AI-Enhanced Digital Twins for Optimizing Virtual Business Operations in Supply Chains

Yugandhar Shilawane
Symbiosis Skills and Professional University Pune

***Abstract*—The results extend the knowledge of how AI-aided demand forecasting and digital twin concepts can be used to generate insights into inventory management and supply chain decisions. In this case, through the comparison of the specific characteristics of the implemented models, the research focuses on the advantage of XGBoost in terms of model complexity, computation time, and prediction performance.**
Some of the limitations cone with this study involve issues to do with data limitation and also the issue of data destiny in some models that take a long to produce results. To effectively apply AI for predicting building failure, there is a need for the following recommendations in the subsequent research; The quality of data must be improved since better data provide better results. These findings correspond with the purpose of this study and offer practical recommendations for supply chain practitioners regarding demand planning enhancement.

***Index Terms*—Demand Forecasting, Supply Chain Optimization., AI-Enhanced Digital Twin, Machine Learning Models, XGBoost, Feature Engineering, Inventory Management Time, Series Analysis, Seasonality and Trends, Mean Absolute Error (MAE)**

List of Abbreviation

- AI – Artificial Intelligence
- MAE – Mean Absolute Error
- LSTM – Long Short-Term Memory
- TCN – Temporal Convolutional Network
- N-BEATS – Neural Basis Expansion Analysis for Time Series
- EDA – Exploratory Data Analysis
- ACF – Autocorrelation Function
- PACF – Partial Autocorrelation Function
- XGBoost – Extreme Gradient Boosting
- RF – Random Forest
- RMSE – Root Mean Squared Error
- KDE – Kernel Density Estimation
- RNN – Recurrent Neural Network
- SCM – Supply Chain Management
- API – Application Programming Interface
- MSE – Mean Squared Error
- CPU – Central Processing Unit
- GPU – Graphics Processing Unit

I. INTRODUCTION

1.1 Introduction

Global supply chain dynamics have been changing at an alarming rate, and it has introduced many open issues to the whole supply chain system such as increasing intricacy, variability, and vulnerability. Solving these problems is possible only by applying complex tools that can increase the work's productivity and stability. Digital twins (DTs), models of real-life supply systems, have become valuable assets in enhancing the management of logistic networks through real-time information and analysis. However, they can only be so effective before they are complemented with AI or artificial intelligence. This work explores the evolution of Digital Twins with AI integration with real-life use incorporating enhanced supply chain demand forecasting and inventory control for demand-driven operations.

1.2 Background

Modern-day business being the global village entails relying on supply chains to facilitate supply and distribution of products and services across the entire globe. However, growth in the intricacy of supply chain networks and every buyer's rising demands and adjusting market nature have posed some operating complexities. Disruptive events like the COVID-19 pandemic revealed gaps and issues in conventional SC

planning, demand signal forecasting, inventory and distribution, and system robustness (Bhandal *et al.* 2022). It occurs from such disruption that the call for reflecting on how supply chain management may enhance response to change, enterprise flexibility, and effective decision-making arises.

Digital Twins (DTs) have been raised as the potential technological approach that provides an accurate digital replica of physical assets and systems.

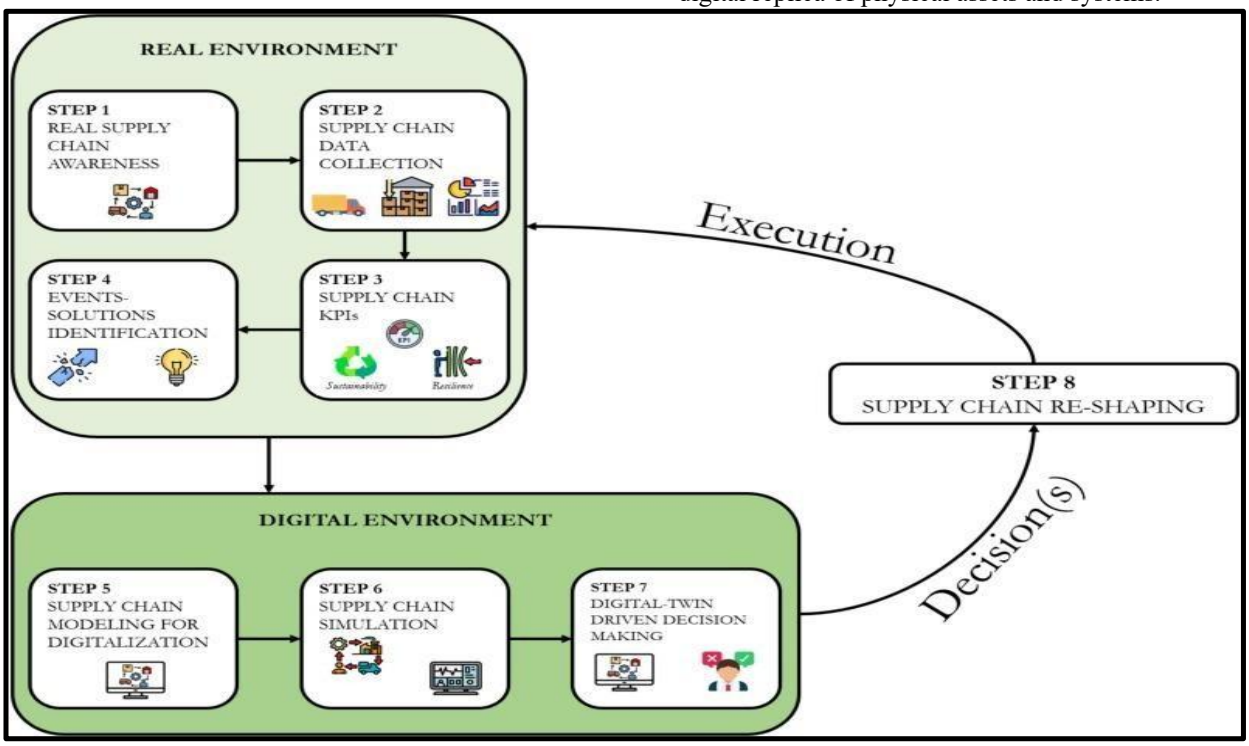

Figure 1.2.1: Supply Chain Digital Twin concept towards a more resilient and sustainable future (Source: Cimino *et al.* 2024)

Being embedded in the supply chain, they provide end-to-end solutions to manage and prevent disruptions and improve supply chain efficiency. However, these instances have demonstrated the effectiveness of using DT to computationally assist with data analysis, while most existing applications of DT are extensions based on monitoring static data sources and do not have robust real-time dynamic capabilities.

The incorporation of AI into DT concepts offers the chance to improve the frameworks' capabilities for prediction and optimization. Through the use of machine learning and the constant stream of new data coming from IoT devices, the effectiveness of demand forecasting can be increased, inventory turnover can be maximized, and costs can be minimized (Helo and Hao, 2022).

1.3 Rationale

What is the issue?

Customers put more pressure on the supply chains to handle volatile demand, improper inventory management, and disruptions from events. Traditional forecasting methods are weak in capturing dynamic demand patterns which produce overstocking, stockout conditions, and high operating costs.

Why is it an issue?

Such inefficiencies cause financial losses to the relevant organizations, low satisfaction of customers, and lesser competitiveness. Stiff and slow supply chain networks are easily affected by instabilities because they cannot easily adjust to changing market conditions, especially when demand fluctuates as is the case for retail and manufacturing companies.

Why is it an issue now?

The current COVID-19 pandemic and other global disruptions have therefore highlighted some of the biggest vulnerabilities in conventional supply chain

strategies. Since industries are trying to post-COVID19 rebrand and improve operations, the demand for new technologies like AIDTs is paramount in supply chain optimization (Argyropoulou *et al.* 2024). The global availability of IoT data and the progress of AI provide a timely chance to address these issues adequately.

