International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 4, Issue 1, 31-39, 2023
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V4I1P104

Original Article

Machine Learning Algorithms in Supply Chain Forecasting: Accuracy, Efficiency, and Scalability Perspectives

Venkatesh Prabu Parthasarathy
President and Key Executive MBA (Pepperdine Univ.) Supply Chain Transformation | Digital Transformation, AI Implementation |IOT/ML Implementation Leader, Lake Forest, California, USA.

Abstract *- For today's global trade market, supply chain forecasting assists companies with productivity, lessening expenses, and satisfying buyers. Traditional methods of forecasting may not be sufficient when facing the challenging data in today's supply chains. ML is now an important tool because it discovers patterns in data and then makes predictions to address these issues. The study examines the use of a number of ML algorithms in supply chain forecasting by considering three main points. Accuracy, efficiency, and scalability. It measures the abilities of algorithms such as Linear Regression, SVM, Decision Trees, Random Forests, GBM, and LSTM networks in Deep Learning. The approach evaluates the effectiveness of forecasts by using records from sales and logistics in different industries. It turns out that while easy-to-understand models are quicker to train and easy to interpret, more advanced LSTM models perform better on data that changes often. We also point out that choosing an algorithm should depend on the situation in a supply chain and the necessary amount of computation. It explains a framework used for the project, which includes collecting, cleaning, selecting features, training models, validating them, and using them in practice. All in all, the study helps professionals and specialists in data science plan and apply machine learning forecasting to their supply chain.*

Keywords *- Supply Chain Forecasting, Machine Learning, LSTM, Random Forest, Scalability, Accuracy, Efficiency.*

1. Introduction

1.1. Importance of Machine Learning Algorithms in Supply Chain Forecasting

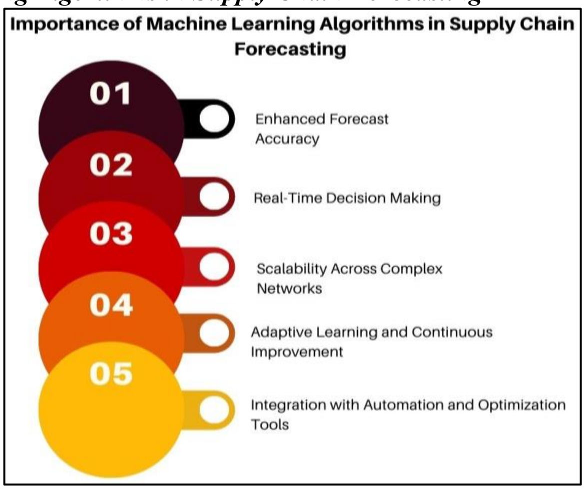

Fig 1: Importance of Machine Learning Algorithms in Supply Chain Forecasting

- **Enhanced Forecast Accuracy:** Some methods used to predict the future often do not handle the issues of complexity and variation in today's supply chains. [1-4] Thanks to Machine Learning, the forecast can be much more precise since it models strong links between variables, learns from past instances, and processes multidimensional data. Using accurate forecasting, businesses are able to manage their stock so that they avoid both shortages and extra waste.
- **Real-Time Decision Making:** ML enables analyses and predictions to happen constantly using data from IoT sensors, ERP systems, and POS terminals. As a result, supply chain managers are able to cope with changes or fluctuations in the market, making the whole process more flexible.
- **Scalability Across Complex Networks:** More and more, supply chains cover the whole world and include a diverse group of suppliers, distributors, and logistics partners. ML algorithms, such as ensemble and deep learning models,

can support many types of data coming from a variety of places. They gather information from both organized and unorganized data on the network and provide results that traditional analysis often fell short of delivering.

- **Adaptive Learning and Continuous Improvement:** One important feature of ML is that it can continually improve as it analyzes more data. When the market or customers' needs change, ML models can be retrained accordingly. As a result, forecasting models are continuously updated, making it less often necessary for them to be readjusted manually.
- **Including Automation and Optimization in the Process:** You can include ML-based forecasts in supply chain processes, for example in demand planning, warehouse management, and route planning. With this integration, the company can optimize every part of the process, including inventory, prices, and resource usage, leading to better performance and happier customers.

