American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Demand Forecasting in Retail Business Using the Ensemble Machine Learning Framework - A Stacking Approach

Chukwuka Izuchukwu C. Obi, Ph.D*

101, 635 57 Avenue, SW, Calgary. Alberta. T2V0H5. Canada
Email: chukaobi231@gmail.com

Abstract

Demand forecasting is an integral component of organizational and supply chain operations. Its primary objective is to anticipate the future demand for products, thereby informing and refining strategic decisions related to inventory management. Despite the inherent complexities in achieving precise demand forecasts, many methodologies have been proposed for the establishment of efficient forecasting systems. Such methodologies encompass traditional statistical approaches, hybrid techniques, and advanced methodologies rooted in machine learning and deep learning. Scholarly investigations within demand forecasting indicate a growing preference for deep learning paradigms, especially when confronted with data characterized by multivariate attributes, high dimensionality, and unpredictable demand fluctuations. Given the research emphasis on the retail domain, a sector inherently marked by data that is both multivariate and possesses volatile demand characteristics, this study devised a Stacking Ensemble learner. A comparative assessment was subsequently conducted, evaluating this ensemble against a trained Multilayer Perceptron , a deep learning archetype. The evaluation utilized a historical sales dataset sourced from ten Walmart outlets across Texas, California, and Wisconsin. Evaluative metrics were employed to discern the forecasting proficiencies of the respective frameworks. The evaluation determined that the Stacking Ensemble model outperformed the Multilayer Perceptron in terms of accurate predictions.

***Keywords*:** Stacking Ensemble; Multilayer Perceptron ; Demand Forecasting.

Received: 7/30/2024
Accepted: 9/30/2024
Published: 10/10/2024

---

* Corresponding author.

1. Introduction

Business advancement hinges on understanding key elements that critically influence inventory management. Businesses dealing with product-handling must be cognizant of challenges in this area, such as controlling stock volumes, optimizing the utilization of inventory space, rectifying imprecise forecasting, handling superfluous and idle inventory, and ensuring timely customer delivery. [1] hold the viewpoint that retail businesses incessantly strive for more precise forecasts to augment the certainty of their decision-making process. Success in inventory management is contingent upon accurate demand forecasting, which entails a precise projection of future demand for a specific time interval [2].Demand forecasting significantly contributes to planning, capacity management, procurement, and inventory management, culminating in judicious decisions. Accurate forecasting boasts several benefits, such as reduced inventory costs, diminished overall expenditures, fewer stock-outs, and increased customer loyalty[4, 5]. Conversely, imprecise forecasting could lead to underestimation or overestimation of demand and stock-outs[3]. The challenge for businesses lies in developing a forecasting method that is accurate, transparent, reliable, and can account for external influences and unpredicted situations. Fortunately, the advent of technologies like machine learning can mitigate the risk of ineffective inventory management by providing accurate forecasts of product demand, allowing businesses to thrive and grow amidst the trials of inventory management.A foremost goal of any organization is the formulation of an accurate demand forecast. Demand forecasting could be short-term or long-term (i.e. extending the prediction beyond a year) based on the time frame Reference [3]. Two main strategies for demand forecasting are qualitative and quantitative methods, with quantitative methods leveraging mathematical computations to ensure accuracy, transparency, and reliability Reference [3].Machine learning, a branch of artificial intelligence, involves the creation of algorithms and models that empower computers to learn, predict, or make decisions based on data inputs [19]. [3] categorized all machine learning techniques as quantitative methods. Machine learning can unearth concealed patterns in a dataset, process colossal volumes of data rapidly and accurately, and adjust to new information and circumstances while preserving transparency and explainability [3]. Its aim is to create systems capable of enhancing their performance through experience without the need for explicit programming. Researchers have leveraged machine learning to devise data-driven solutions for various issues in business operations, including inventory management[8, 6, 7]. And as an integral part of inventory management, numerous machine learning approaches have been proposed by researchers to tackle this issue[3]. Artificial neural networks are recognized as the most potent technique for demand forecasting[10, 11, 9].To evaluate the effectiveness of advanced computational models in achieving high accuracy and reliability in predictive tasks, this study drew a comparison between the performance of a Stacking Ensemble machine learning framework and a multi-layer perceptron neural network application. Statistical metrics were employed to identify the method with the superior performance.

2. Literature Review

This segment presents the literature review for this study on demand forecasting in retail business. Traditionally, retailers have employed an array of forecasting strategies to project production and product demand, with numerous technologies existing to anticipate consumer demand. This literature review emphasizes on the prevailing methodologies for demand forecasting, their practical applications, and their intrinsic limitations. It scrutinizes the extant comprehension of the literature concerning demand forecasting techniques, particularly

within the retail sector. The objective is to construct a robust understanding of existing methodologies and to illustrate how the study's proposed methodology contributes to the body of knowledge pertaining to demand forecasting.

Enterprises are necessitated to anticipate demand, thereby facilitating projections for future necessities of their products or services. This foundational foresight underpins informed decision-making pertaining to production planning, inventory control, and pricing strategies, with the potential for minimizing costs and maximizing profits[4]. To gain a competitive edge in the dynamic landscape of the retail industry, corporations leverage cutting-edge technologies to project customer demand. The objective is to maintain a lean inventory while ensuring customer satisfaction, thereby curbing expenses related to distribution and storage[12]. Accurate demand projections are instrumental in planning for capacity expansion and efficient resource distribution[4]. Present demand forecasting methodologies have catalyzed the conception of a plethora of frameworks for anticipating future market needs, facilitating the formulation of suitable design and operational strategies[4]. Given the strides in technology and data analysis, increasingly sophisticated methodologies, including artificial intelligence, can be utilized to yield more accurate forecasts by considering a wider spectrum of variables. The existing demand forecasting methodologies that have been comprehensively researched can be classified into four primary categories.

2.1.Traditional statistical methods

Traditional statistical methodologies employ historical sales data to predict future product demand [4]. Two of the widely recognized traditional approaches are the ARIMA model and the various forms of Holt-Winters Exponential Smoothing. Both are explored in depth to show the advantages and disadvantages associated with inventory forecasting using traditional statistical approaches.

2.1.1.ARIMA model

The ARIMA (Autoregressive Integrated Moving Average) model, alternatively known as the Box-Jenkins method, was developed by Box and Jenkins in 1970 [40]. This assortment of time-domain models is routinely employed for estimating and forecasting time series exhibiting temporal correlation [40]. As noted by [4], the model is typically applied to non-stationary data wherein the summary statistics fluctuate over time.