1.4 Problem Statement

Supply chain technologies remain the strategic priority even though the industry is aware of main supply chain challenges related to demand forecasting and inventory management, specifically, in situations with high variation and disruption (Ivanov, 2024). In its current form, Traditional DT is suited more for static, performance monitoring, and visualization applications and does not include the dynamic, real-time change prediction necessary for timely and effective action. This constraint hinders supply chain processes from reaching their full efficiency, which results in problems including demand forecasting problems, inventory system problems, and expensive and dissatisfied consumers.

The overall formula is enhanced by integration of AI into DT frameworks, which offers decent, yet quite demanding, solution of the problem. These are data integration problems, scalability difficulties and computational load arising from processing actual IoT data feed. Second, where there is no well-defined model of AI-OR implementation in DT, the applicability and adaptability of DT in diverse sectors is also restrained.

1.5 Research Aim

The purpose of this dissertation is therefore to suggest a framework of how AI can be used to integrate DT with demand forecasting and inventory distribution in supply chains. The research focuses on determining nine areas of technology and functions where the reduction of these values would go a long way in enhancing supply chain improvement, reduction of costs and increased responsiveness to change. Moreover, the research evaluates the model to determine the general applicability of the model in regard to application to organizations in various industries with diverse operations.

1.6 Research Objectives

1. To develop a scalable AI-enhanced Digital Twin model that will be used to enhance the accuracy of predictive demand forecasting within supply chain operations. The adaptability and scalability of this model shall, therefore, constitute its core to enable application within diverse contexts of supply chains, especially those with highly variable demands like retail and manufacturing industries.
2. Emphasize the identification and mitigation of critical technological-operational challenges related to deploying AI-enhanced Digital Twins in predictive demand forecasting and inventory management. This work will discuss possible manifestations of adoption barriers in data integration, scalability, and computational issues, and further present the best practices with recommendations that will ensure successful integration.
3. Assessing scalability and adaptability of AI-enhanced Digital Twins across various supply chain contexts, it evaluates what influences successful implementation in different industries. This objective will let one see with what degree of performance the model performed in various sectors and what level of customization it takes to fit it for retail, manufacturing, or logistic environments.

1.7 Research Questions

1. How would AI-driven digital twins drive better demand forecast accuracy and inventory management efficiency in the supply chain, while at the same time positively impacting cost reductions and improvements in resiliency?
2. What might be some of the most crucial issues of technology and operation regarding AIimpelled Digital Twins used for predictive demand forecasting, and how would businesses overcome those challenges to ensure complete efficiency in their supply chains?
3. How scalable and adaptable are AI-enhanced Digital Twins for a wide variety of supply chain environments, and what is the most determining factor in their successful implementation across various industries?

1.8 Research significances

This is where the importance of this research is rooted as it aims to tackle key issues contemporary supply chains struggle with, especially in high-demand volatility and disreputability industries. Conventional techniques of demand forecasting and inventory

control do not suit changing market conditions and business processes leading to wastage of resources, higher costs, and an unsatisfied customer base.

This research aims to extend the limitations of DT frameworks by incorporating Artificial Intelligence (AI) (Riad *et al.* 2024). AI integrating in DTs can enhance the use of machine learning algorithms, real-time IoT data, and provide high accuracy in forecasts, increase turnover of inventories, and optimize operations.

That is why the study is especially valuable since it concentrates on the aspects of scalability and flexibility The best practices described can be easily implemented in elemental supply chains belonging to the retail, manufacturing, and logistic sectors. Furthermore, it addresses the main bottlenecks of the implementation, such as data concatenation, computational issues, and practical application.

1.9 Research hypothesis

H_0 (Null Hypothesis):

AI-enhanced Digital Twins fail to enhance the forecast demand accuracy or inventory management and operational reliability of the supply chain and supply industries. It is envisaged that incorporating AI into Digital Twin frameworks will encounter challenges ranging from data integration issues, scalability issues, and high degrees of computation to exerting marginal positive effects on the supply chain. *H_1 (Alternative Hypothesis):*

Supply chain demand forecasting, inventory management, and general operations are particularly benefitted by AI-driven DTs. These systems use real-time IoT data and transcend the capabilities of other methods to offer concepts that are scalable and flexible with solutions that can support KPIs and contain systemic methods to decrease and respond to costs and material variability.

1.10 Research structure

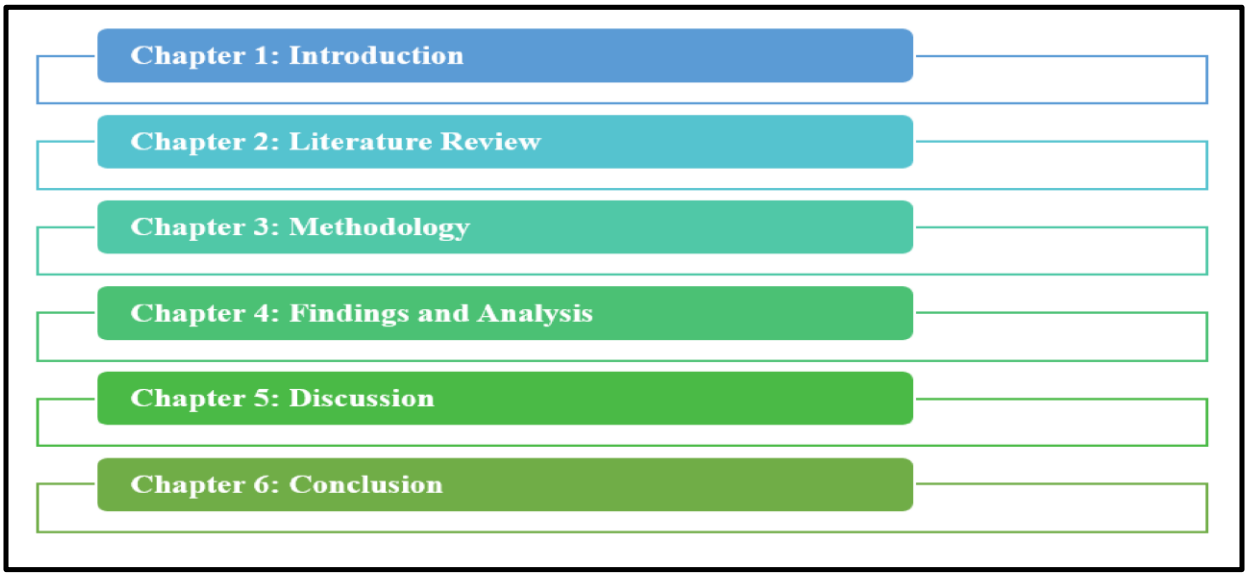

Figure 1.10.1: Structure of the Research

(Source: Self-created)

1.11 Summary

The Introduction chapter introduces the research topic of the research study by highlighting current difficulties in complex supply chains and using AI-based Digital Twins to overcome them. First, it defines the research context and motivation, pointing out the inefficiency of conventional forecast and stock approaches in change and variability conditions (Biller and Biller, 2023). By analysing the gaps in the existing applications of the Digital Twin, which include absence of the action capability and adaptability, the problem statement highlights the centrality of AI for supply chain enhancement.

Similar to the previous chapters, the chapter defines the research objectives and questions, which are to design, implement, and test the scalable AI-based Digital Twin model, identify engineering and operational issues in adopting Digital Twin solutions, and assess their generalizability across different industries.

II. LITERATURE REVIEW

2.1 Introduction

The chapter introduces the reader to the current state of the literature regarding AI-supported Digital Twins (DTs) in supply chain contexts. Purposedly, this chapter reviews selected literature from academic and industrial reports to ground the objectives of the study and reveal research gaps that the dissertation aims to fill. It examines the development history of Digital Twin tech and its connexion with artificial intelligence (AI), how both disciplines may boost predictive demand and stock control.

Digital Twin can be described as the innovative approach in Industry 4.0, which provides organizations with the enhanced opportunity to build an actual and constantly updated virtual representation of the concerned object. These virtual models are beneficial in emulating, controlling and steering supply chain activities as have been documented by other scholars (Semeraro *et al.* 2021). AI complements Digital Twins by enhancing their functions, functions like data analysis, modelling and decision making. A number of studies have shown the manner in which Digital Twins increases efficiency in supply chain management by increasing demand forecasts, decreasing cost, and managing risks that result from volatile market conditions.