1.2. Challenges in Supply Chain Forecasting

Fig 2: Challenges in Supply Chain Forecasting

- **Data Sparsity and Noise:** Supply chain data is sometimes unfinished, varied, and unclear because of problems like missing fields, invalid data entry, or disturbed systems for gathering data. Such problems can confuse the computer algorithms that look for patterns. It is necessary to carefully construct ML algorithms that can deal with missing data, remove odd data points, and stay strong in the face of imperfect data.
- **High-Dimensionality:** There is an enormous amount of data in modern supply chains, covering all areas, from the kinds of products to stockpiles, how suppliers act, and how customers behave. When there are too many variables, the risk of overfitting and slow treading can occur if not handled well. It is important to use feature selection, reduce the dimension of data, or apply regularization methods to guarantee a model that is both efficient and interpretable.
- **Multi-Modal Data:** Supply chain forecasting now makes use of various sources like transactional data, data from sensors in IoT devices, news reports, and information about the weather or the market. Working with so many different types of data, where the formats and importance of data are not the same, causes big technical challenges in designing the model and preprocessing the data.
- **Seasonal and Trend Variations:** Holidays, fiscal quarters, inflation, and demand growth are all components that lead to seasonality in supply chains. Taking these changes into account is necessary for good forecasting. On the other hand, standard models usually miss out on these variations without automatic detection, whereas ML models get all the information needed from a lot of past data.

1.3. Accuracy, Efficiency, and Scalability Perspectives

In the context of supply chain forecasting, there are three important things to look at when choosing and putting into use predictive models: accuracy, efficiency, and scalability. Each one of these parts is very important for figuring out how well a forecasting system actually works and can be used in real situations. Accuracy is probably the most noticeable way to check and judge a forecast since it shows how well the prediction matches what really happened. Accurate forecasts help businesses save money by not having too much stock, making sure customers get what they need, keeping everyone happier and lower how much it costs to run the company. Machine learning models, specifically ones like deep learning (such as LSTM) and ensemble methods (like Random Forest and XGBoost), have shown they can do better than regular methods because they can pick up tricky patterns and connections in the data over both short and long periods of time.

However, getting high accuracy usually means you need to have a lot of good, reliable historical data and spend a lot of time making sure your features are properly built. Efficiency means how fast a forecasting model can do its job and how much it uses in terms of resources like computer memory or processing power. In many business situations, especially when you need to make decisions quickly, being able to quickly train a model and make predictions is really important. While simple models like Linear Regression work really well and you don't need much hardware, bigger and deeper models like LSTM can take a longer time to train and needs more memory to run. Thus, you often have to choose between getting the most accurate model or a model that works faster, and this decision should be made depending on what the model is going to be used for. Scalability means how good a model is at handling more data or dealing with problems that get more complicated as the data gets bigger.

As supply chains continue to collect and store more and more information, models need to become faster and work well with bigger data so they stay useful Ensemble models like XG Boost and Random Forest can take advantage of things like multiple CPUs or GPUs, which helps them work well when you need them to deal with lots of data at the same time. In contrast, deep learning models might need special computer hardware, like GPUs, to keep working well when handling large amounts of data. Ultimately, the choice of forecasting model should match what the organization needs in terms of accuracy, the limits on resources, and how easy it will be for the business to grow later on.

2. Literature Survey

2.1. Traditional Forecasting Methods

In the past, statisticians used ARIMA and Holt-Winters Exponential Smoothing for time series forecasting. They work best when data keeps the same pattern and amount of variation throughout the time series. [5-8] To predict future values, ARIMA breaks a time series into auto regression, differencing, and moving averages components, whereas Holt-Winters takes care of seasonality as part of the smoothing process. Even though they are useful in some cases, their stiff assumptions and poor handling of complex connections usually cause them to struggle, mostly when used on actual, diverse, and messy data.