ARIMA models are defined by the term ARIMA (p, d, q) encapsulating three elements: autoregressive (AR), integrated or differencing (I), and moving average (MA), corresponding to the order p, d, and q, respectively[40]. The autoregressive (AR) model, as posited by [21], utilizes a linear relationship between the output variable and its historical values, employing past values of the time series to project future values. The formula representing the AR model is presented in (1) where $Y[t]$ represents the value of the time series at time t, c denotes a constant, $\Phi[i]$ signifies the coefficient for the ith autoregressive term, $\varepsilon[t]$ indicates the residual or white noise at time t, and p represents the order of the autoregression determining the number of lagged values used in data modelling.

$$Y[t] = c + \Phi[1] * Y[t-1] + \Phi[2] * Y[t-2] + \ldots + \Phi[p] * Y[t-p] + \varepsilon[t] \quad (1)$$

To attain stationarity in a dataset, some degree of differencing operation is necessitated [21]. Differencing serves to eliminate or reduce trends and seasonality by removing variations in the level of a time series, and it can also help stabilize the variance of a time series [22].

The moving average (MA) model incorporates past forecast errors to enhance future projections[21]. The MA model is presented in (2) . Here Y[t] is the value of the time series at time t, μ denotes the mean of the series, ε[t] represents the residual or white noise at time t, and Θ[i] indicates the coefficient for the ith moving average term.

$$Y[t] = \mu + \varepsilon[t] + \Theta[1] * \varepsilon[t-1] + \Theta[2] * \varepsilon[t-2] + \ldots + \Theta[q] * \varepsilon[t-q] \quad (2)$$

By amalgamating differencing with autoregression and moving average modeling, a non-seasonal ARIMA model emerges[22]. The complete model can be depicted in (3):

$$Y[t] = \mu + \phi[1] * (Y[t-1] - \mu) + \phi[2] * (Y[t-2] - \mu) + \ldots + \phi[p] * (Y[t-p] - \mu) + \Theta[1] * \varepsilon[t-1] + \Theta[2] * \varepsilon[t-2] + \ldots + \Theta[q] * \varepsilon[t-q] + \varepsilon[t] \quad (3)$$

This is referred to as an ARIMA (p, d, q) model wherein p is the order of autoregression, d is the degree of the differencing component, and q is the order of the moving average. The coefficients $\phi[i]$ and $\Theta[i]$ are estimated from the data, and the residuals ε[t] are utilized to identify any residual patterns in the data that remain unaccounted for. The differencing component is used on the time series to mitigate non-stationarity, whereas the autoregression and moving average components are deployed to model the relationships within the stationary data.

2.2. Holt-Winters exponential smoothing

As per [22], the concept of exponential smoothing, initially proposed in the late 1950s, has underpinned several of the most successful forecasting methodologies. [20] contended that the Holt-Winters exponential smoothing technique is employed for time series data displaying both trends and seasonal variations. The approach encompasses the four forecasting methodologies.

2.2.1. Weighted average

Reference [20] elucidated that a weighted average is computed by summing up n numbers, each possessing a specific weight assigned by a weight function. The divisor in this computation is the aggregate of the n weights. A diverse range of weight functions such as linear, quadratic, cubic, logarithmic, and exponential may be utilized in the process. This methodology proves beneficial for time series forecasting as it allows for equalization of oscillations in the historical data during prediction phases.

2.2.2. Exponential smoothing

Reference [20] discussed the exponential smoothing (ES) method for predicting future values of a time series, achieved by computing a weighted average of all preceding values with diminishing weights from the most recent

to the earliest. A critical supposition when using ES is that more recent time series values bear more significance than earlier ones. However, [20] noted a significant limitation of ES forecasting methods, which is their ineffectiveness when a time series displays both trend and seasonal variations. The mathematical representation for exponential smoothing is (4), where $St[t]$ is the level estimation smoothed at time t, $Y[t]$ is the observed value at time t, and α denotes the smoothing parameter ranging between 0 and 1.

$$St[t] = \alpha * Y[t] + (1 - \alpha) * St[t-1] \quad (4)$$

2.2.3.Holt exponential smoothing

As per [22], in 1957, Holt expanded simple exponential smoothing to accommodate the forecasting of data with a trend. This variant of exponential smoothing incorporates both trend and level components. According to [20], the method is applicable for forecasting time series data exhibiting a trend but fails to be effective when the time series shows seasonal variations. The mathematical formula for Holt exponential smoothing is presented in (5) and (6), where $St[t]$ is the level estimation smoothed at time t, $Tt[t]$ is the trend estimation smoothed at time t, $Y[t]$ is the observed value at time t, and α and β are the smoothing parameters for level and trend components respectively, ranging between 0 and 1.

$$St[t] = \alpha * Y[t] + (1 - \alpha) * (St[t-1] + Tt[t-1]) \quad (5)$$

$$Tt[t] = \beta * (St[t] - St[t-1]) + (1 - \beta) * Tt[t-1] \quad (6)$$

The optimal values for α and β, which minimize the sum of squared errors between observed and smoothed estimates, can be determined through trial and error, grid search, or other advanced optimization techniques.

2.2.4.Holt-Winters Exponential Smoothing

Holt and Winters further extended the Holt exponential smoothing method to encompass seasonality, as pointed out by [22]. This method, requiring three smoothing parameters α, β, and γ, is suitable for forecasting time series data demonstrating seasonal variations[22]. [4] defined seasonality as repeating patterns in time series that occur at regular intervals. Depending on the seasonality characteristics, either the additive method (for time series with constant seasonal variations) or the multiplicative method (for time series where seasonal variations alter in proportion to the series level) is employed [22].

For time series data, Holt-Winters exponential smoothing can be mathematically represented in (7), (8), and (9) where $St[t]$ is the level estimation smoothed at time t, $Tt[t]$ is the trend estimation smoothed at time t, $Y[t]$ is the observed value at time t, m represents the number of seasons, and α, β, γ are the smoothing parameters for level and trend components respectively, both ranging between 0 and 1

$$St[t] = \alpha * Y[t] + (1 - \alpha) * (St[t-1] + Tt[t-1]) \quad . \quad (7)$$
$$Tt[t] = \beta * (St[t] - St[t-1]) + (1 - \beta) * Tt[t-1]$$

(7)

(8)

$$St[t] = \gamma * \frac{Y[t]}{S[t-m]} + (1 - \gamma) * (St[t] + Tt[t-1]) \quad (9)$$

As with Holt's method, the choice of smoothing parameters' values is vital, and can be selected through trial and error, grid search, or more sophisticated optimization techniques.