2.2 The Role of Digital Twins in Supply Chain Management: Opportunities and Current Limitations

Digital Twins (DTs) have become an innovative tool in the context of supplying chain management (SCM) and hold a huge potential for facing the challenges of new SCM environments. Digital Twins use real-time, data generated through constructive virtual representations of the physical structures with supply chain processes to monitor, simulate, and optimize the methods. Since world supply chains are getting more complex and unpredictable, DTs are being viewed as a valuable asset to turn a business, and its operations, higher and better, in terms of its operations, demand prediction, and inventory control.

The real benefit of Digital Twins is the possibility of aligning the actual activities with digital entities as discussed below. This coordination helps in collation of data in real time, analysis and prediction of business trends and patterns thus allowing efficient business decisions to be made (Preil and Krapp, 2022). For instance, DTs facilitate an understanding of the supply chain structures and processes, dynamics of disruptions, and different scenarios and their implications for performance. Using of the DTs in the context of predictive demand forecasting contributes to detailed understanding of market behaviour to minimize potential forecasting errors and excess inventory. The literature identifies such capabilities as necessary for industries with high variability of demand, for instance, retailing and manufacturing.

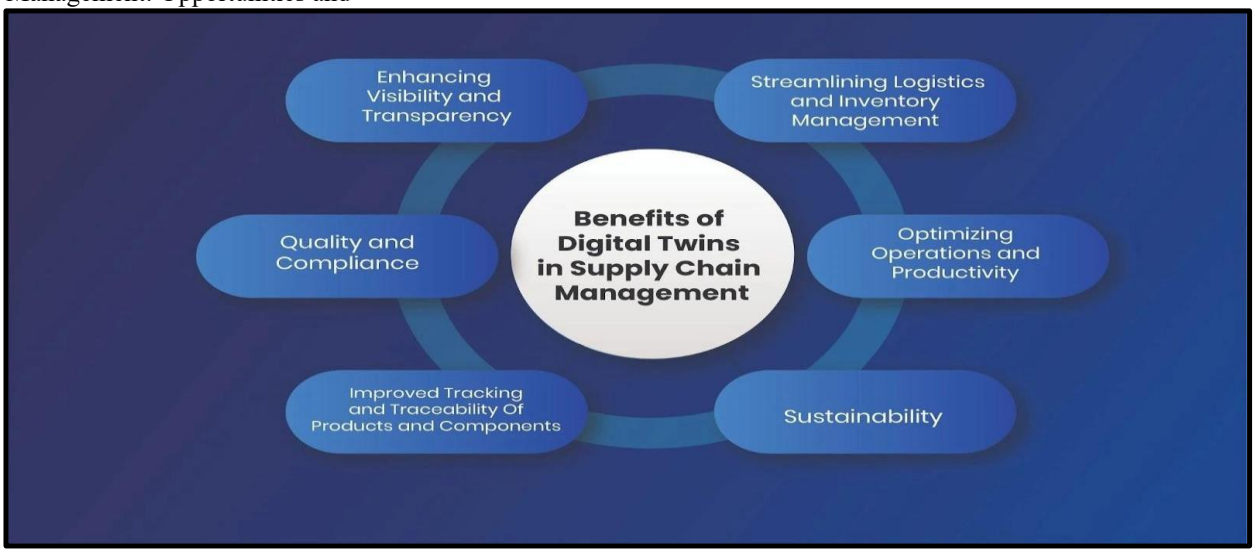

Figure 2.2.1: Digital Twins in Supply Chain Management

(Source: toobler.com, 2024)

Moreover, it is important to notice that supply Digital Twins can also strongly improve supply chain robustness. Due to connection with various platforms like IoT devices, blockchain, and ERP system, DTs enables supply chain visibility from end to end. This requires having the networks and resources visible, so that vulnerabilities can be seen and addressed, and the risks mitigated, and where there are interruptions, such as during the COVID-19 outbreak, the network enjoys the flexibility needed. As analysed in the current studies, DTs can simulate changes in the supply chain of demand variations, supplier downtimes, and transportation difficulties, all of which will allow companies to forge continuity.

2.3 AI Integration in Digital Twin Frameworks: Advancing Predictive Analytics and Operational Efficiency

AI utilization inside DT structures for supply chain is profoundly transforming by improving its prescient investigation and operational effectiveness. Integration and application of AI technology exhibit DTs offer the organizations an opportunity to transition from addressing issues as and when they emerge to solving them as they emerge and at the same time, using data to support their decision making processes (Kafy *et al.* 2021). Fragmented and intricate supply chains make the integration of Artificial Intelligence with the support of Digital Twin technology more and more indispensable for analysing data and creating effective prognosis and supply chain solutions.

New to DTs proposed by the integration of AI mechanisms is the enhancement in predictive analytics. Alerts in terms of expected future requirements and possible disruptions of supply chains have been forecasted with deep learning, support vector machines (SVM), and reinforcement learning algorithms (Al-Ali *et al.* 2020).

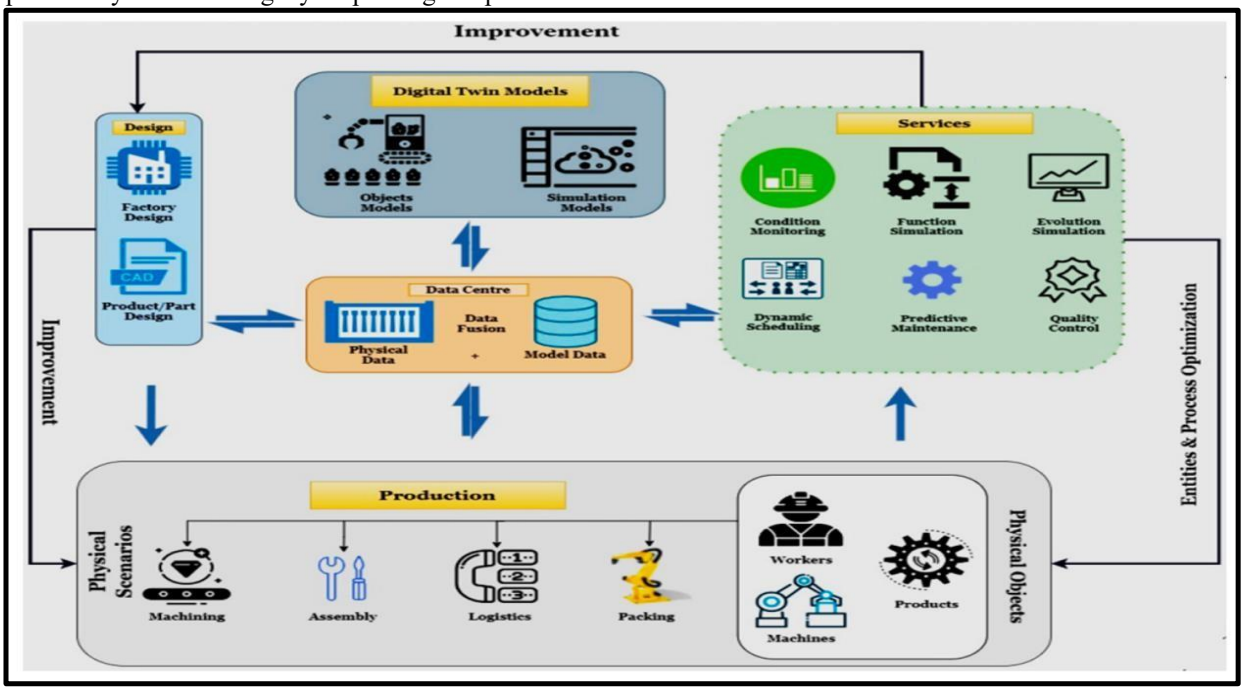

Figure 2.3.1: AI Integration in Digital Twin Frameworks

(Source: mdpi.com, 2024)

Additionally, AI integration helps to optimise inventory management in DTs. Using real-time data on inventory status, orders and supply chain, AI integrated Digital Twins can constantly update operational plan and make intelligent decision about resource utilization to minimize wastage. Computerized decision aids in conjunction with DTs are capable of dynamically changing different aspects of supply chain operations in response to demand variations, production capabilities, and transport limitations (Yaqoob *et al.* 2020). This leads to a generally flexible supply chain network which can

easily change its operations as the market demands. Research has also demonstrated that incorporating AI models to DTs can increase its reliability by eradicating some of the supply chain inefficiencies somewhere like bullwhip effect where a slight change of customers' demand significantly alters the supply chain rate.

