and selecting appropriate ARIMA models can be challenging due to their complexity, especially in the context of time series forecasting.

A voting technique based on traditional time series models has been employed to analyze the monthly wage index of Russian macroeconomic statistics [7]. In this approach, 2/3 of the training set is utilized to construct ARIMA models and five ”good” models are selected for the subsequent stage. Equal weights are assigned to their votes, and the voting approach is applied using the remaining 1/3 of the training set. Each chosen model generates a one-month projection, and their predictions are compared against the actual data. The model with the most accurate prediction receives a higher weight, while the weights of other models are reduced, ensuring that the combined weight remains equal to one. It is important to note that the weights should not fall below zero during this process [8].

Initially, all models are considered equal in terms of their prediction quality. However, as the voting evaluation progresses, the weight of the model producing the most accurate forecasts is increased, while the weights of other models are decreased. This dynamic weight adjustment mechanism allows superior models to be identified and rewarded. The approach of combining models has demonstrated improved prediction quality in these studies, providing a valuable framework for evaluating models and their predictions.

Additionally, further investigation is needed to explore the prediction intervals of mixed models. Although the combination of models often leads to comparable or even superior forecasts, the narrowing of prediction intervals for model combinations is still a topic of ongoing research and will be addressed in future studies.

The concept of forecasting based on a collection of time series models can be likened to the bagging strategy used in classification and regression. However, it is crucial to establish and test the specific requirements that should be met by individual models aggregated into a set, analogous to the constraints imposed on weak classifiers.

By leveraging voting techniques and combining ARIMA models, this study aims to enhance the accuracy and reliability of time series forecasting. The subsequent sections will delve into the experimental setup, methodologies, and results, providing valuable insights into the field of time series analysis and forecasting.

B. Bagging for Time Series Forecasting

Bagging techniques have gained prominence in time series forecasting due to their ability to improve accuracy across a wide range of applications.

Fotios [9] proposed the Simple Combination of Univariate Models (SCUM) technique for generating point predictions and prediction intervals in the M4-competition entry. SCUM combines the point forecasts and prediction intervals from four models, namely, Exponential Smoothing, Complex Exponential Smoothing, Automatic Autoregressive Integrated Moving Average, and Dynamic Optimized Theta, using the median combination approach. This method performed well in the M4 competition, ranking 6th for point predictions and prediction intervals, and 2nd and 3rd for point forecasts of weekly and quarterly data, respectively.

Matheus [10] introduced an efficient bootstrap stacking technique applied to the Wind energy project to enhance its economic and environmental benefits. Forecasting time series data for wind energy generation is challenging due to the complex interplay of meteorological and demographic factors. Matheus employed an ensemble learning model that incorporates both bagging and stacking techniques to improve short-term wind energy generation evaluations. The ensemble model integrates samples using arithmetic and weighted average values, with weights determined through multi-objective optimization using a non-dominated sorting genetic algorithm of version II. Experimental results demonstrated that the proposed ensemble learning model outperformed single forecasting models, including stacking, machine learning, artificial neural networks, and statistical models, resulting in reduced error rates for out-of-sample forecasting. These findings highlight the effectiveness of integrating ensemble techniques for accurate forecasting in renewable energy.

Egrioglu [11] introduced a novel bootstrapped hybrid artificial neural network (ANN) for prediction. This approach utilizes the residual bootstrap technique to provide input significance testing and hypothesis testing for linearity and non-linearity. The technique employs bagging to generate predictions and outperforms other prominent neural networks and models in terms of prediction accuracy. Moreover, the suggested method exhibits improved stability and robustness