According to research, the Holt-Winters method has proven to be a reliable forecasting tool for predicting food product demand, delivering accurate results on par with more intricate forecasting models, thereby demonstrating its suitability for practical use[4]. Traditional statistical methods demonstrate efficacy when applied to univariate, stable data with a non-volatile demand pattern, as corroborated by numerous studies[13, 15, 14, 12].

2.3. Machine learning methods

Machine Learning (ML), as delineated by [23], represents a multidisciplinary field of study dedicated to empowering computational systems with the ability to self-learn without explicit programming. The focus of ML lies heavily on the utilization of algorithms that, over time, are capable of autonomously evolving and learning from data[4]. This objective endeavors to facilitate computer systems in executing independent actions and learning, eliminating the need for human intervention[23]. Throughout the learning process, the ML model is nurtured with historical data, thus allowing it to anticipate outcomes and extract insights from unseen data. As indicated by [4], the application of ML methodologies offers a notable advantage in precision, adaptability, and resilience when compared to traditional statistical approaches.

In the field of demand forecasting, prelabeled historical data is harnessed to construct regression models via supervised learning to prognosticate the quantity of products that consumers will purchase in the future. Regression models create a relationship between predictor and outcome variables to forecast new data points[4]. Most of the renowned supervised ML algorithms that are utilized to construct regression models are discussed below.

2.3.1. Linear regression

Linear regression, recognized as the most fundamental form of regression analysis, endeavors to ascertain a linear association between the dependent and independent variables by fitting a straight line to the observed data[24]. This methodology is categorized under the umbrella of supervised learning and is deployed to model and predict variables of a continuous nature. Linear regression is classified into the following sub-categories.

2.3.1.1.Simple linear regression

Simple linear regression illustrates the relationship between the independent variable (x) and dependent variable (y) via a linear equation that adheres to the structure, as shown in (10).

$$y = \beta 0 + \beta 1 * x \quad (10)$$

In this equation, $\beta 0$ signifies the intercept of the regression line, whereas $\beta 1$ corresponds to its slope[25].

2.3.1.2.Multivariate linear regression

Originating from the field of statistics, the multivariate linear regression technique functions to conceptualize the nexus between multiple independent variables and a dependent variable[25]. The general form of the multivariate linear regression model encompassing 'p' independent factors is:

$$Y = \beta[0] + \beta[1] * X[1] + \beta[2] * X[2] + \ldots + \beta[p] * X[p] + \varepsilon \quad (11)$$

Here, Y symbolizes the dependent variable, $X[i]$ corresponds to the 'i'th independent variable, $\beta[i]$ signifies the coefficient for the 'i'th independent variable, and ε is the residual or the error term. The coefficients $\beta[i]$ are calculated from the collected data through processes like least squares, and the residuals assist in pinpointing unresolved patterns in the data that warrant further exploration. The primary objective of multivariate linear regression is to discern the $\beta[i]$ values that optimally elucidate the correlation between the independent variables and the dependent variable[25]. Research indicates the applicability of multivariate linear regression in forecasting the demand for restaurants and restaurant chains[4].

2.3.2 Lasso regression

Lasso regression, a variant of regularized linear regression, utilizes the L1 penalty to diminish the size of the coefficients and execute proficient feature selection[26]. It is a form of linear regression that employs shrinkage, which compacts data values towards a central datum such as the mean[26]. Lasso regression is especially advantageous in dealing with models that exhibit substantial multicollinearity. The formulation of Lasso regression mirrors that of uncomplicated linear regression, albeit with the inclusion of a penalty term λ that modulates the size of the coefficients:

$$Y = \beta[0] + \beta[1] * X[1] + \beta[2] * X[2] + \ldots + \beta[p] * X[p] + \varepsilon \quad (12)$$

In this representation, Y stands for the dependent variable, $X[i]$ represents the 'i'th independent variable, $\beta[i]$ signifies the coefficient for the 'i'th independent variable, and ε is the residual or error term. The penalty term in Lasso regression is defined as follows:

$$Penalty = \lambda * \sum|\beta[i]| \quad (13)$$

In this equation, λ is a scalar that influences the intensity of the penalty (Glen, 2021). The more substantial the value of λ, the more the coefficients are diminished, leading to a lesser number of predictor variables being incorporated in the final model[26]. Therefore, Lasso regression serves as a straightforward and effective mechanism for feature selection and enhancing the interpretability of the model.

In a study centered on demand forecasting for high-dimensional retail products, multistage lasso regression emerged as an instrumental tool for feature selection and model estimation[4]. During the development of a forecasting model at the granular level of Stock Keeping Units (SKUs), pivotal variables may exhibit strong correlation with nonessential variables, thereby giving rise to issues of multicollinearity. Lasso regression can

mitigate this problem by isolating a solitary variable from an aggregation of highly interrelated variables[4]. As a result, lasso regression demonstrates effective management of multivariate and high-dimensional retail data.

2.3.3. Random forest

Random Forest is an ensemble learning method utilized for classification and regression tasks, which amalgamates the predictions rendered by multiple decision trees[27]. This algorithm was first presented in 2001 by Breiman and colleagues. It is predicated on the principles of bootstrapped aggregation (often referred to as bagging) and decision trees[27]. Within a Random Forest, each tree is constructed from a bootstrapped sample of the data. Furthermore, during each tree split, a random subset of the features is selected as splitting candidates. This strategy effectively mitigates the issue of overfitting, a common challenge inherent to decision tree algorithms[27].

Owing to its high precision, resilience to outliers and noisy data, along with its proficiency in managing non-linear relationships between features and target variables, Random Forest has exhibited commendable performance across a variety of application domains[37]. Given that the relationship between input features and the target is encapsulated by an ensemble of trees, the algorithm carries a higher computational expense in comparison to singular decision trees and can pose challenges in terms of interpretation.

Notwithstanding these constraints, Random Forest continues to be a prevalent algorithm within the realm of machine learning, and ongoing research endeavors are being pursued to augment its performance and interpretability. In an empirical study undertaken on a company's supply chain management platform with the objective of predicting customer demand for food products, Random Forest models outperformed alternate methods[28].

2.3.4 Extreme gradient boosting

Extreme Gradient Boosting (XGBoost) is an ensemble learning algorithm that utilizes gradient-boosted decision trees, gaining considerable recognition within the machine learning domain due to its exceptional efficiency and adaptability[4]. The underlying principle of XGBoost is the iterative fitting of an ensemble of rudimentary models, such as decision trees, to enhance the overarching prediction[27]. The algorithm constructs trees in a greedy fashion, sequentially adding trees that yield the most significant reduction in loss[27]. XGBoost, like any other machine learning algorithm, presents certain limitations. It can be computationally intensive, particularly when dealing with extensive datasets. This necessitates a substantial allocation of memory and processing capability, which may pose a constraint for certain applications. Identifying the optimal set of hyperparameters for XGBoost can be time-consuming and necessitates a comprehensive understanding of the algorithm. XGBoost incorporates a multitude of hyperparameters that must be meticulously tuned to optimize performance.