However, these development factors make AI integration in Digital Twin frameworks effective despite the existence of challenges (Huang *et al.* 2021). One major disadvantage is the acquisition of high-quality real-time data on which the AI algorithms would run on. Incomplete data can lead to steps being taken when those steps should not be taken or not taking steps when there are likely to be beneficial. Secondly, AI models need a lot of computations and the computational possibility may not be available to all business organizations, especially those that can be categorized as small or those that operate on a low budget.

2.4 Challenges in the Deployment of AI-Enhanced Digital Twins: Data Integration,
Scalability, and Computational Demands

The use of robotic DTs in facilitating supply chain management comes with several issues especially with relation to data convergence, expansion, and processing intricacies. Such barriers should be discussed to make efforts in adopting and sustaining effective AI-powered DT systems in various contexts of the supply chain. This is a major hurdle that organizations must overcome: including various source of data, the necessity for developing systems suited to organization of any size, and the massive computational power necessary for real time processing.

1. Data Integration Challenges

Another risk, when adopting AI-driven DTs, is data integration – an organization needs to combine multiple sources and types of information into one system. Supply chain create huge data throughout various processes such as sourcing and procurement, manufacturing, order fulfilment, logistics, and feedback from customers (Abolghasemi *et al.* 2020). Unfortunately, such data sources are not coherent, instead, they stem from various systems. Importing such data into a central Digital Twin model is a challenging task since there is tanned to maintain integrity, quality, and real-time consistency across multiple sources of data.

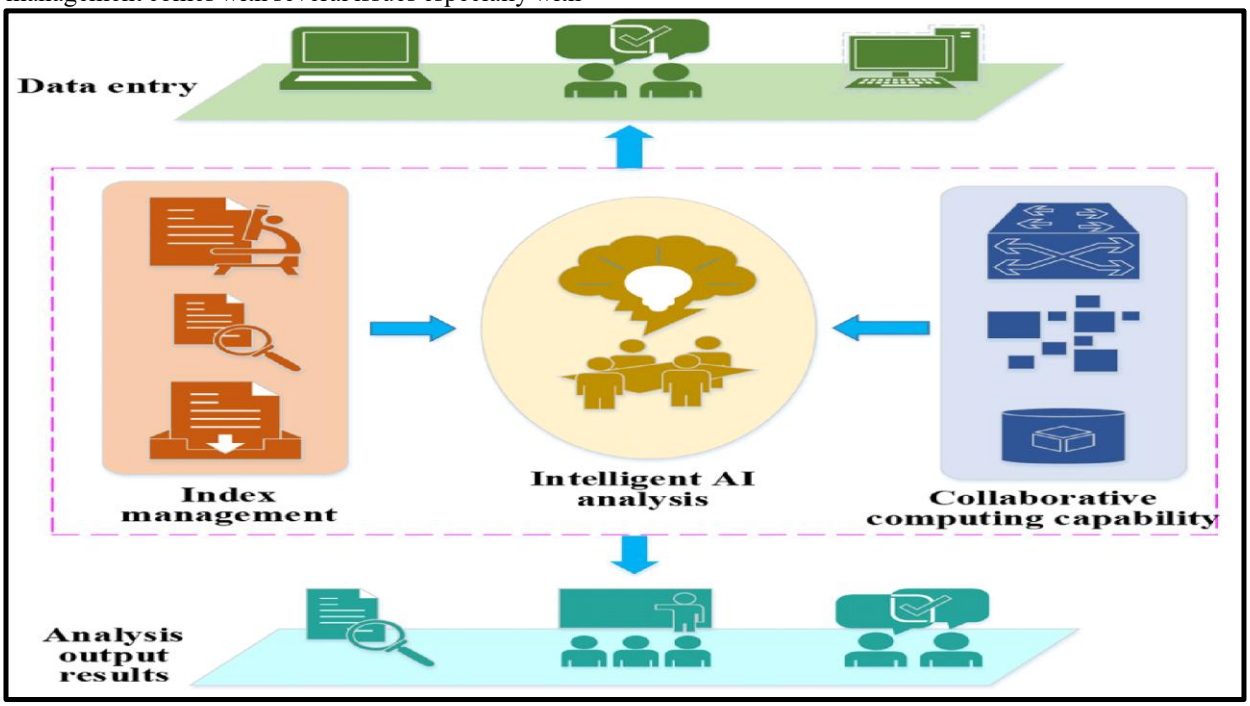

Figure 2.4.1: Challenges in the Deployment of AI-Enhanced Digital Twins
(Source: digitaltwin1.org, 2024)

AI-driven Digital Twins are based on real-high quality data to generate effective models and predictions. Even though correct details are a key in making accurate forecasts, when spaces, mistakes or disparities appear in the data, incorrect final estimations are produced, affecting the endeavours to make correct decisions, or whatsoever, which finally influences organisational operations.

2. Scalability Issues

Another key factor which is rather a challenge for the employment of AI-based Digital Twins for the PE industry is scalability. But advanced ICTs and intelligent Digital Twins demand complex models that offer better simulation and scale for handling big data. Due to the fact that supply chain may differ in form, scale and nature of the process, it is paramount for Digital Twin to be integrated to suit the required and specific business environment, starting from the multinational companies to local SMEs. The term scalability also encompasses the provision for handling ever-growing volumes of information while also providing flexibility depending on the organization's environment.

3. Computational Demands

There is a challenge to the implementation of AI-based DTs due to their large computation requirement. Big data, detailed computations, practical simulations, real-time results, and more things need the use of hefty computer systems for AI models (Ivanov and Dolgui, 2021). For instance, neural networks, which are used predominantly in developing predictive solutions, are often computationally expensive to use on large datasets for creating accurate predictions. Since these computational demands may be quite significant, so is the demand for high end computing resources especially cloud, may prove to be expensive and challenging. For such technologies, organisations, especially small firms, may not possess the requisite structures to support them.

2.5 Evaluating Scalability and Adaptability of AI-Enhanced Digital Twins Across Industries

This paper identifies flexibility and extensiveness of the Digital Twins enhanced by artificial intelligence as main primary factors influencing their success in different industries. Digital twin fundamentals stay constant, but requirements and uses in various industries drive the need for a diverse set of solutions.

1. Scalability in Manufacturing

It can be stated that Manufacturing industries are one of the first industries adapted to Digital Twin technologies because of using them in the optimization of production lines, predictive maintenance, and managing resources. AI-augmented Digital Twins within the manufacturing industry are typically used to predict models and optimize production workflows in real-time (Winter and Chico, 2023). Nevertheless, their application scale is reliant on the complexity of the manufacturing setting. Complex production lines as seen in automotive manufacturing and electronics are central to industries with elaborate processes wherein high-fidelity models with the capability of handling large volumes of data from sensor-equipped machines, robots, and operators are needed.

2. Adaptability in Retail Supply Chains

Retail is another field must suitable for AI improved Digital Twins especially to handle uncertain demand, inventory, and supply chain logistics (Sheraz *et al.* 2024). Most of the time, retail environments have a situation where demand is very volatile due to factors such as seasonal, or promotion or changing trends in consumer habits. These variations make the predictive capability and business effectiveness all the more important.