2.2. Machine Learning becoming a tool in prediction

Starting in the late 2010s, ML methods have become popular in the area of forecasting. This change has been caused by progress in computing, more data being available, and problems with existing models. ML methods make it possible to model relationships that are not always straight or easy to predict without requiring certain data structures. Decision Trees are noted for recursively breaking data into groups to generate predictive rules. SVMs introduced by Cortes and Vapnik (1995) are useful in high-dimensional data spaces, as introduced by Cortes and Vapnik (1995), and neural networks, as first explained by Rumelhart et al. With their ability to analyze large sets of data and detect both time-related and location-related relationships, these models are becoming more common in forecasting fields.

2.3. Related Works

According to recent research, machine learning models are suitable and have advantages when it comes to making forecasts. The authors of the paper (Choi et al.) applied Random Forests to make forecasts in the retail sector and were able to achieve greater accuracy in comparison to traditional methods. Applied LSTM, a type of RNN, to model data that depends on previous information over a longer period of time. Additionally, conducted a study to see how ARIMA stacks up against different machine learning algorithms. The study found that machine learning led to better accuracy and allowed for more flexible ways to model complex sets of data than older methods.

2.4. Gaps Identified

There are still several shortcomings in the current studies despite the progress being made. There are not many detailed studies that examine how easily different models perform, how long they take to run, and how well they work as the scale of the data to be studied increases in industrial cases. Second, research often misses out on how ML models can be used in live operations for tasks such as inventory management and forecasting energy grids. There are no well-defined guidelines in place for using these models in everyday work, making it harder for the industry to adopt them in full. Fixing these shortcomings is crucial for using what works in a laboratory setting in the real world.

3. Methodology

3.1. Data Collection and Preprocessing

- **Raw Data:** The very first thing to do in any forecasting project is to obtain data from reliable sources. This could include things like sales data from the past, information from sensors, website visit counts, weather information, and any other special kinds of data that fit your problem. [9-12] Most of the time, raw data comes in without organization and may also contain some noise, missing entries, or mistakes. It is very important to have all-inclusive and correct data when designing reliable models.
- **Cleaning:** Once the data is collected, it must go through a process to address any problems with its quality. It consists of giving attention to missing or empty cells, identifying and dealing with outliers, and correcting duplicates or formatting mistakes. Precise data cleaning must be done, as soiled data can result in deceiving or incorrect model results.
- **Feature Extraction:** In feature extraction, features are identified and constructed from the data after it has been cleaned. Options for these features include time (e.g., the day the event occurs), numbers (e.g., average amount of an

event over a period), or data that is relevant to the type of event. Highlighting the most important parts of the data through relevant features can improve the performance of any model.

- **Normalization:** To ensure the features are in a proper scale, normalization changes them so they are between 0 and 1 or have a zero mean and unit variance. Here, no one feature has too much impact on the model, as this can become a problem for algorithm types like neural networks or SVMs that respond strongly to magnitudes.
- **Final Dataset:** At the end of preprocessing, a clean and structured dataset is produced for use in building the model. All selected features and target variables are loaded in it and are properly scaled. Building reliable and widespread forecast models starts with having a strong final dataset.