Empirical evidence from numerous studies attests to the efficacy of XGBoost in a variety of real-world applications, including demand forecasting. For instance, research conducted by [30] illustrated the proficiency of XGBoost in time series prediction pertinent to electricity load forecasting, underscored by efficient utilization of memory resources and computational time. Further, in an investigation aimed at forecasting sales for Big Mart, XGBoost exhibited superior performance compared to other machine learning algorithms[29].

2.3.5 Light gradient boosting machine

Light Gradient Boosting Machine (LightGBM), an effective and scalable gradient boosting framework based on decision trees, was developed by Microsoft and is openly accessible[31]. According to existing literature, LightGBM serves as a more expeditious alternative to other well-established gradient boosting frameworks, while simultaneously preserving or even augmenting prediction accuracy[31]. The algorithm operates by successively integrating decision trees into the model, each tree engineered to correct the predecessors' inaccuracies[31]. The comprehensive model is acquired by amalgamating the weighted predictions of each tree.

LightGBM's primary innovation is the incorporation of Gradient-based One-Side Sampling (GOSS), which escalates training efficiency. GOSS selectively gives priority to instances with larger gradients, concurrently discarding instances with small gradients during the tree-building process[35]. This strategy curtails the number of instances requiring evaluation during model training without forfeiting its precision. LightGBM also deploys additional techniques to enhance training speed, including histogram-based binning and feature parallelism. Feature parallelism facilitates the distribution of computation across several cores, while histogram-based binning diminishes the count of discrete values that need to be contemplated for each feature.

However, LightGBM also comes with certain limitations. There is a profusion of hyperparameters in LightGBM that necessitate meticulous tuning for the attainment of optimal performance. The algorithm's extreme sensitivity to hyperparameter selections can complicate the process of determining the ideal hyperparameter values.

As per a comparative study conducted within the context of a multinational retail corporation, LightGBM outperformed Long-Short Term Memory (LSTM) in terms of statistical efficacy in demand forecasting[31].

Broadly speaking, it has been substantiated through multiple studies that machine learning methodologies are apt for dealing with unstable and high-dimensional datasets and are particularly potent in volatile demand circumstances[16, 5, 18,17].

2.4. Deep Learning Methods

Deep learning, a sub-branch of machine learning, utilizes artificial neural networks to tackle complex issues. It has acquired widespread acceptance owing to its capabilities in handling vast data sets and modelling non-linear data, thereby yielding more precise predictions compared to conventional statistical methodologies[4]. The spectrum of deep learning research incorporates the advancement of more effective and precise neural network architectures, refining training algorithms and optimization strategies, and delving into novel application fields. Its structural design is reminiscent of the human brain, replete with interconnected artificial neurons[19].

The architecture of a neural network encompasses its comprehensive layout, encapsulating the total number, configuration, and count of neurons per layer[10]. The design of a neural network's architecture is meticulously tailored to suit the specific issue it is intended to resolve. The most common neural network architectures are Multilayer Perceptron and Recurrent Neural Network.

2.4.1. Multilayer perceptron (MLP)

The Multilayer Perceptron (MLP) is classified as a feedforward neural network, embodying an input layer, one or more intermediary or hidden layers, and an output layer. Within this network, each neuron is integrally connected to every other neuron present in its neighboring layers, with corresponding weights[27]. The network is purposed to comprehend complex non-linear representations of the input data, with the final hidden layer's output signifying this feature extraction process[32]. Subsequently, the output layer, functioning as a singular-layered perceptron, maps these extracted features onto the desired output target. The network employs activation functions and forward propagation to compute predictions, while biases and weights are adjusted via backward propagation to minimize errors. This approach aids in identifying the optimal parameters for the prediction[27]. The choice of the activation function considerably influences the training efficacy of the network[32].

A study on sales prediction conducted by [33] discusses the use of neural networks for sales forecasting, particularly in scenarios where companies have limited historical data due to factors such as changes in warehouse structure. The study emphasizes the challenges of forecasting sales with small datasets and proposes the use of a multilayer perceptron for making sales predictions. The authors found that variations in learning rates did not significantly affect computing time, and the model achieved validation errors below five percent, demonstrating the potential effectiveness of neural networks in sales forecasting even with limited data. The study's findings suggest that neural networks, due to their flexibility and independence from traditional statistical assumptions, can be a valuable tool for sales forecasting.

2.4.2. Recurrent neural network

Recurrent Neural Networks (RNNs) are specialized neural networks designed for sequential data like time series or natural language. Unlike conventional neural networks that process fixed-size inputs, RNNs have a feedback mechanism that retains information from previous inputs, allowing them to detect patterns across sequences by maintaining a "hidden state" [32]. RNNs are trained using Backpropagation Through Time (BPTT), a version of backpropagation adapted for temporal data. However, they face issues like vanishing and exploding gradients. Vanishing gradients hinder learning by making weight updates insignificant, while exploding gradients lead to instability with excessively large weights. To mitigate these issues, reducing the number of hidden layers can help simplify the model. Notably, the vanishing gradient problem limits RNNs' ability to capture long-term dependencies[19,1].

In 1997, Hochreiter and Schmidhuber proposed the long short-term memory network (LSTM) as a remedy to the vanishing gradient problem[10]. The primary goal was to handle the challenge of long-term dependencies[1]. In essence, if the RNN model is required to consider an older state to inform the current prediction, it might fail to produce accurate predictions. To rectify this, LSTMs incorporate three gated 'cells' within the hidden layers of the neural network: an input gate, an output gate, and a forget gate. These gates govern the flow of necessary information to forecast the network's output[1].

As stated by [1], LSTM is deployed for retail forecasting owing to its proven efficacy in dealing with both linear

and non-linear time series, obviating the necessity to segregate the time series into linear and non-linear constituents. Further, [1] highlighted that LSTM networks are adept at handling challenges such as unstable data and fluctuating demand situations.

Neural networks are typically compatible with high-dimensional and multivariate datasets, demonstrating their capacity to effectively manage volatile demand scenarios[16, 4, 5, 18,17]. When their parameters are precisely calibrated, the accuracy of neural network predictions can be exceptionally high [4].