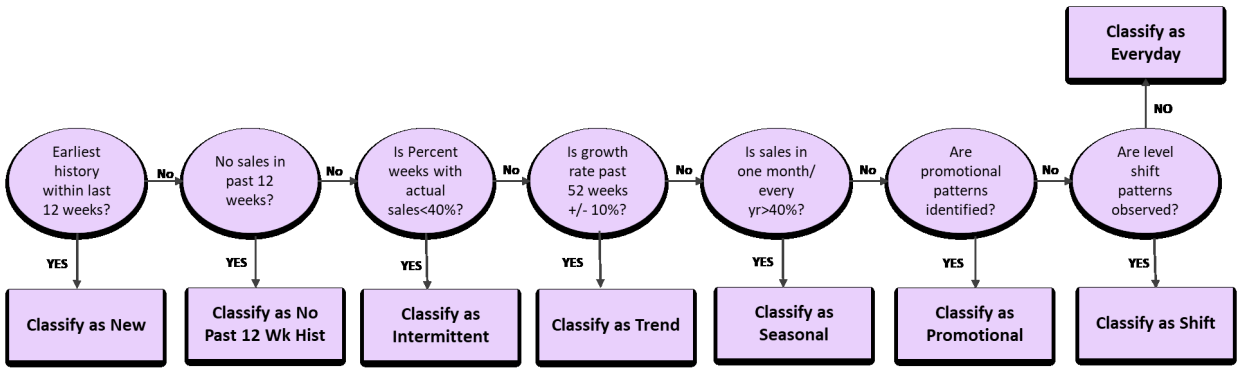


Fig. 1. Flow chart of classification.

by being less affected by initial random weights compared to previous neural networks.

By leveraging bagging techniques in time series forecasting, researchers have achieved significant improvements in prediction accuracy across various domains. The subsequent sections of this paper will explore the experimental setups, methodologies, and results, providing valuable insights for the application of bagging in time series forecasting.

III. Time Series and Deployment Framework in Large-Scale Dataset

Large-scale demand forecasting is a technique used to forecast the future consumption of electricity, natural gas, or other fuels in an energy market. Large-scale demand forecasting is one of the most important tools for planning and managing supply systems for power plants, transmission lines, distribution systems, and end users. Large-scale demand forecasts are also useful for evaluating the economic viability of new power plant projects. For example, the largest single-use case for large-scale demand forecasting is predicting power plant load factors (the percentage of time that a generator operates). Load factor forecasts are essential to plan how much fuel is needed to meet peak demands during periods when there may not be enough generation available from other sources, such as wind or solar farms. Power plants can also benefit from this information by adjusting their operating schedules based on expected load conditions. Let us consider a scenario where a plant will operate longer than usual during winter months because it has been forecasted that loads will increase due to cold weather. In that case, we can adjust the schedule to operate more hours per day during those colder months and fewer hours per day in warmer months without extra generation capacity. This allows us to maximize the amount of money generated from each hour of operation without having excessive generating capacity running around waiting in the standby mode, and get ready to ramp up production when needed by customers who have requested it through their utility company's call center or online portal system (e.g., PJM Interconnection).

The ETL process extracts data from the master data in the enterprise relational database, which defines for each combination of product and customer where sales orders should ship from. This information is only available in the database for distribution sales networks. Warehouse and export distribution networks are derived based on historical shipment data in the database and then maintained as location realignment instructions when changes from the default are part of the location realignment process governed by the management team. The supply chain sends inventory to the locations where customer services send sales orders. One of the goals of the demand planning process is to send forecasts to the exact location where orders will be sent so that when the orders are received, there is inventory to fill them. This is accomplished by taking the actual order on the shipment history, aligning it to the default ship-from location in the master data, and applying location realignments to history and forecast based on the business scenario.

A. Feature Detector and Weak Learner Classifier

In this work, we first focus on feature engineering and weak model classification. Therefore, we design an ETL MapReduce algorithm to extract large-scale data and classify all data into different training datasets for weak models based on the criteria in time. Each time series has only one classification. Given the classification and the series level in the data structure, the ensemble algorithm selects the model that can handle the type of series in the most efficient way. If the proposed algorithm needs to choose between several models for a time series, it lands on the simplest one that requires the least running time. Figure 1 illustrates the Decision Tree process that this feature detector uses to classify each data time series. The process is carried out automatically, and the classification for the series is stored in a column in the control table at the specific level of the time series with labels shown in Figure 2.

The classification for each series is automatically performed, and the resulting classification is stored in a specific level column of the control table. The classification components of the time series include the following:

- $Trend$ represents the long-term pattern of a series' means, which consists of a level and one or more movement characteristics.
- $Seasonality$ refers to the predicted deviation of series

patterns from the trend component. These deviations follow a periodic pattern, such as 52 weeks for the weekly series or 12 months for the monthly series. The nature of these departures from the series can be smooth (sinusoidal) or non-smooth.