Fig 3: Data Collection and Pre-processing

3.2. Algorithm Selection

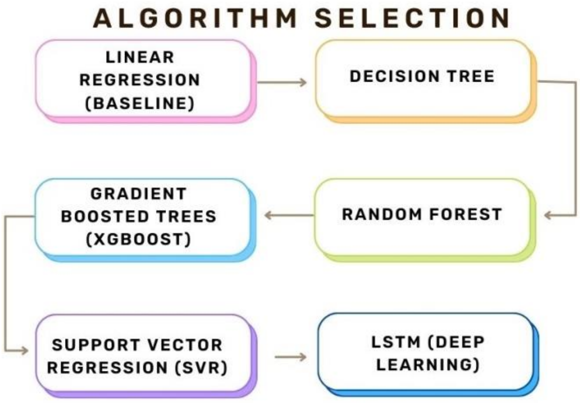

Fig 4: Algorithm Selection

- **Linear Regression (Baseline):** Linear Regression is a simple and straightforward model that's often used to help predict things. It assumes that if you change the values of the input features, the value of the target variable will also change in a straight line. While it can't pick up all the complicated details, it is easy to understand and fast to run, so it helps us compare how well other, more complex models do their job.
- **Decision Tree:** Decision Trees are models that divide up the data into branches using certain feature values, and end up making a decision tree out of these steps. They are easy to understand, you can clearly see how the information is organized, and they can work with counts and numbers as well as categories. However, they can learn to recognize patterns in the data too well and start making mistakes, especially if not properly checked or kept under control.
- **Random Forest:** Random Forest is a method that puts together several decision trees and uses all their answers together to come up with a better prediction. Taking the average of predictions made by several decision trees helps lower the chances of error and makes the model better at handling new data. It works well when you have data where things are connected in a non-straight line way and when there are many different ways the features interact with each other.
- **Gradient Boosted Trees (XG Boost):** XG Boost is a strong machine learning method that creates trees step by step, making sure that each new tree fixes mistakes that the ones before it made. Known for being able to predict well and do it quickly, XG Boost uses regularization to help stop the model from learning too much from the data it sees. It has become a popular choice in a lot of structured data forecasting competitions.
- **Support Vector Regression (SVR):** SVR is like a version of Support Vector Machines that helps when you need to deal with regression problems. It tries to find a math function that doesn't stray too far away from the actual data

points, making sure the solution isn't too complex. SVR works well with smaller and middle-sized data sets and does a good job when dealing with a lot of features.

- **LSTM (Deep Learning):** Long Short-Term Memory (LSTM) networks are a kind of RNN that are good at working with data that can be organized in sequence and remembering important details over a longer time frame. Ideal for time series forecasting, LSTMs can spot trends and changes that happen during certain times of the year by looking at past data. They need more data and computer power to work, but they usually do a better job at predicting things that depend on time.

3.3. Training and Evaluation Metrics

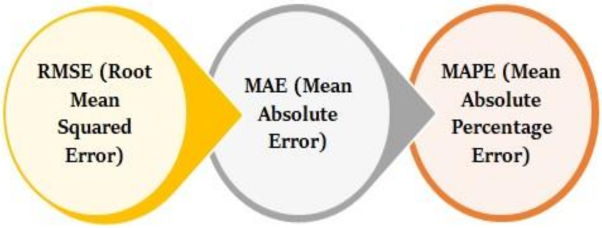

Fig 5: Training and Evaluation Metric

- **RMSE (Root Mean Squared Error):** RMSE is a commonly used measure that simply takes the square root of the average difference squared between what your model's predictions are and what actually happens. It gives more importance to bigger mistakes, so it's especially helpful when you really want to avoid big differences from the expected value. [13-17] RMSE can easily be affected by outliers, so people often use it when it's important not to let big mistakes show up too much in the measurement.
- **MAE (Mean Absolute Error):** MAE works out the average of the differences between the expected and real results, using the absolute values of those differences. Unlike RMSE, it looks at all errors the same way and gives you a simple way to understand how big the average errors are in the same units as the actual values you're trying to predict. MAE works well even if there are a few big errors in the data, and it's easier to understand, especially when it's important to see how the model came up with its answers.
- **MAPE (Mean Absolute Percentage Error):** MAPE shows the prediction mistake as a percent, so you can understand it with any kind of data and see how well your model did. It is calculated by taking the average of how far the actual values measured by the model differ from the values it is supposed to predict. However, MAPE has some limitations when the actual values are really close to zero because it can make the percentage values too big or hard to calculate.