2.5. Hybrid methods

Hybrid machine learning refers to an enhanced workflow that amalgamates diverse algorithms, processes, or techniques from various knowledge domains or application areas with the objective of mutual complementarity[34]. Traditional machine learning methodologies can be hybridized in a limitless array of ways, thereby facilitating the creation of innovative hybrid models in a multitude of ways. Selecting an optimal model or method for implementation can prove to be a complex and time-consuming endeavor, given that varying scenarios may necessitate disparate approaches[4].

Multiple models are ordinarily trained and statistically evaluated to ascertain the most accurate solution for a specific task. However, each model may come with its own set of limitations, either inherent to the model or tied to the data utilized[38]. As there exists no universally applicable solution, a singular model might not suffice for all tasks. For instance, certain machine learning methodologies might excel with noisy data yet falter with high-dimensional data, while others may adeptly handle high-dimensional data but not sparse data. In such instances, hybrid machine learning methods may be employed.

3. Methods

3.1. Data collection

This study utilized the Walmart M5 dataset available on the Kaggle platform, which contains the unit sales of 3,049 products sold in the United States from 2011 to 2016 and is categorized into three product categories (Hobbies, Foods, and Household) and seven product departments. Ten stores in California, Texas, and Wisconsin sold the products, and the University of Nicosia provided the dataset. The figure below illustrates the organization of this dataset.

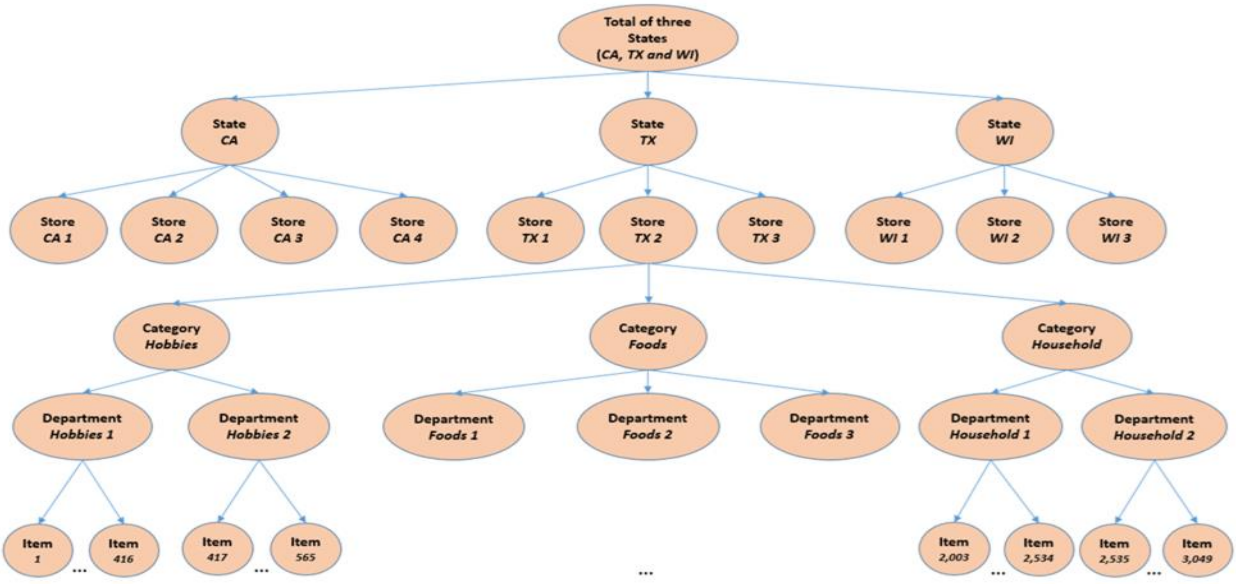

Figure 1: Organization of the dataset

3.2. Data preprocessing

This stage of the study involved converting raw data into an interpretable format. A range of data processing methodologies, including feature engineering/extraction, feature selection, and data normalization, were employed. The product of data preprocessing was the finalized sample used for model training and testing. The machine learning models, inclusive of the Multilayer Perceptron model, were trained using these extracted variables.

Table 1: Variables and their description

Variables	Description
state_id	The State where the store is located.
d	The number of units sold at day i, starting from 2011-01-29.
month	The month of the date.
year	The year of the date.
snap_CA, snap_TX, and snap_WI	A variable (yes or no) indicating whether the stores of CA, TX or WI allow SNAP purchases on the examined date. Yes, indicates that SNAP purchases are allowed.
sell_price	The price of the product for the given week/store.
simple_moving_average	The simple moving average (SMA) is a calculation that takes the average of a specific number of data points over a set period. It's used to smooth out short-

	term fluctuations and highlight longer-term trends. A 28-day SMA used for this study is the average of the last 28 days' values calculated and updated as each new day is added.
cum_moving_average	The cumulative moving average (CMA) is the average of all data points up to the current point. It recalculates the average each time a new data point is added, considering all prior data.
exp_weighted_moving_average	The exponentially weighted moving average (EWMA) gives more weight to recent data points, making it more responsive to changes compared to a simple moving average. It reduces the impact of older data points using an exponential decay factor.
total_price	total_price represents the aggregate price, which is computed through the multiplication of two variables: sales and sell_price.
sales	The dependent variable

3.3. Model training

This study's endeavor aims to construct a Stacking Ensemble framework that utilizes base models derived from Random Forest and LightGBM algorithms, with the objective of comparing its performance against a Multilayer Perceptron model - an application of deep learning. The purpose is to ascertain which method yields the most accurate forecast. A Stacking Ensemble represents a machine learning technique that employs multiple learning algorithms with the objective of optimizing predictive capability[39]. The schematic representation of the Stacking Ensemble framework implemented in this study is exhibited in the figure below.