- In contrast to the seasonal component, the $cycle$ component represents a regular, recurring cycle that does not have a fixed duration. For example, one cycle may last 20 months, while the next cycle may last 25 months.
- $Exogenous$ effects are external factors that cause fluctuations in the series but are not considered part of the series itself. These effects can be related to or caused by the values of another series. They are sometimes referred to as explanatory effects, and their relationship with the series can be concurrent or delayed. Additionally, an "event" refers to a form of external effect, where the occurrence or non-occurrence of the event can lead to temporary, semi-permanent, or permanent alterations in the series' evolution.
- Finally, the $irregular$ component represents the residual variation that remains after accounting for all other impacts and components. It is characterized by unpredictable fluctuations. Although white noise can serve as an irregular component, it is not necessary. The residuals may contain local, forecastable patterns, such as autocorrelation and moving averages.

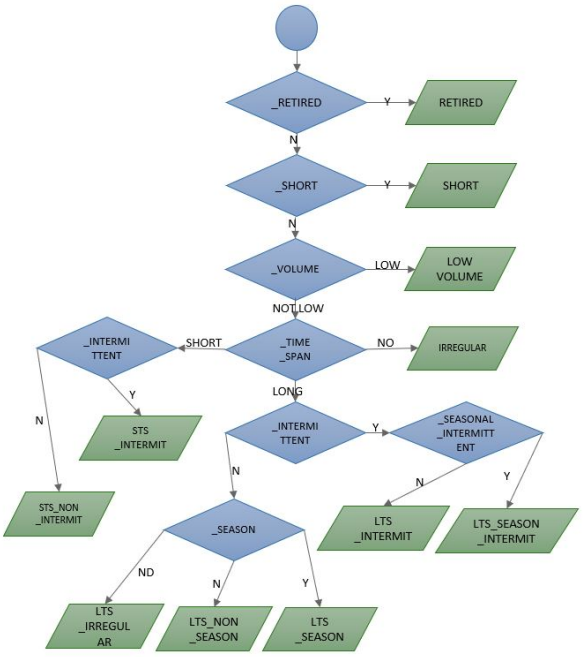


Fig. 2. Classification design for time series units.

B. ETL Training Dataset Generator with MapReduce Design

This section presents the proposed prediction model, which leverages the stacking integration method to enhance prediction performance. The architecture of the method is depicted in Figure 2, illustrating three key modules: data preprocessing and feature selection, stacking ensemble training, and model selection. These modules handle crucial tasks such as preprocessing the data, training the basic ensemble model, and selecting the final predictive model based on its score. The prediction algorithm is employed to estimate the running time of the CSM application. The monitor collects parameters, which serve as input for the ensemble algorithm, and the algorithm's output includes the running time of the simulation application and the CPU cores allocated to it.

C. Ensemble Model Predictor

The resource scheduler and simulation application scheduler are responsible for reallocating resources and redistributing entities within the simulation application to achieve load balance. Specifically, when the predictor forecasts the shortest running time for the simulation application, the resource scheduler reallocates optimal resources for the application's execution. Subsequently, the simulation application scheduler employs a comprehensive approach that minimizes synchronization overhead among simulation entities while ensuring load balancing. It redistributes the application across specified nodes for parallel acceleration.

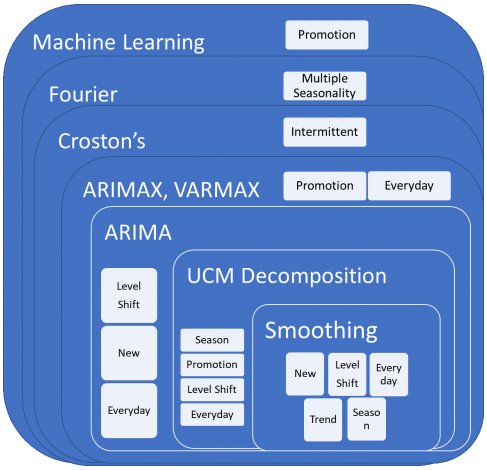


Fig. 3. Forecasting model family based on classification type.

IV. Prediction Algorithm Based on Stacking Ensemble Learning

The combination of multiple models' predictions has been widely acknowledged as a powerful approach that outperforms single models and effectively reduces prediction result variance [12]. Ensemble learning aims to create a stronger predictor by integrating individual prediction results obtained from various learning methods. In this paper, we propose a prediction method based on stacking ensemble learning (PASEL), as illustrated in Figure 6.