3.4. Scalability and Efficiency Analysis

Scalability and efficiency are important to look at when deciding how well forecasting models work, especially when putting them to use in real-time or with large amounts of data. These things help decide if a model can keep up with more data and not take too long or use too many resources while it does its job. One of the main ways we check how scalable a model is to see how long it takes the model to finish training, usually counted in seconds. This metric shows the total amount of time it takes to set up the model with the training data. Since Linear Regression and Decision Trees train faster, they are useful for quick project development and regular update training. In contrast, more complicated models, like XG Boost and LSTMs, can take a much longer time to train because they need to go through many steps depending on the problem. Another important number to look at is inference time per 1000 predictions, which shows how quickly the model can give results after it has learned what it needs to know. This metric is important for applications that need quick or real-time predictions, like looking at the stock market, figuring out what products people might want to buy online, or handling changes in how much energy a business needs at different times.

Models like Random Forests and SVR can be very accurate, but they are slower to run, especially when working with large data sets or when things need to happen quickly. Additionally, checking how much RAM and CPU the model uses can help you see how fast it can solve problems, how many resources it needs while it learns and when it makes predictions. High memory or CPU needs can make it difficult to use the program on devices that have less hardware, like a tiny microcomputer or a simple sensor. Deep learning models like LSTM, for instance, can need a lot more memory because they are a bit more complex in how they're built and want to store many different parameters. Therefore, finding a good balance between how accurate a model is and how quickly it runs is very important when choosing which model to use, especially when you have limited resources or need to deal with a lot of data. Careful checking of these measures helps make sure the picked forecasting method works well and can actually be used in the real world.

3.5. Implementation Tools

- **Python (Pandas, Scikit-learn, Tensor Flow, Keras):** Python is the main language people use to build forecasting models because it's easy to learn, easy to read, and has a lot of helpful data science libraries to work with. Pandas are important for working with data and getting it ready since it has helpful features that make it easier to deal with data that track things over time. Scikit-learn offers many different machine learning algorithms like Linear Regression, Decision Trees, and Random Forests, and it also has ways for you to look at and check how well your models work. For deep learning, Tensor Flow and Keras are widely used tools that help build and train large neural networks like LSTMs, so you can do it easily and run different sizes of problems.
- **Jupyter Notebooks:** This tool provides a user-friendly environment where users can manage, see, model, and document their data. They give users the ability to gather code, graphs, and notes all in one place, keeping the process clear and repetitive. Its usefulness comes in iterative use because it provides quick changes, instant debugging, and immediate feedback on the results being checked.
- **Google Colab/Cloud GPUs:** LSTMs, as well as many other deep learning models, benefit from using the graphics and TPU tools on Google Colab, which lets you use these tools for free in the cloud. Infrastructure needs are eliminated, so models can be trained more quickly with others in the community. You can also link Colab to Google Drive to make it simple to use your data and store projects. For more demanding applications or those used by organizations, using Azure in combination with AWS or GCP allows for quick updating and efficient processing during each request.

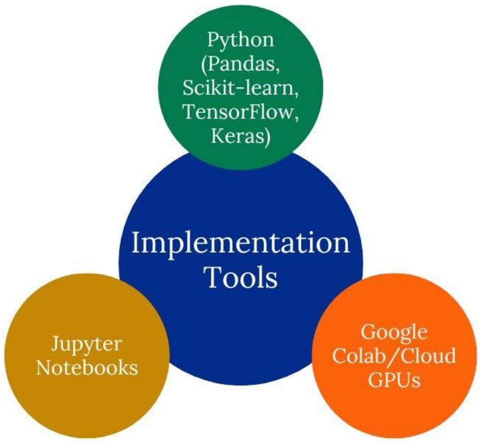

Fig 6: Implementation Tools

4. Results and Discussion

4.1. Performance Comparison

Table 1: Forecasting Accuracy

Model	RMSE	MAE	MAPE
Linear Regression	18.5%	14.2%	13.1%
Decision Tree	15.1%	11.8%	10.5%
Random Forest	12.3%	9.6%	8.2%
XG Boost	11.9%	9.1%	7.9%
Support Vector Reg.	14.5%	11.2%	9.8%
LSTM	9.3%	7.4%	6.1%