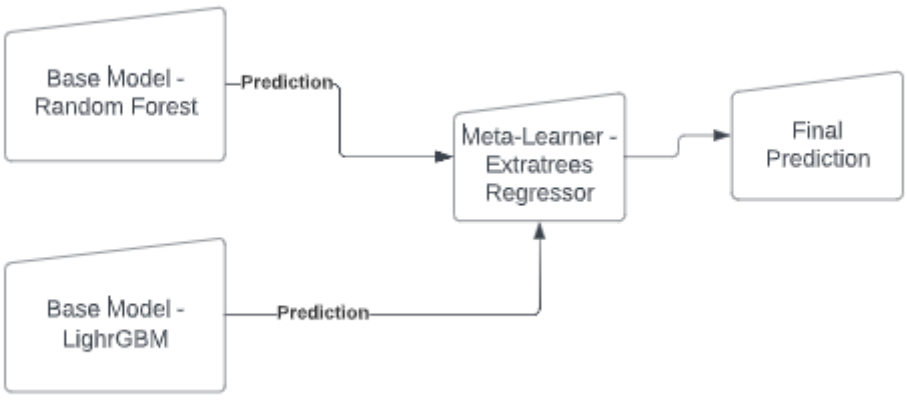

Figure 2: Study's stacking ensemble framework

In the context of this study, the framework was executed in two phases: initially, all learning algorithms, namely Random Forest and LightGBM, were trained on historical sales data after rigorous data preprocessing. Subsequently, a meta-learner algorithm, in this case, the Extremely Randomized Trees (Extra Trees) regression algorithm, was used to generate a final prediction based on all the forecasts yielded by the algorithms trained in the initial phase. The Random Forest algorithm employed in this study was sourced from the open-source Scikit-Learn library and the LightGBM algorithm was obtained from the Microsoft library, which is also open source. Additionally, the Multilayer Perceptron network was trained using TensorFlow, an open-source framework for deep learning networks. The study intentionally avoided the use of proprietary or commercial software, opting instead for open-source alternatives to promote transparency, reproducibility, and flexibility in the research process. This approach enabled a more thorough exploration and customization of the algorithms, thereby enhancing the understanding of the underlying methodologies and their influence on the study's outcomes.

3.4. Model evaluation

The discrepancy between the actual and predicted values is often referred to as the "residual." Evaluating the difference between these values is pivotal in assessing the accuracy and reliability of a predictive model. A model that generates predicted values close to the actual values is typically considered to have a good fit, while large differences indicate potential issues with the model's assumptions, variables, or parameters. To access the performance of the frameworks, the study evaluated how closely the forecasted values were to the actual values for the frameworks investigated.

4. Limitations

The machine learning frameworks developed within this study excluded supply chain variables and processes. The integration of supply chain facets, such as supplier selection, risk prediction, transportation, production, and storage, would have substantially inflated the scope of the study, rendering it unmanageably broad.The central objective of this study is to propose a machine learning framework aimed at enhancing the precision of demand forecasting within the retail sector, a specific segment of supply chain management. The research did not aspire to devise a comprehensive framework encapsulating all variables and processes inherent to the supply chain.

5. Findings

The study's intent was to juxtapose the statistical efficacy of a Stacking Ensemble framework, underpinned by the Random Forest and LightGBM models, with that of a Multilayer Perceptron framework, a neural network application. The aim was to discern which structure exhibited superior predictive precision through rigorous statistical evaluations.

5.1. Predictive analytics (Actual vs Predicted Values)

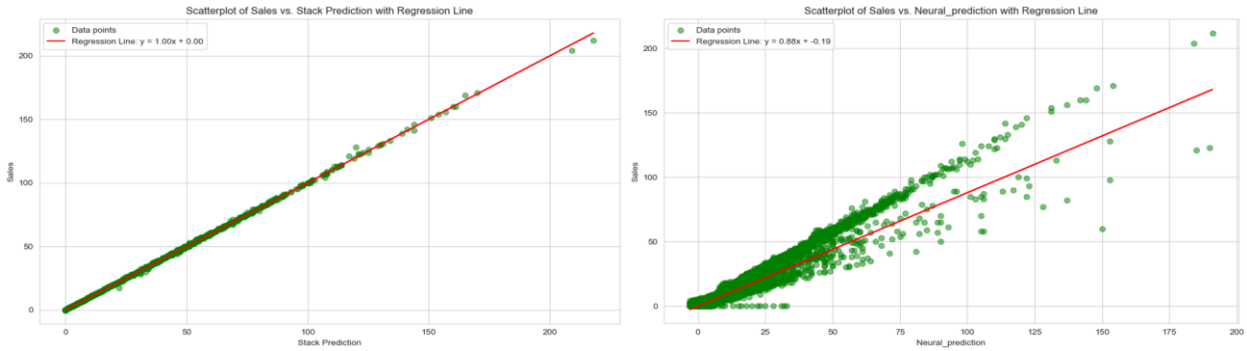

Figure 3: Actual vs predicted values plot

The presented plots offer a comparative analysis of actual sales against predictions from the Stacking Ensemble and the Multilayer Perceptron models developed in this study.

In the left plot, illustrating the Stacking Ensemble model, the predictions are represented on the horizontal axis while the actual sales are depicted on the vertical axis. The data points, marked in green, exhibit a strong alignment with the identity line y = x, which is also visualized as a red regression line. This perfect alignment indicates that the predictions precisely match the actual sales values. The tight clustering of the points around this line demonstrates the model's high accuracy and effectiveness, suggesting a strong linear correlation between predicted and actual sales values.

Conversely, the right plot, which details the Multilayer Perceptron model, shows a slight deviation from the ideal prediction line. Here, the regression line is characterized by the equation y = 0.88x + 0.19, indicating a consistent underestimation in the model's predictions, where the actual sales tend to be higher than the predicted values. The slope of 0.88 suggests that the model predicts a lower value for each unit increase in actual sales, coupled with a small positive intercept of 0.19, pointing towards a systematic bias in predictions. The scatter of points is broader than in the Stacking Ensemble model, reflecting a higher variability in the predictions and higher error rates.

The Table below presents a comparative analysis in numbers of the two frameworks developed and assessed in this study. This analysis aimed to discern the distinctions between the frameworks and unequivocally ascertain the superior performing model

Table 2: Prediction accuracy of stacking ensemble and multilayer frameworks

	Accurate	**Underestimate**	**Overestimate**
Stacking Ensemble	99.98394	0.00882	0.00724
Multilayer Perceptron	66.16471	32.75772	1.07757

It is evident from the table that:

1. **Accurate Predictions**: The Stacking Ensemble model shows a markedly superior performance,

with an accuracy rate of 99.98394% compared to the Multilayer Perceptron's 66.16471%. This represents a significant difference of approximately 33.82 percentage points in favor of the Stacking Ensemble model.

2. **Underestimations**: The Multilayer Perceptron has a much higher rate of underestimation, registering 32.75772%, whereas the Stacking Ensemble model's rate is almost negligible at 0.00882%. This is a substantial difference of approximately 32.75 percentage points.
3. **Overestimations**: Both models exhibit relatively low overestimation rates. The Multilayer Perceptron's rate stands at 1.07757%, which is notably higher than the Stacking Ensemble's rate of 0.00724%. Nonetheless, the difference between the two is just about 1.07 percentage points.

The Stacking Ensemble model outperformed the Multilayer Perceptron in terms of accurate predictions.