A. Dataset Generation

The business data is collected daily and preprocessed before being stored in the time series database as feature data. For

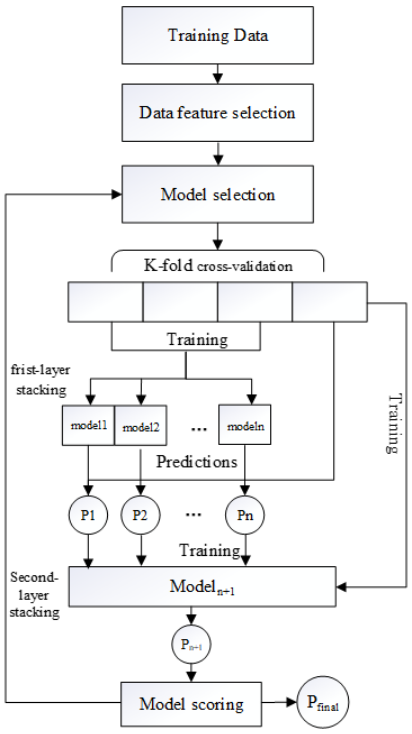


Fig. 4. A schematic diagram of the PASEL prediction algorithm.

Algorithm 1 A Prediction Algorithm Based on Stacking Ensemble Learning(PASEL)

Require: $Data(TS)$, $ModelList\ M = \{M_1, M_2, ..., M_n\}$
Ensure: BestMSet, RuntimePred, BestMScore
1: **for** each $key \in activekeylist$ **do**
2: **for** each $modelset \in modellist$ **do**
3: $start\ \{1 - layer - stacking\}$
4: Randomly split Data(TS) into k chunks $\{TS^j\}_{j=1}^{k}$
5: **for** j=1 to k **do**
6: $start\ \{k - fold\ bagging\}$
7: **for** each model m in M **do**
8: Training m-model on $\{TS^{-j}\}$
9: Make predictions P^j on TS^j
10: **end for**
11: **end for**
12: Choose model M_i in M $start\ 2 - layer - stacking$
13: Train m model on $\{TS^j,\ Y^j\}$ and predict P_{final}
14: $end\ stacking$
15: Compute StackingMScore= R2
16: **if** $StackingMScore > BestScore$ **then**
17: $bestscore \leftarrow StackingMScore$
18: $BestMSet \leftarrow ModelSet$
19: $P_{final} \leftarrow P_{final}$
20: **end if**
21: **end for**
22: **end for**=0

our prediction approach, we consider time series parameters that contribute to the prediction performance. The specific parameters used are outlined in Table I.

B. Selection of Prediction Model

Meeting the resource requirements for simulation accuracy with a single machine learning model can be challenging. To improve performance, the ensemble model combines multiple basic models [13]. However, not all basic models contribute positively to the integration model's performance. Therefore, a separate pre-run is conducted to pre-select machine learning prediction models for each classification shown in Figure 3. The model with the highest score (AIC or MAPE) is then determined through the model selection process. The specific model and its parameters are detailed in Table II.

C. Proposed Algorithm

The ensemble model offers a solution to the limitations of individual basic predictions by leveraging the interactions between these models to enhance prediction capabilities [14]. In this study, we introduce the PASEL algorithm, as shown in Figure 4, which selects an optimal subset of base models based on the characteristics of the data. By doing so, we aim to further enhance the prediction accuracy of the ensemble model. The specific steps of the algorithm are outlined in Algorithm 1.

In this algorithm, the list of models utilized for ensemble learning is denoted as ModelSet, while ModelList represents the list of all basic models used by the method. It is important to note that ModelSet is a subset of ModelList ($ModelSet \subseteq ModelList$). Within the context of the current ModelSet, "BestMSet" refers to the most effective combination of models, and "BestMScore" represents the highest achievable model score.

Firstly, a pre-selected method is employed to select a model combination from the ModelList, initiating the stacking integration training process.

Lines 1 to 11 depict the training of all models in the first layer using k-fold cross-validation, resulting in the generation of predicted values (Pj) for each model (M), as indicated in the output.

Lines 12 to 15 demonstrate the second layer of the stacking integration process. During this stage, the predicted values from the first layer models are used as features, and the final ensemble model is trained by combining these features with the initial ones.