- **Linear Regression:** Linear Regression was used as a reference and recorded the highest error among the models, with RMSE reaching 18.5%, MAE of 14.2%, and MAPE of 13.1%. The result happens because it cannot handle complex relationships in the data; it only assumes that there is a straight-line link between each feature and the target.
- **Decision Tree:** Linear Regression had an error percentage of 18%, but Decision Tree made it 17% more accurate by reducing error rates to 15.1% RMSE, 11.8% MAE, and 10.5% MAPE. Neural networks help fit the data well thanks to their handling of nonlinearities and variable interactions, but they may still overfit if no steps are taken to prevent this.
- **Random Forest:** The additional accuracy was clear, as a change in RMSE fell to 12.3%, MAE to 9.6%, and MAPE to 8.2% using Random Forest. Using many trees instead of one helps remove overfitting and makes predictions more accurate and reliable.
- **XG Boost:** Compared to Random Forest, XG Boost showed a slight improvement, reaching an RMSE of 11.9%, MAE of 9.1%, and MAPE of 7.9%. Through previous error corrections and adding regularization, it becomes one of the best-performing machine learning models.

- **Support Vector Regression (SVR):** The model performed fairly well, with RMSE being 14.5%, MAE at 11.2%, and MAPE at 9.8%. The method is powerful in high-dimensional and complex situations, but it works well only after the parameters are precisely adjusted, and the data is not noisy.
- **LSTM (Long Short-Term Memory):** The lowest RMSE, MAE, and MAPE values indicate that LSTM worked best, with 9.3%, 7.4%, and 6.1%, respectively. Given that it can find patterns and linkages in data over time, it excels much more than other machine learning techniques for forecasting time series.