6. Conclusion and Recommendations

Considering the conducted research, this study advocates for the adoption of the Stacking Ensemble Framework constructed herein for demand forecasting within the retail sector. This endorsement arises from the rigorous analysis detailed within this segment. Upon perusal of existing literature, it was discerned that a significant portion of scholars concur that applications utilizing neural networks yield the most precise demand forecasts. However, the evaluation of the neural network against the formulated Stacking Ensemble Framework in this study challenges this prevailing assertion.

The exceptional predictive prowess of the proposed Stacking Ensemble framework can be attributed to its foundational methodology. This structure incorporates the Random Forest and LightGBM models as foundational layers and employs the Extra Trees model as the meta-learner. Notably, both the Random Forest and Extra Trees algorithms are encapsulated within the bagging algorithms category, while the LightGBM algorithm is classified under boosting algorithms.

Bagging, an acronym for "Bootstrap Aggregating," is an ensemble machine learning strategy that augments model precision by concurrently training multiple models on varied data subsets and subsequently amalgamating their outputs[19]. This approach encompasses generating multiple random samples from the primary dataset, with replacement, and then independently training a distinct model on each. Predictive outputs are unified either by averaging for regression tasks or majority consensus for classification tasks. Such a methodology invariably yields a more resilient model less prone to overfitting. Furthermore, bagging permits an "Out-of-Bag" error estimation, leveraging unutilized data from each subset to assess model efficacy without the necessity for an isolated validation set[36]. Fundamentally, bagging enhances model generalization by mitigating discrepancies from singular models via an aggregation process.

Conversely, boosting represents an ensemble machine learning modality that sequentially trains models to optimize their efficacy[19]. Commencing with homogenized data point weights, an initial model is trained, and its errors gauged. Data points predicted inaccurately are subsequently prioritized in subsequent training iterations.

Ensuing models concentrate on these more challenging instances, perpetuating this methodology either for a predetermined number of cycles or until a specified performance benchmark is attained. The culminating prediction constitutes a weighted amalgamation of all model results. Boosting continually hones its models by emphasizing intricate portions of the dataset, striving to diminish the overall bias of the ensemble.

The inherent methodologies of these algorithmic categories in enhancing predictive performance are undeniably robust. Integrating a model from each algorithmic category and synthesizing their predictions through the Extremely Randomized Trees (Extra Trees) model amplifies this robustness. Based on these underpinnings, this study asserts with confidence that replicating the framework's development procedure from this study and applying it to any dataset, especially those with demand fluctuations and multivariate attributes from a retail organization, assures superior predictive outcomes.

6.1 Recommendations for future research

In the literature examined during this study, a prevalent consensus amongst many scholars suggests that neural network applications yield superior accuracy in demand forecasting. This study proposed an ensemble framework, which was subsequently evaluated against a Multilayer Perceptron model—a neural network application—using the same historical sales dataset. The comparative analysis demonstrably showed that the proposed ensemble framework statistically surpassed the performance of the Multilayer Perceptron model, thus challenging the widely accepted scholarly stance in this field. To further this line of research, it is recommended that future evaluations benchmark the ensemble framework developed in this study against the Long Short-Term Memory (LSTM) model, another prominent neural network application, to determine superiority in predictive ability, utilizing the same historical sales dataset.

Long Short-Term Memory (LSTM) networks, a subtype of recurrent neural networks, are adeptly designed to capture long-term dependencies in sequential data. This intrinsic ability renders them particularly suitable for time series forecasting. The LSTM architecture employs a unique gated mechanism, comprising input, forget, and output gates, which facilitates the selective retention of relevant historical data. Additionally, its capacity to model intricate non-linear relationships and facilitate end-to-end learning positions it as a preferred choice in various sectors for forecasting tasks, such as sales forecasting. Thus, LSTMs present a potent and academically recognized approach for time series forecasting, especially in contexts characterized by complex patterns and long-term dependencies.

References

[1] S. Punia, K. Nikolopoulos, S. P. Singh, J. K. Madaan, and K. Litsiou, "Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail," *International Journal of Production Research*, vol. 58, no. 16, pp. 4964–4979, Mar. 2020, doi: 10.1080/00207543.2020.1735666.

[2] Kiefer, Grimm, Bauer, and Van, "Demand forecasting intermittent and lumpy time series: Comparing statistical, machine learning and deep learning methods," *Proceedings of the Annual Hawaii*

International Conference on System Sciences, 2021, doi: 10.24251/hicss.2021.172.

[3] Farzana and Prakash, "Machine learning in demand forecasting - a review," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3733548.

[4] Ingle, Bakliwal, Jain, Singh, Kale, and Chhajed, "Demand forecasting : Literature review on various methodologies," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Jul. 2021, doi: 10.1109/icccnt51525.2021.9580139.

[5] Kilimci *et al.*, "An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain," *Complexity*, vol. 2019, pp. 1–15, Mar. 2019, doi: 10.1155/2019/9067367.

[6] Y.-H. Kuo and A. Kusiak, "From data to big data in production research: the past and future trends," *International Journal of Production Research*, vol. 57, no. 15–16, pp. 4828–4853, Mar. 2018, doi: 10.1080/00207543.2018.1443230.

[7] B. Shen, T.-M. Choi, and S. Minner, "A review on supply chain contracting with information considerations: information updating and information asymmetry," *International Journal of Production Research*, vol. 57, no. 15–16, pp. 4898–4936, May 2018, doi: 10.1080/00207543.2018.1467062.

[8] S. Huang, A. Potter, and D. Eyers, "Social media in operations and supply chain management: State-of-the-Art and research directions," *International Journal of Production Research*, vol. 58, no. 6, pp. 1893–1925, Dec. 2019, doi: 10.1080/00207543.2019.1702228.

[9] Yeasmin, Amin, and Tosarkani, "Machine learning techniques for grocery sales forecasting by analyzing historical data," *Learning and Analytics in Intelligent Systems*, pp. 21–36, Dec. 2021, doi: 10.1007/978-3-030-85383-9_2.

[10] Benidis *et al.*, "Deep learning for time series forecasting: Tutorial and literature survey," *ACM Computing Surveys*, May 2022, doi: 10.1145/3533382.

[11] D. Wiyanti, I. Kharisudin, A. Setiawan, and A. Nugroho, "Machine-learning algorithm for demand forecasting problem," *Journal of Physics: Conference Series*, vol. 1918, no. 4, p. 042012, Jun. 2021, doi: 10.1088/1742-6596/1918/4/042012.

[12] Wang, Liu, and Liu, "A selection of advanced technologies for demand forecasting in the retail industry," *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)*, Mar. 2019, doi: 10.1109/icbda.2019.8713196.