Lines 14 to 18 outline the model selection procedure. The R2 value is employed to evaluate the ensemble model. If the score of the currently active model combination surpasses the score of the best model, the best model combination and best model score are updated accordingly. Ultimately, the algorithm outputs the best model combination and the corresponding forecast.

D. Evaluation Metrics

In this paper, we employ various evaluation metrics to assess the performance of the demand prediction models. These metrics include Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root-Mean-Squared Error (RMSE), Accuracy (ACC), Coefficient of Determination (R2),

TABLE I
CLASSIFICATION AND FORECASTING METHODS FOR EACH TYPE

Classification	Implication for Forecasting
New	Insufficient data available for reliable statistical forecasting.
No Past 12 Wk Hist	At this point, the statistical forecast stops updating and becomes zero or assumes an end-of-life profile. These special cases have a zero baseline, and the forecast relies on enrichment. Intermittent sales products are treated differently, utilizing intermittent classification and a continuation field.
Intermittent	Forecasting intermittent series pose challenges. The process employs Croton's Model, which distinguishes two components of demand: interval and magnitude. It models these components separately, providing a stable forecast even in periods with no sales.
Trending	Exponential Smoothing models are used for series exhibiting trends. These models differentiate trends from the base and assign more weight to recent data. They are particularly effective for short-term forecasting.
Seasonal	Seasonal trends are more accurately identified using monthly data rather than weekly data. For seasonal series, the process tests Holt-Winters, UCM Decomposition, and Fourier models to select the best-performing model. Fourier models are used when the data exhibits multiple seasonality patterns and requires the identification of peaks and troughs.
Promotional	Promotional series utilize a baseline of non-promoted volume, with promotions enriching the forecast. UCM, ARIMAX, VARMAX, and Machine Learning models are tested to identify the best-performing model.
Level shift	Holt-Winters, UCM Decomposition, and ARIMA models are tested to detect structural changes and select the best-performing model. These models are effective in identifying shifts but require substantial amounts of data.
Everyday	This scenario presents the best case for forecast accuracy, allowing the process to test multiple models and select the best-performing one. The process considers Smoothing models, UCM Decomposition, ARIMA, and VARMAX models for selection.

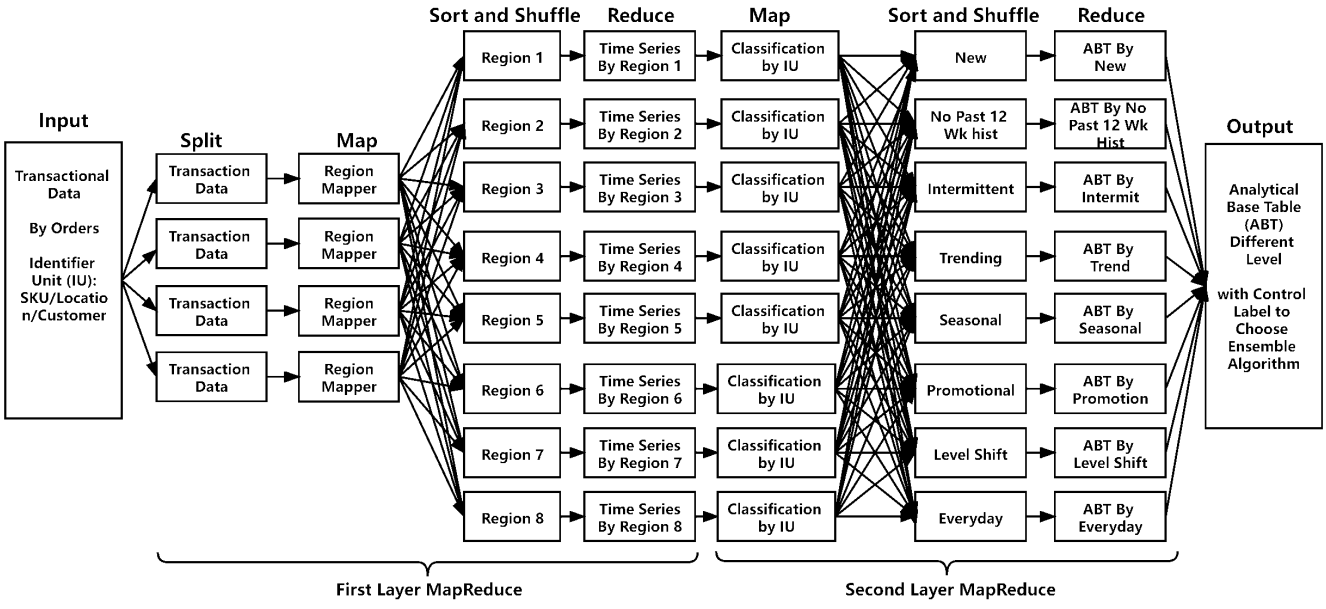
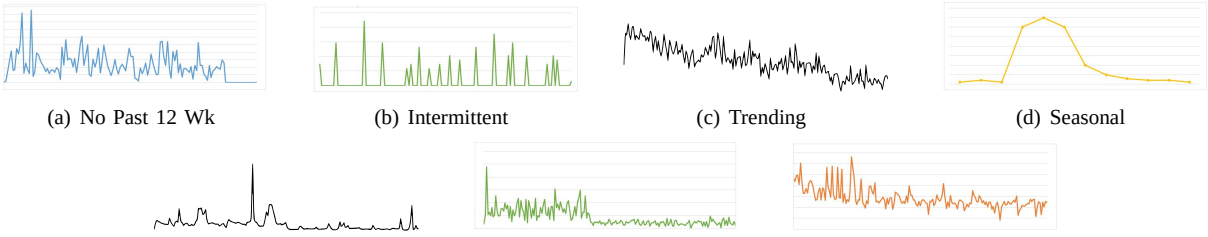