3. Logistics and Supply Chain Resilience

Derivative from the above discussion logistics that in most cases entail the transportation of goods and services from one geographical point to another encounter's major challenges as far as scalability and flexibility are concerned. AI integrated Digital Twins can involve end-to-end supply chain and help in decision making for inventory, logistics, and inventory prediction (Nimmagadda, 2021). Flexibility is very important in logistics in terms of the transportation of goods depending on the amount of cargo handled during different time periods or the lack of it as seen in the current global pandemic times.

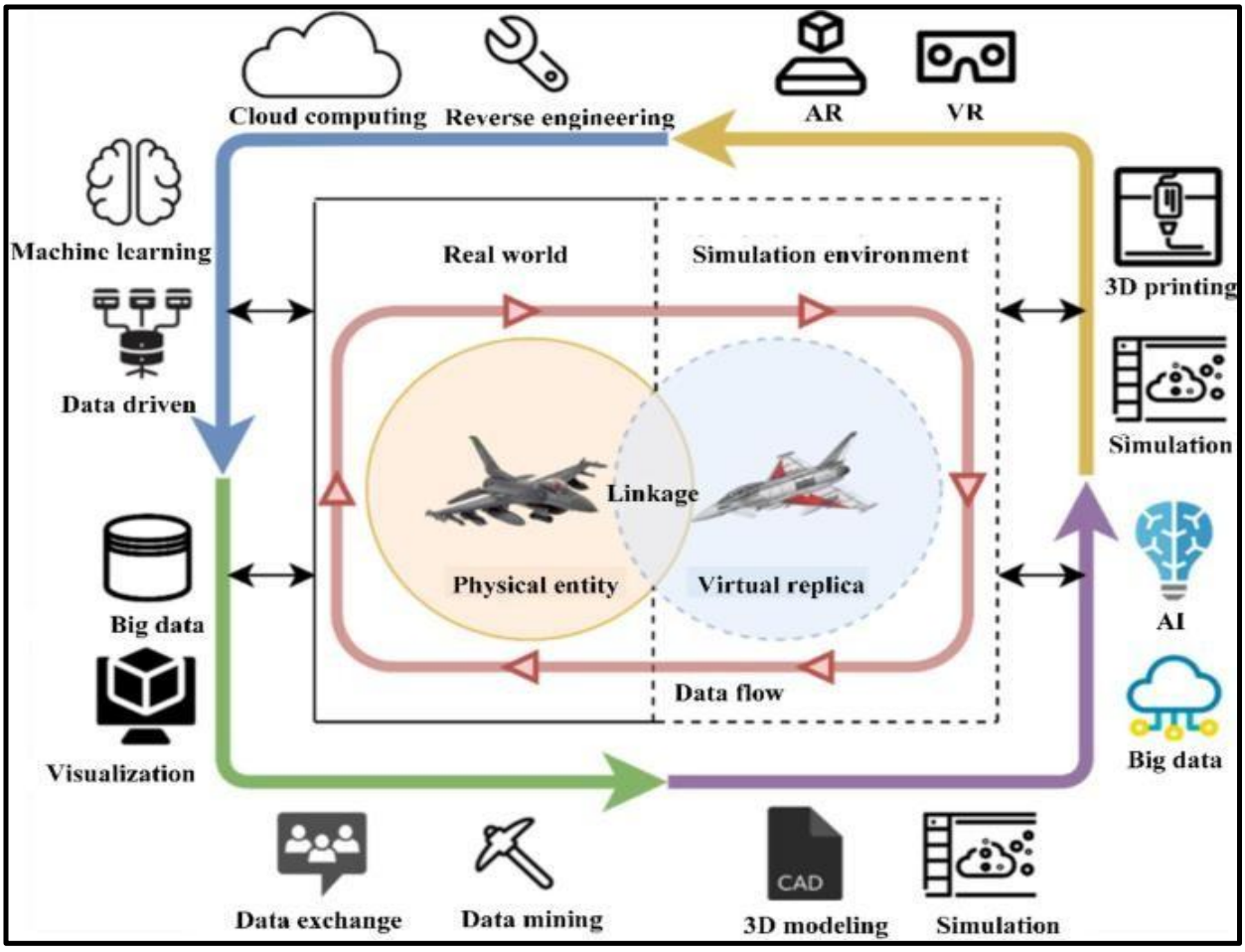

Figure 2.5.1: Evaluating Scalability and Adaptability of AI-Enhanced Digital Twins
(Source: vciba.springeropen.com, 2024)

Flexibility is another requirement as it implies that the players involved in the management of the logistic companies are plotting their moves in uncertainty and characterized by factors such as changes in the supply chain due to aspects like disasters, strikes, or higher demand.

4. Cross-Industry Adaptability

Although scalability and adaptability are assessed in a dissimilar manner across industries there are guidelines for the use of the AI-assisted Digital Twins. In this regard, many AL models have diverse structures, technologies, and operational features that should be compatible with every line of industrial business (Nguyen *et al.* 2022). Important factors include compatibility of different data sources, real-time inputs, and configurability to work along structural dimensions of vertical extent (depth) and horizontal extent (breadth).

2.6 Theoretical Framework

This theoretical review section of the dissertation on AI-enhanced Digital Twins in supply chain management adopts a number of existing theories derived from information technology, supply chain management, artificial intelligence, and systems theory (Pal, 2023). To some extent, this section seeks to align and apply these theories in am HRD frame providing outset to practicing a systemic type of thinking when identifying the usefulness, prospects and limitations of Digital Twins (DTs) in predictive demand forecasting and inventory management.

1. Systems Theory and Cyber-Physical Systems

To better comprehend Digital Twins, it is crucial to go back to the roots of systems theory, more precisely cyber-physical systems (CPS). A Digital Twin can be described as a mirror image of an existing object/structure that is capable of real-time monitoring, analysis, and simulation. The coupling of digital and physical elements within a supply chain is at the heart of CPS, computers, machines, and processes are all linked to control the physical environment.

2. Theory of Predictive Analytics and Forecasting

Demand forecasts are critical components of the AI-augmented Digital Twin of Supply Chain, where predictive analytics has enormous importance. In the theory of predictive analytics, ML algorithms assume the past behaviour is a good predictor of future behaviour (Tirkolaee *et al.* 2021). These insights help AI algorithms, which are built into Digital Twins, to forecast shifts in demand, manage inventory, and fine-tune the supply chain.

3. Resource-Based View (RBV)

The application of the Resource-Based View theory is critical to appreciate the strategic value that AI-enhanced Digital Twins may afford to supply chains. As for the RBV framework, companies achieve a competitive advantage by acquiring, developing, and deploying exclusive and valued resources such as Digital Twins in supply chain management integrated with the assistance of AI (Qian *et al.* 2022). The opportunity to receive accurate information in realtime, having access to the robust staking analytical tool, and being ready for alterations in demand patterns prove that AI-powered OT Is Digital Twins are beneficial as they can lead to cost optimization in the provided scale.

4. Technology Acceptance Model (TAM)

The TAM has been effective in identifying this critical aspect regarding the extent to which the implementation of AI-enhanced Digital Twins enhances the functioning of the supply chain. As proposed by TAM, perceived ease of use and perceived usefulness serve as the main factors influencing the adoption of a particular technology. The use of Digital Twins, especially those driven by Artificial Intelligence, is expected to bring much value in terms of enhanced decision support, prognosis accuracy as well as increasing organizational productivity (Patnaik *et al.*
2024).

5. Diffusion of Innovations Theory

Another theory relevant to this research is the theory of Diffusion of Innovation (DOI) developed by Everett Rogers. It fits in the way technology continues to transfer within industries regarding the formulation of the adoption case in the supply chain system. The theory outlines several factors like relative advantage, compatibility, complexity, and trialability that determine the adoption of Artificial Intelligence Digital Twins in different fields at a certain rate. The conclusions of this dissertation will illustrate the extent to which these factors influence the deployment and utilization of Digital Twins, focusing particularly on managing the technological, operational, and organizational challenges of implementing Digital Twins (Hui, 2024).