[13] Ghosh, "Forecasting of demand using ARIMA model," *American Journal of Applied Mathematics and Computing*, vol. 1, no. 2, 2020, [Online]. Available: https://ajamc.smartsociety.org/wp-

content/uploads/2020/09/paper4.pdf

[14] Silva, Figueiredo, and Braga, "Demand forecasting: A case study in the food industry," *Computational Science and Its Applications – ICCSA 2019*, pp. 50–63, 2019, doi: 10.1007/978-3-030-24302-9_5.

[15] R. Priyadarshi, A. Panigrahi, S. Routroy, and G. K. Garg, "Demand forecasting at retail stage for selected vegetables: a performance analysis," *Journal of Modelling in Management*, vol. 14, no. 4, pp. 1042–1063, Oct. 2019, doi: 10.1108/jm2-11-2018-0192.

[16] Aishwarya, Aishwarya, Kumari, Mishra, and Rashmi, "Food demand prediction using machine learning," *International Research Journal of Engineering and Technology*, vol. 7, no. 6, 2020, [Online]. Available: https://www.irjet.net/archives/V7/i6/IRJET-V7I6686.pdf

[17] Ramya and Vedavathi, "An advanced sales forecasting using machine learning algorithm.," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 5, 2020, [Online]. Available: https://www.ijisrt.com/assets/upload/files/IJISRT20MAY134.pdf

[18] Lakshmanan, Vivek Raja, and Kalathiappan, "Sales demand forecasting using LSTM network," *Advances in Intelligent Systems and Computing*, pp. 125–132, 2020, doi: 10.1007/978-981-15-0199-9_11.

[19] IBM Cloud Education, "Machine learning," *IBM Cloud Education*, Jul. 15, 2020. https://www.ibm.com/cloud/learn/machine-learning

[20] Date, "Holt-Winters exponential smoothing," *Time Series Analysis, Regression and Forecasting*, Sep. 30, 2021. https://timeseriesreasoning.com/contents/holt-winters-exponential-smoothing/

[21] Hebbar, "Time series forecasting with ARIMA model in python for temperature prediction," *Medium*, Dec. 15, 2021. https://medium.com/swlh/temperature-forecasting-with-arima-model-in-python-427b2d3bcb53

[22] Hyndman and Athanasopoulos, *Forecasting: Principles and practice*. OTexts, 2018.

[23] D. Painuli, D. Mishra, S. Bhardwaj, and M. Aggarwal, "Forecast and prediction of COVID-19 using machine learning," *Data Science for COVID-19*, pp. 381–397, 2021, doi: 10.1016/b978-0-12-824536-1.00027-7.

[24] Ray, "A quick review of machine learning algorithms," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Feb. 2019, doi: 10.1109/comitcon.2019.8862451.

[25] Maulud and Abdulazeez, "A review on linear regression comprehensive in machine learning," *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, Dec. 2020, doi:

10.38094/jastt1457.

[26] S. Glen, "Lasso regression: Simple definition," *Statistics How To*, Apr. 27, 2021. https://www.statisticshowto.com/lasso-regression/

[27] Ş. Özlem and O. F. Tan, "Predicting cash holdings using supervised machine learning algorithms," *Financial Innovation*, vol. 8, no. 1, May 2022, doi: 10.1186/s40854-022-00351-8.

[28] N. Vairagade, D. Logofatu, F. Leon, and F. Muharemi, "Demand forecasting using random forest and artificial neural network for supply chain management," *Computational Collective Intelligence*, pp. 328–339, 2019, doi: 10.1007/978-3-030-28377-3_27.

[29] G. Behera and N. Nain, "A comparative study of big mart sales prediction," *Communications in Computer and Information Science*, pp. 421–432, 2020, doi: 10.1007/978-981-15-4015-8_37.

[30] R. A. Abbasi, N. Javaid, M. N. J. Ghuman, Z. A. Khan, S. Ur Rehman, and Amanullah, "Short term load forecasting using XGBoost," *Advances in Intelligent Systems and Computing*, pp. 1120–1131, 2019, doi: 10.1007/978-3-030-15035-8_108.

[31] P. Saha, N. Gudheniya, R. Mitra, D. Das, S. Narayana, and M. K. Tiwari, "Demand Forecasting of a Multinational Retail Company using Deep Learning Frameworks," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 395–399, 2022, doi: 10.1016/j.ifacol.2022.09.425.

[32] Benidis *et al.*, "Neural forecasting: Introduction and literature overview.," *arXiv: Learning*, Apr. 2020, [Online]. Available: https://deepai.org/publication/neural-forecasting-introduction-and-literature-overview

[33] R. M. Cantón Croda, D. E. Gibaja Romero, and S. O. Caballero Morales, "Sales prediction through neural networks for a small dataset," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 4, p. 35, 2019, doi: 10.9781/ijimai.2018.04.003.

[34] F. Anifowose, "Hybrid Machine Learning Explained in Nontechnical Terms," *JPT*, Feb. 06, 2020. https://jpt.spe.org/hybrid-machine-learning-explained-nontechnical-terms

[35] A. Sharma, "What makes LightGBM lightning fast? - towards data science," *Medium*, Nov. 14, 2022. https://towardsdatascience.com/what-makes-lightgbm-lightning-fast-a27cf0d9785e

[36] Z. Khan, N. Gul, N. Faiz, A. Gul, W. Adler, and B. Lausen, "Optimal trees selection for classification via Out-of-Bag assessment and Sub-Bagging," *IEEE Access*, vol. 9, pp. 28591–28607, Jan. 2021, doi: 10.1109/access.2021.3055992.

[37] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *The Stata Journal*, vol. 20, no. 1, pp. 3–29, Mar. 2020, doi: 10.1177/1536867x20909688.

[38] Y. Nie, P. Jiang, and H. Zhang, "A novel hybrid model based on combined preprocessing method and advanced optimization algorithm for power load forecasting," *Applied Soft Computing*, vol. 97, p. 106809, Dec. 2020, doi: 10.1016/j.asoc.2020.106809.

[39] M. G. Meharie, W. J. Mengesha, Z. A. Gariy, and R. N. Mutuku, "Application of stacking ensemble machine learning algorithm in predicting the cost of highway construction projects," *Engineering, Construction and Architectural Management*, vol. 29, no. 7, pp. 2836–2853, Jun. 2021, doi: 10.1108/ecam-02-2020-0128.

[40] Y. Lai and D. A. Dzombak, "Use of the Autoregressive Integrated Moving Average (ARIMA) model to forecast Near-Term regional temperature and precipitation," *Weather and Forecasting*, vol. 35, no. 3, pp. 959–976, Apr. 2020, doi: 10.1175/waf-d-19-0158.1.