Fig. 5. A schematic diagram of the two-layer MapReduce algorithm (PA2LMR).



(a) No Past 12 Wk (b) Intermittent (c) Trending (d) Seasonal

(e) Promotional (f) Level Shift (g) Everyday

Fig. 6. Pattern detectors in individual ensemble weak models.

TABLE II
MACHINE LEARNING OR TIME SERIES MODELS.

Model	Method Used	Classification
ARIMA	Auto-regressive Integrated Moving Average	Seasonal, Non Seasonal, Other, Short, Low Volume, Retired
ESM(BESTS)	Exponential Smoothing Model-Seasonal	Seasonal
UCM	Unobserved Components Model	Seasonal, Non Seasonal, Other, Short, Low Volume, Retired
SIMPLEREG	Simple Regression Model	Seasonal, Non Seasonal, Other, Short, Low Volume, Retired
AVERAGE	Simple Average	Seasonal, Non-Seasonal, Other, Short, Low Volume, Retired
IDM	Intermittent Demand Model	Intermittent
ESM(BESTN)	Exponential Smoothing Model-NonSeasonal	Other, Short, Low Volume, Retired

and Akaike Information Criterion (AIC), defined as follows:

$$MAE = \frac{\sum_{i=1}^{n} |y_r^i - y_p^i|}{n}, \tag{3}$$

$$MAPE = \frac{\sum_{i=1}^{n} |\frac{y_r^i - y_p^i}{y_r^i}|}{n} \times 100, \tag{4}$$

$$ACC = 1 - \frac{100\%}{n} \sum_{i=1}^{n} \frac{|y_r^i - y_p^i|}{y_p^i}, \tag{5}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_r^i - y_p^i)^2}{n}}, \tag{6}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_r^i - y_p^i)^2}{\sum_{i=1}^{n} (\overline{y^i} - y_p^r)^2}, \tag{7}$$

$$AIC = 2K - 2ln(\hat{L}), \tag{8}$$

where y_r^i and y_p^i represent the real and predicted values of the i-th sample, respectively, $\overline{y_r}$ is the average of the true values, K is the number of independent variables used, and $\hat{L}$ is the log-likelihood estimate.

V. Implementation and Performance Evaluation

The ETL process relies on two components: (1) a substantial volume of transaction data generated from daily business activities, and (2) layers of business domains that aggregate to form the final analytical base table in a time series format. In our algorithm, we employ the MapReduce schema, which utilizes a cluster of virtual machine instances for parallel computing. These instances are allocated to perform transformation and classification tasks. The objective of the MapReduce algorithm is to enhance performance and minimize the data preprocessing time. Figure 5 illustrates the specific process.