2.7 Literature Gap

Although there has been a vast amount of work published on DTs and application of AI in conjunction with DTs in different contexts, there seems to be a shortage of literature concerning the use of AI-improved DTs in the context of supply chain demand forecasting and inventory control (Ressi *et al.* 2024). The present literature review has mostly concerned with the general conceptual and technical context of DTs and their parts including smart manufacturing and industrial automation.

Moreover, numerous research has been conducted concerning the impact of AI in SCM however, such research has not been focused and directed toward specific applications like DTs (Monizza *et al.* 2024). The current literature lacks a detailed approach to optimally utilizing AI and DTs to forecast real-time demand and optimize inventory in industries with volatile demand fluctuations such as the retail and manufacturing industries.

Filling these gaps will help to advance the knowledge of how implementing AI-supported Digital Twins in supply chain networks is possible and how obstacles to its implementation can be overcome, as well as how effectiveness is attainable.

2.8 Conceptual Framework

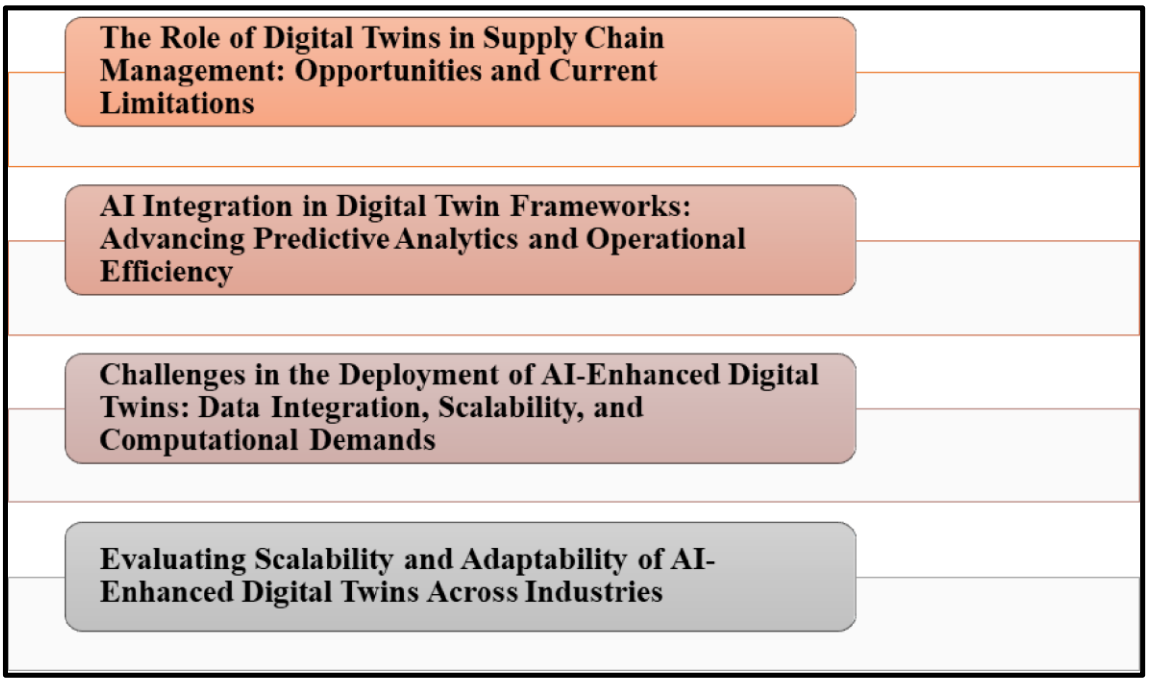

Figure 2.8.1: Conceptual Framework

(Source: Self-created)

2.9 Summary

Chapter three offers a comprehensive literature analysis of the DT, related technologies, and issues surrounding the use of AI-based DTs in a supply chain environment. In this context, the review reveals how incorporating artificial intelligence in the use of Digital Twins enriches opportunities in different operational areas, as well as in the forecasting of the demand and inventory (Ansari *et al.* 2020). When industries have adopted the Industrial 4.0 model, the usage of DTs in the supply chain has remained quite popular. To achieve these objectives, this chapter relies upon the following scholarly works: To identify how Digital Twins are helping organizations modernize their supply chains, to establish the enhancements driven by the inclusion of Artificial Intelligence and related issues and challenges Organizations face in deploying these two technologies.

DTs, are digital shadow copies of physical objects and are useful for decision-making since they continuously feed real-time data from the physical world to the digital. This capability increases the operational efficacy of an organization, as described in the review, but the key benefit of DTs is best realized when used in conjunction with AI for analysis (Falatouri *et al.* 2022). Digital twins created with or assisted by AI can be powerful when it comes to demand forecasting and inventory management that involve future trends, as a result helping with better decision-making.

The literature also finds the increasing role of AI in enhancing the effectiveness of Digital Twins. Machine learning and AI algorithms, as well as other techniques, improve the forecast precision of such models, thus enabling businesses to address variable demand patterns and reorganize their inventory levels. AI integration also makes the decision-making process automated, hence increasing efficiency and reducing cost. Still, the review shows that the AIenhanced Digital Twins' benefits, especially in the case of simulation and decision-making, suggest some of the obstacles one might expect when implementing them, especially concerning the nature and combination of data streams and computations involved. Such challenges related to data integrity, real-time processing, and applicability of the solution across different supply chain scenarios are some of the key contentions that are key barriers to implementation (Kulshrestha *et al.* 2020).

III. RESEARCH METHODOLOGY

3.1 Introduction

Chapter three of this study discusses the approach, philosophy, and methods that were used in studying the application of AI-enabled, fully-realized, predictive digital twins with SCM about SDP and inventory control. This chapter gives details on the philosophical underpinning of the study, research type, data collection tools, and data analysis tools.

The research uses the positivist philosophy, which contends that knowledge is created based on observable and quantifiable data (Matinheikki *et al.* 2022). The identification of this perspective corresponds well with the study aims as it is an objective assessment of the efficacy of the AI-enabled Digital Twins. Through the positivist approach, it is possible to gather data that can be compared with the hypothesis constituting the backbone of theoretical concepts about the efficiency and applicability of Digital Twins in supply chains.

The research method implemented in this study is experimental type where the researcher intervenes with the real reserve supply chain and implements several controlled observations and simulations to judge the potential of AI-driven Digital Twins. Thus, the experimental approach provides an opportunity to evaluate several hypotheses associated with the potential of these models as a tool for enhancing the accuracy of demand forecasts and inventory control. This research adopts descriptive research design since its main aim is to portray the nature, attributes, and behaviour of the phenomenon under consideration, hence offering a comprehensive look at the industry-specific operational issues and technology-related constraints to the integration of AI with Digital Twins. The research applies a secondary quantitative data collection method, to collect published data from papers, journals, current reports, case studies, and any other related sources. This secondary data enables us to perform a thorough scrutiny of trends, risks, and results relating to AI-based Digital Twin in SCM.

3.2 Research Philosophy

This dissertation will employ the positivism philosophy, which is one of the basic assumptions of scientific research that encourages the measurable use of data, and real stashing data to acquire knowledge and test hypotheses. Positivism has its foundation in the belief that the operation of reality can be measured without distortion of its actuality. However, it appears especially suitable within the given context of this research since it enables the collection of quantitative data and specific testing of hypotheses connected with demand forecast and inventory management in the context of AI-supported Digital Twins of the supply chain. The first and one of the key benefits arising from the positivism approach is the assumption that only objective knowledge is possible (Alabdali and Salam, 2022). Because this research focuses on the assessment of digital models and technologies, the accumulation of information and its subsequent analysis in a structured manner provides support to the conclusions from empirical data rather than conjectures. The positivism research approach is viewed as helpful in the evaluation of the efficiency and possibility of application of AI-supported Digital Twins in the context of different SC circumstances, while, at the same time, implying strict researcher impartiality when considering the analysed data.