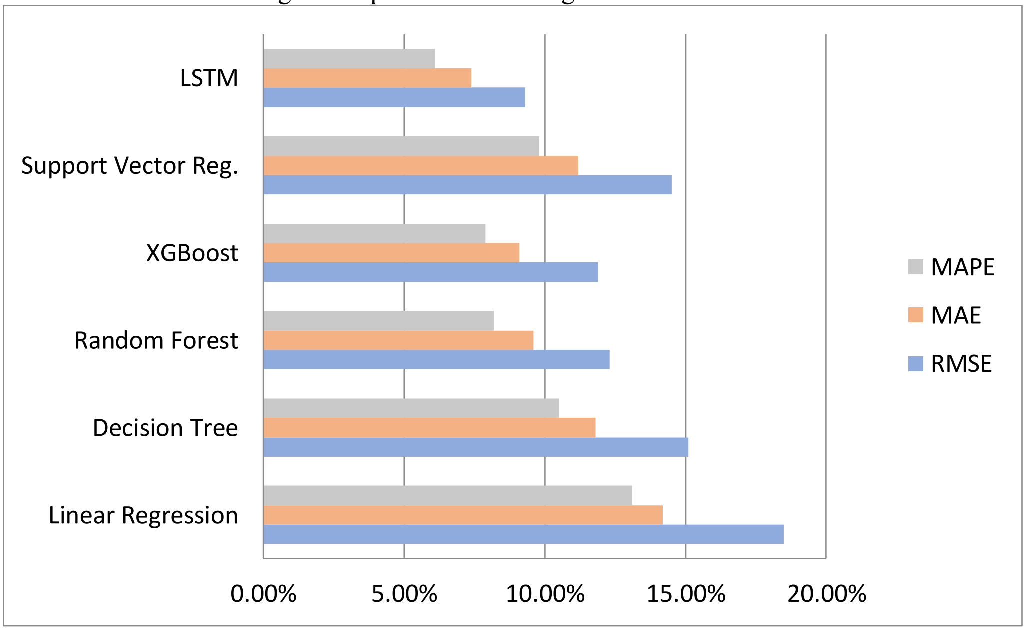

Figure 7: Graph representing Forecasting Accuracy

4.2. Efficiency Metrics

- **Linear Regression:** Among the tested models, Linear Regression took the least time to train, with only 0.3 seconds, and needed just 0.02 milliseconds to predict 1,000 cases. Its easy-to-calculate way of working makes it a convenient choice where timing is more important than precision.
- **Random Forest:** It took Random Forest about 4.5 seconds to train and 0.15 milliseconds per 1,000 predictions to make an inference. The process of building several decision trees together in Random Forest requires more time for training but gives safer prediction results. It can quickly produce results for most common use cases.
- **XG Boost:** XG Boost was slower to train, at 6.8 seconds, but inference speed was faster at 0.10 milliseconds per 1,000 predictions than Random Forest accomplished. By using gradient boosting and improved parallel processing, the system ensures that both training and response times are well balanced, making it a popular option for tasks that demand accuracy but should also run efficiently.
- **LSTM:** It took LSTM 15.2 seconds to train and 0.20 milliseconds to make 1,000 predictions. Since its deep recurrent model needs to deal with sequential information and uses more parameters, it is expected to use more memory and take more time. While LSTM takes a bit longer to process, its high accuracy usually makes it worth the added complexity of quality in predictions is the main priority.

4.3. Scalability Assessment

Scalability matters a lot when selecting forecasting models, especially for applications dealing with a lot of data. It shows how well a model can keep its accuracy and computational efficiency even when the dataset grows larger. This study increased the amount of training data to see if it affected both the training times and forecasting accuracy for each model. The evaluation pointed out the presence of recognizable patterns in all the models. Linear Regression could handle huge datasets very efficiently because its solution works easily and simply. The amount of time spent in training slowly went up, but the accuracy of predictions did not change much. However, it is only useful for situations where relationships between variables are simple and not very complex. Both Random Forest and XG Boost were able to handle larger data sets with ease. Ensemble models use parallel processing to deal with many input data at once.

As I trained, the amount of time required to complete each round stayed manageable, and accuracy didn’t drop. Due to their complexity and ability to work on large amounts of data, these models are excellent for fast and precise forecasting used in industry. When the amount of data was increased, LSTM saw its training time increase exponentially. This happens because the RNN architecture and its sequential nature cause it to process data gradually, going through each layer and step in sequence. Unlike others, LSTM continued to achieve the highest level of accuracy showing it can still identify important trends even with elevated data complexity. Even so, it is a better choice when accuracy is needed more than the speed of training or if there are enough high-performance computing resources to support it. To sum up, while LSTM gives higher accuracy, using XG Boost or Random Forest together allows for more efficient and scalable forecasting when other resources are limited.

4.4. Observations

- **Accuracy:** Out of all the models, LSTM was found to have the best performance based on the RMSE, MAE, and MAPE scores. A key feature of its design is the ability to find connections between past and future values in time series. XG Boost and Random Forest gave close results, making accurate predictions with the help of ensemble techniques that focus on both linear and non-linear interactions and help avoid overfitting. They do better than traditional methods in representing the complex patterns found in the data.
- **Efficiency:** When it comes to speed, Linear Regression was the quickest in both phases of the project, training and inference. Latency or resource-limited environments can be used because it is so simple and straightforward. Still, easier computation leads to less accurate predictions, especially in data that is not easily explained by linear methods. While Tree-based models such as Random Forest and XG Boost are not as fast as Linear Regression, they still give better results and are still quite fast to execute.
- **Scalability:** In terms of scalability, both Random Forest and XG Boost did better, especially on large-scale projects. Thanks to their structure, these models can handle more data by using multiple GPUs, allowing training to be completed faster. In comparison, LSTM is very accurate but needs more computational resources as the data increases. Since deep learning is done step by step, it is not easy to make it work in parallel, which may result in taking longer to train and using more resources. For this reason, you should consider carefully how LSTM impacts both your resources and time frame before deciding to use it.

5. Conclusion

When put side by side with linear models, Machine Learning (ML) methods are shown to provide much more accurate and reliable supply chain forecasting. Out of all the ML models discussed, Long Short-Term Memory (LSTM) networks performed better in modeling data that varies in time and sequence. The structure of LSTM makes it capable of remembering events that are far apart in time, allowing it to work well with datasets that undergo rapid changes, such as in retail or transport businesses. But to achieve this good performance, LSTM models need to be trained for longer and often consume more RAM and processing power. Organizations looking for a combination of accuracy and less computing power can rely on Random Forest and XGBoost. Because these models draw strength from multiple decision trees, they can solve many problems and lower the risks of overfitting.