In this study, we implement the MapReduce process as a two-layer framework consisting of Mappers and Reducers. This framework facilitates the cleaning and preprocessing, joining, and partitioning of transaction data. The sheer magnitude of the data necessitates the utilization of parallel computing to address real-world business challenges. However, due to the complexity of business cases, a single-layer MapReduce job is insufficient for solving large-scale forecasting problems. By employing multiple rounds of data processing and employing a combination of mapper and reducer operations, we can effectively tackle large-scale business problems within an acceptable running time.

A. Implementation of the Proposed Algorithm

The utilization of MapReduce not only facilitates the preparation of input data but also enables the resolution of real-world problems on a large scale. In this work, the PA2LMR algorithm has been designed by incorporating synchronization. This algorithm transforms the vast amount of transactional data into time series data, region by region, and employs the second-layer MapReduce to perform classification and generate forecast data. The specific steps of the algorithm are outlined in Algorithm 2.

In this algorithm, Data TD represents all transaction data with Identifier Unit (IU), which combines SKU, Location, and Customer IDs. Each Data entry contains a timestamp along with corresponding demand quantities. The first-layer MapReduce sorts the Data by region and transforms it into a time series format, aggregating quantities on a weekly

Algorithm 2 A Preprocessing Algorithm based on 2-Layer MapReduce(PA2LMR)

Require: $Data(TD)$ {Transactional Data by Identifier Unit -IU}
Ensure: $AnalyticalBaseTable(ABT)$
1: MAPPER1(RegionID r, IU) {1 Layer Mapper}
2: **for** all $recordt \in r$ **do**
3: EMIT(record r, count sum)
4: **end for**
5: REDUCER1(RegionID r, IU) {1 Layer Reducer}
6: **for** all $recordt \in IU$ **do**
7: Transform Transaction Data to Time Series Data by Weekly and Run Classification Algortims on each IU
8: EMIT(Identifier IU, Data TD)
9: **end for**
10: MAPPER2(Classification c, IU) {2 Layer Mapper}
11: **for** all $recordr \in c$ **do**
12: EMIT(record r, count sum)
13: **end for**
14: REDUCER2(Classification c, IU) {2 Layer Reducer}
15: **for** all $recordt \in IU$ **do**
16: Run PASEL algorithm on every IU based on Classification.
17: EMIT(Identifier IU, Data TD)
18: **end for**=0

basis. The second-layer MapReduce applies a classification algorithm within each region, categorizing the data into different model categories, and subsequently performs forecast algorithms based on these classifications.

Initially, transaction data is extracted from the system using ETL queries. Lines 1 through 4 illustrate the first layer mapper, which sorts and shuffles all transaction data to the respective regions. Lines 5 through 9 correspond to the first-layer reducer, which transforms the data into a time series format by aggregating quantities on a weekly basis. Lines 10 through 13 represent the second-layer mapper process, responsible for identifying the classification and shuffling the data accordingly. Finally, lines 14 through 18 depict the second-layer reducer process, which employs the aforementioned PASEL ensemble models to generate predictions based on the current classification.

B. Experimental Results and Analysis

1) Data Collection and Experiment Environment

The experimental data remain confidential due to the need to maintain the anonymity of the private business collection. The time series history period is collected as indicated in Table III. The experiment was conducted in an environment consisting of a cluster of 10 nodes running on cloud VMs with E2 instances and 128GB memory.

2) Running Time and Performance Evaluation

This MapReducer approach, employed as a data preprocessing strategy, serves to reduce model complexity, harness the power of parallel computing, and eliminate unnecessary attributes. The overall processing time, from input data to output generation, is reduced to 7-8 hours, as shown in Table IV.

TABLE III
PARAMETER CONFIGURATION.

Parameters	value
Level of Time Series History	[1,2,3]
Period of History	156
Holdout Periods	[0,12,16,20]
Prediction Period	131

TABLE IV
MODEL PERFORMANCE EVALUATION.