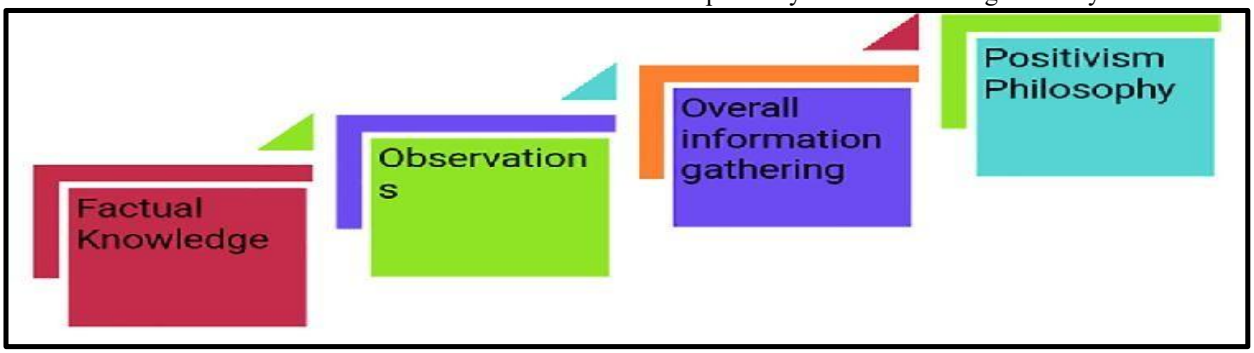

Figure 3.2.1: Positivism Research Philosophy

(Source: researchgate.net, 2024)

Moreover, positivism contributes to the replication of the study since it follows certain techniques and has sharp, tangible results. This increases the construct validity and generalizability of the research results on the effects of Digital Twin technologies for different conditions and sectors.

3.3 Research Approach

In this research, one of the main strengths of adopting an experimental approach is that it can provide an accurate causal link between the implementation of technologies of artificial intelligence and the effectiveness of supply chains (Li *et al.* 2021). In this way, by isolating such variables as AI algorithm settings or Digital Twins in demand forecasts, the researcher receives direct straightforward evidence of how certain variables affect supply chain performance, forecast accuracy, and operational costs. This results in a level of controlled environment that will allow an accurate and measurable assessment of the results to draw conclusions corresponding to objective and factual data.

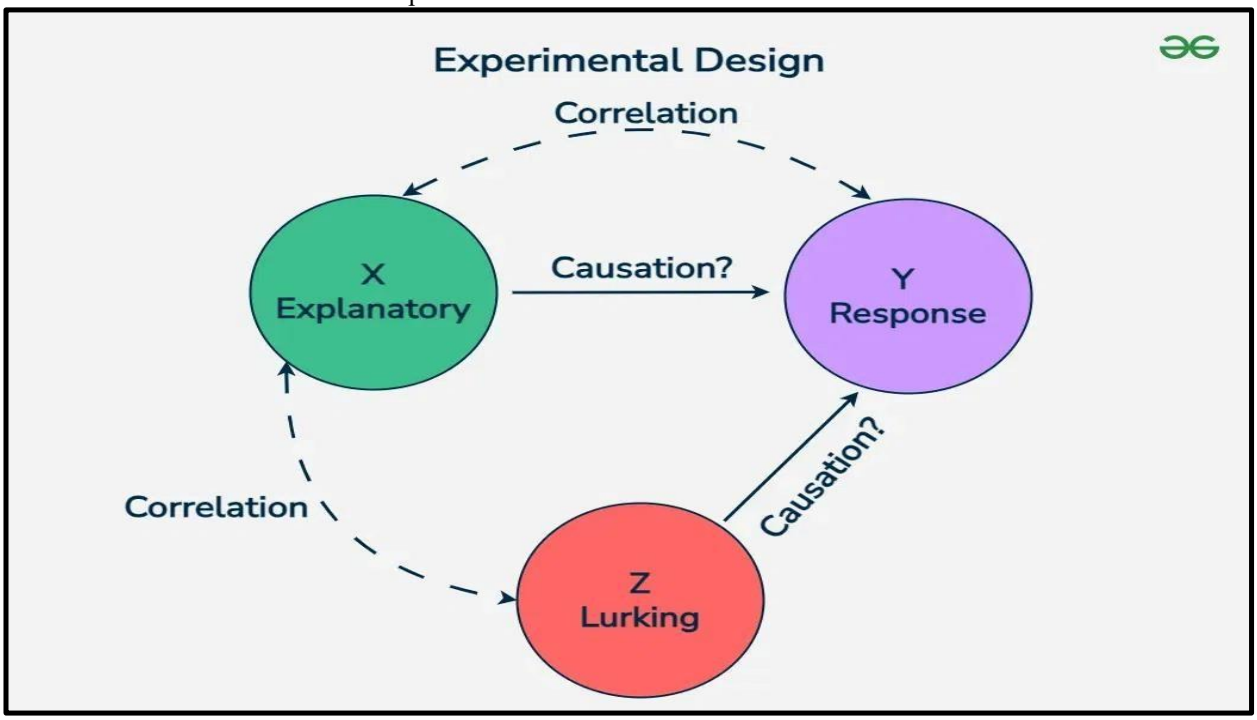

Figure 3.3.1: Experimental Research Design

(Source: geeksforgeeks.org, 2024)

Another benefit is that E2E field experiments do not have to lock one to a specific scenario: levels of demand variability one wants to study, or the use of AI in particular industries, be it retail, manufacturing, or logistics. This flexibility only benefits the research to some degree as it means that they get to see how the real environment affects the functioning of the proposed AI-enhanced Digital Twin model (Ivanov, 2024).

Furthermore, the structure of the experiment allows replication of the study thus enabling future research to test and verify the results presented in the study. This increases the reliability of the conclusions and contributes to the generalization of AI-based Digital Twin in supply chain scenarios.

3.4 Research Design

This dissertation therefore uses descriptive research design, which aims at identifying the characteristics of the research problem. In this context, the study looks at the Applied AI Digital

Twins focused on supply chain management with an understanding of how the predictive demand model, inventory regulation, and integrated operations are affected. Descriptive research design is especially appropriate in research that aims to describe a subject without the intention of altering it.

Therefore, the advantages that are enjoyed by this research through the application of a descriptive design cannot be underestimated (Pal, 2023). First, it enables one to gain insights into the current state of

Advanced Intelligent Digital Twin solutions by industries such as retail, manufacturing, and logistics industries. This design captures and evaluates the current practices in the technologies within the design given opportunities, challenges, and lights thus aiding in the development of the research objectives.