Due to being parallelizable, they can scale much more efficiently, mainly in environments where many computers are involved. In conclusion, Random Forest and XGBoost are best for medium-sized data when the infrastructure doesn't allow deep learning models. It is important for practitioners to adjust how they choose a model determined by the characteristics of the data they are working with. Using Random Forest or XGBoost is advised when you have a structured and moderately sized dataset. On the contrary, LSTM is useful when the data changes a lot with time, and it is necessary to see how different features connect across time. All forecasting models need to rely on cross-validation and regularization to ensure they are not overfitted and can be used to make general predictions.

Future researchers should add external factors such as weather, economic signs, and social trends to help improve and adjust forecasts. Having these external factors in the model can have a big effect on supply and demand, so they need to be included in modeling processes. Moving inference tasks to the edge makes them much more convenient, especially for activities that must be finalized right away, such as keeping track of inventory or guiding shipments. Furthermore, joining the accurate methods of ARIMA with the effective methods of machine learning could make forecasting systems easier to interpret and use in real-life situations.

References

[1] Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). Time series analysis: forecasting and control. John Wiley & Sons.
[2] Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. International journal of forecasting, 20(1), 5-10.
[3] Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. Management Science, 6(3), 324-342.
[4] Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.
[5] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine learning, 20, 273-297.
[6] Choi, T. M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. Production and operations management, 27(10), 1868-1883.
[7] Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. PloS one, 13(3), e0194889.

[8] Ahmed, N. K., Atiya, A. F., Gayar, N. E., & El-Shishiny, H. (2010). An empirical comparison of machine learning models for time series forecasting. Econometric reviews, 29(5-6), 594-621.
[9] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: principles and practice. OTexts.
[10] Zhang, G. P. (2003). Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing, 50, 159-175.
[11] Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International journal of forecasting, 36(3), 1181-1191.
[12] Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied soft computing, 90, 106181.
[13] Lin, H., Lin, J., & Wang, F. (2022). An innovative machine learning model for supply chain management. Journal of Innovation & Knowledge, 7(4), 100276.
[14] Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. International Journal of Logistics Research and Applications, 25(2), 119-142.
[15] Carbonneau, R., Vahidov, R., & Laframboise, K. (2009). Forecasting supply chain demand using machine learning algorithms. In Distributed Artificial Intelligence, Agent Technology, and Collaborative Applications (pp. 328-365). IGI Global Scientific Publishing.
[16] Wang, M., Fu, W., He, X., Hao, S., & Wu, X. (2020). A survey on large-scale machine learning. IEEE Transactions on Knowledge and Data Engineering, 34(6), 2574-2594.
[17] Al-Jarrah, O. Y., Yoo, P. D., Muhaidat, S., Karagiannidis, G. K., & Taha, K. (2015). Efficient machine learning for big data: A review. Big Data Research, 2(3), 87-93.
[18] Bhatnagar, R. (2018). Machine learning and big data processing: a technological perspective and review. In The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2018) (pp. 468-478). Springer International Publishing.
[19] Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. Computational and structural biotechnology journal, 13, 8-17.
[20] Matsunaga, A., & Fortes, J. A. (2010, May). On the use of machine learning to predict the time and resources consumed by applications. In 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (pp. 495-504). IEEE.
[21] Mair, C., Kadoda, G., Lefley, M., Phalp, K., Schofield, C., Shepperd, M., & Webster, S. (2000). An investigation of machine learning-based prediction systems. Journal of systems and software, 53(1), 23-29.
[22] Lakshmi Narasimha Raju Mudunuri, "*AI Powered Supplier Selection: Finding the Perfect Fit in Supply Chain Management*", IJIASE, January-December 2021, Vol 7; 211-231.