Model	R^2	RMSE	MAPE	ACC (%)
ARIMA	0.802	91.79	61.13	71.31
ESM BESTS)	0.742	57.28	31.83	80.59
UCM	0.748	53.03	32.23	81.35
SIMPLEREG	0.846	56.61	33.51	82.37
AVG	0.857	48.67	31.43	82.45
IDM	0.762	45.71	30.03	84.37
ESM (BESTS)	0.742	52.28	31.83	80.59
PASEL	0.877	39.56	25.69	88.54

To assess the efficacy of the algorithm, a series of extensive experiments were conducted, varying the number of virtual machine instances as worker nodes from 3 to 99 and the number of master nodes from 1 to 10. Initially, the performance of each cluster distribution was measured, followed by an evaluation of the ensemble model. The results of these experiments are presented in Table IV. These findings reveal that each model exhibits distinct performance characteristics across different evaluation metrics. Specifically, while a particular prediction model may outperform others in terms of error rate, its accuracy might be comparatively lower. For instance, referring to Table 4, the Simple Regression model achieves an accuracy of 82.37% (higher than that of the UCM model), but its RMSE and MAPE values are 56.61 and 33.51, respectively, surpassing the error rates of the UCM model. Moreover, the ensemble model underwent rigorous testing through a model selection process, wherein it was observed that the PASEL model achieved enhanced prediction accuracy (88.54%) while exhibiting lower error rates (39.56/25.69) compared to individual models.

VI. CONCLUSIONS AND FUTURE WORK

In this study, we have introduced an ensemble learning model for predicting demand in large-scale time series data, aiming to provide accurate and reliable forecasts for a company's products. Our approach involves an ETL process that integrates data from diverse sources into a centralized data lake on a cloud server, followed by the implementation of a parallel and distributed two-layer MapReduce algorithm on a cluster to aggregate transaction data into an abstract base table suitable for time series forecasting. Furthermore, the al-

gorithm automatically classifies time series into categories for utilization by the ensemble learning model. By leveraging the strengths of individual models, the ensemble model enhances overall performance and reliability, with the stacking method employed as the ”meta-model” at a higher level, capable of capturing a wider range of patterns in large-scale data and generating robust forecasts covering major patterns shown in Figure 6.

Our research findings indicate that the proposed method, successfully implemented in a cloud-based business environment, delivers accurate demand predictions and facilitates their effective execution. Through a comparative analysis of various time series analyses and machine learning techniques, we have demonstrated the advantages of our framework.

Looking ahead, future research endeavors aim to further enhance the accuracy of the meta-models in the ensemble learning approach. This will involve exploring a broader range of base models to diversify the ensemble and improve its performance. Additionally, we are interested in investigating novel ensemble methodologies, extending stacking to boosting techniques, and incorporating penalty, generalization, and regularization methods into the model-building process to further enhance forecasting performance. These investigations hold promise for advancing the field and refining the proposed approach.

REFERENCES

[1] Piotr F. Borowski. Digitization, digital twins, blockchain, and industry 4.0 as elements of management process in enterprises in the energy sector. Energies, 14, 2021.

[2] Digitalization within food supply chains to prevent food waste. drivers, barriers and collaboration practices. Industrial Marketing Management, 93:208220, 2021

[3] Victor Roudometof. Recovering the local: From glocalization to localization. Current Sociology, 67(6):801817, 2019.

[4] Forecasting and planning during a pandemic: Covid-19 growth rates, supply chain disruptions, and governmental decisions. European Journal of Operational Research, 290(1):99115, 2021.

[5] Christian Esposito and Massimo Ficco. Recent Developments on Security and Reliability in Large-Scale Data Processing with MapReduce. 2016.

[6] G.Peter Zhang. Time series forecasting using a hybrid arima and neural network model. Neurocomputing, 50:159175, 2003.

[7] Dynamic series of macroeconomic statistics of the russian federation. wage index, income index. March 2021.

[8] Denis Petrusevich. Improvement of time series forecasting quality by means of multiple models prediction averaging. 2021.

[9] A simple combination of univariate models. International Journal of Forecasting, 36(1):110115, 2020. M4 Competition.

[10] Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting. International Journal of Electrical Power Energy Systems, 136, 2022.

[11] Fildes R. E grio glu, E. A new bootstrapped hybrid artificial neural network approach for time series forecasting. Comput Econ, 2020.

[12] Thomas G Dietterich. Ensemble methods in machine learning. In International workshop on multiple classifier systems, pages 115. Springer, 2000.

[13] Rafael MO Cruz, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. Meta-des: A dynamic ensemble selection framework using meta-learning. Pattern recognition, 48(5):19251935, 2015.

[14] Mark J Van der Laan, Eric C Polley, and Alan E Hubbard. Super learner. Statistical applications in genetics and molecular biology, 6(1), 2007.

[15] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.