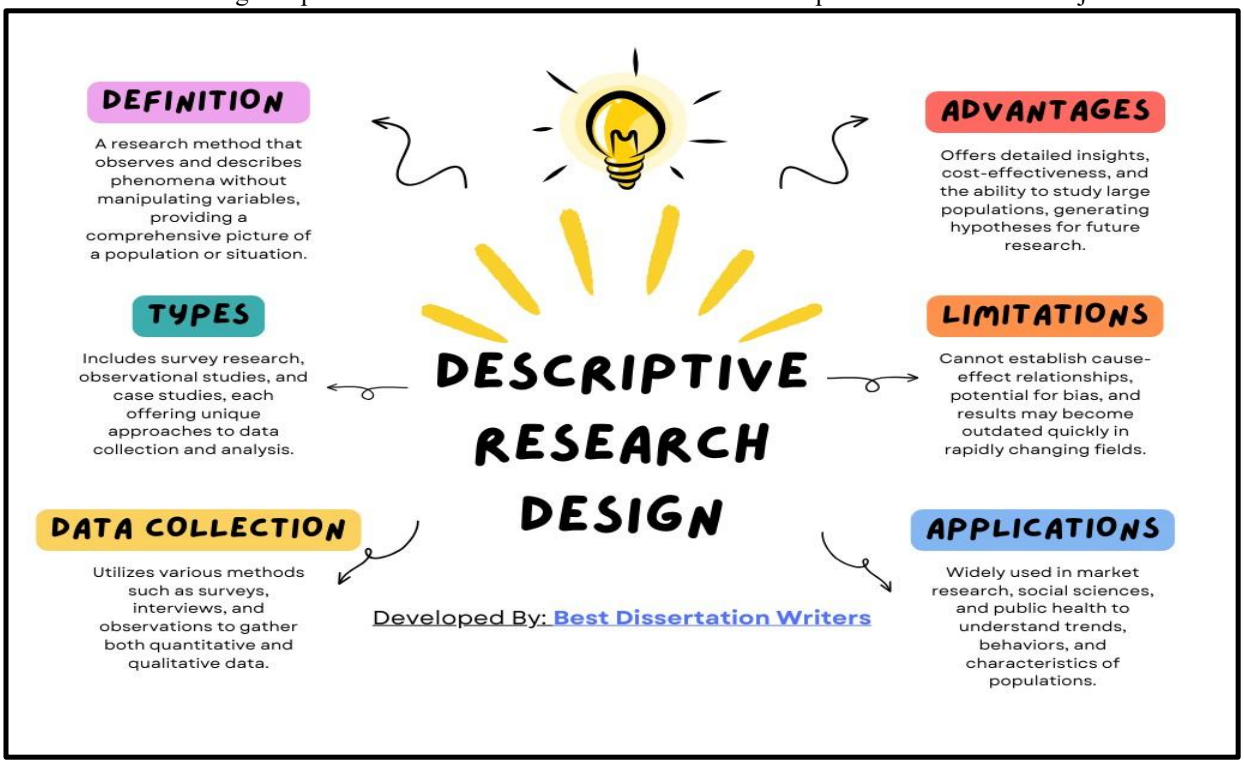

Figure 3.4.1: Research Design Descriptive

(Source: bestdissertationwriter.com, 2024)

Further, the descriptive design contributes to the creation of a clear reference model for how the advanced concept of AI-enriched Digital Twins is being implemented within supply chain contexts in various industry fields. In this way, the research can provide a detailed description of the technological, operational, and economic conditions and impacts affecting the adoption and performance of the systems. Furthermore, this approach enables the researcher to establish the relationships between variables which may be hard to determine in experiments (Agi and Jha, 2022). All these findings are valuable when assessing the feasibility and flexibility of integrated Digital Twins in more diverse scenarios. Also, as will be established in the subsequent sections of this paper, descriptive studies form a good base for experimental studies or case-based research that could expound more on causal relations.

3.5 Data Collection Method

This dissertation employs the secondary research technique of quantitative data collection to capture appropriate data for the study. Secondary data is data gathered by other researchers' organizations and institutions for purposes similar to the objectives of this dissertation. Here the secondary data sources will comprise reports, datasets, academic journals, industry reports/analysis, and case studies that look at AI-integrated Digital Twins in supply chain management. Below are the advantages of using the secondary quantitative data Collection method. First, sample sources are ready and compiled therefore making research and collection of data easier and faster. This saves a lot of time and energy that otherwise would have been used to conduct surveys or experiments and get primary data (Ahmetoglu *et al.* 2022). Secondarily data sources are huge in terms of data provision which can be valuable especially when analysing trends, patterns, and fair relations in different industries including manufacturing, retailing, and logistics.

Figure 3.5.1: Quantitative Research Methods
(Source: questionpro.com, 2024)

Another is that data collection in this case enables the historical data to be accessed so that long-term trends concerning the adoption and performance of AI-driven DT applications can be evaluated. This approach also makes it easier to work with reliable datasets compiled from previous research, which have already passed the tests of reliability by being published by qualified researchers on credible sources.

Moreover, secondary quantitative data testing implies the goal of revealing scalability and adaptability comparison across industries, which, when using only primary data collection, is rather difficult due to the amount and variety of data necessary for that. Hence, the secondary mode of quantitative data collection is another preferred and feasible approach to conducting this research to gain insights into the existing state of affairs, in AI-integrated Digital Twin implementation in supply chain systems.

3.6 Data Sources

In this dissertation, the dataset consists mostly of data that is sourced from the internet and other sources of open data, literature, and case studies. Internet resources, as the major sources of information, are important to ensure that the implemented AI-based Digital Twin technologies in various industries are diverse and current. These sources include and are not limited to Google Scholar, Publication Repositories, scholarly journals, government documents, and industry-specific Databases.

The use of online sources has several advantages over other sources of data collection. From these sources, it can get a variety of international studies and outcomes on AI in supply chain management, Digital Twins, and predictive analysis (Manning *et al.* 2023). These sources are also revised periodically, and therefore the information that fills the data tends to be current regarding trends or new technologies and problems. As the topic area of Digital Twins and AI in supply

chains is still relatively new, the use of online sources allows including all the most relevant and up-to-date material in the dissertation.

Further, data from online sources include any kind of case study, industry standards, and empirical research done by different scholars and organizations in different parts of the world. This richness of data provides a general view of the subject matter and is connected with topics various from retail to manufacturing and logistics.

However, it is worth understanding that access to quantitative databases can be obtained on the Internet, and they are necessary to achieve qualitative results of statistical analysis (Eyo-Udo, 2024). To this end, the involved data of this dissertation can provide insights into whether this AI-based Digital Twins approach is efficient and feasible for predictive demand forecasting and inventory management. Conducting the study with reliable sources that are from the online and public domain guarantees the relevance of the data to address the research objectives and, thus, strengthen the study.

3.7 Data Analysis

The evaluation of the obtained data used in this dissertation is implemented using a clear set of steps to guarantee the factual adequacy, specificity, and credibility of information. The data was initially collected from other internet-based sources such as peer-reviewed journals, industry reports, government documents, and datasets. These sources include both, the qualitative and quantitative data that are crucial for assessing the efficiency of enhanced Digital Twin by AI in making predictions regarding demand levels and inventory.

Data cleaning and preprocessing is the initial process in the data analysis process. This involves checking for and managing missing, incomplete, or inconsistent values to obtain only the right and resourceful values for analysis (Jahani *et al.* 2023). This also involves making data consistent with each other and performing the entire conversion of the data to another format. After data cleaning is done, probabilities and frequencies and all such analysis tools and techniques are used to analyse the data. As with any kind of statistics, descriptive first serves the purpose of presenting the data collected simply and giving an overview of the most important trends, patterns, and distributions. The results further assist in knowledge of the general cues of AI-assisted DTs in distinct supply chain scenarios.

In addition, complex methods like multiple regression analysis, correlation analysis, and machine learning techniques are used to analyse relationships mainly concerning how the DT models impact the measurement of demand forecast and stock control. These analyses assist in defining factors that enable recognition of the success or failure of the integrated Digital Twins with AI elements in various industries (Zamani *et al.* 2023).

The results obtained from these analyses are used to make conclusions regarding the scalability, flexibility, and efficiency of the AI-based DT models and to offer the outcomes, which correspond to the goals of the study.

3.8 Ethical considerations

The issue of ethical consideration is given seminal consideration in the research methodology of this dissertation by guaranteeing that the research process will be ethical. As this research is focused on quantitative data collection from public domain sources the major ethical issues related to privacy and ownership are not much of a concern here. But it is crucial to admit that the application of external data that is available to the public, accurate, and belongs to no one can be used, as a violation of Intellectual Property is strictly prohibited.

The dissertation also considers the aspect of ethical conduct in data analysis as well as the reporting of such results. Specifically, all the collected and analysed data shall be portrayed in an accurate manner free from spinning or contributing to the creation of misleading hypotheses (Shamsuzzoha *et al.* 2020). Moreover, every bias in the data collected shall be shown to make the study conducted more transparent as much as possible.

However, since the study does not involve people or particularly susceptible information, plagiarism policy will also be adhered to with strict consideration given to authors and contributors will always be made. These ethical principles seek to maintain the reliability and accuracy of the study to ensure a sensible addition to an effective knowledge base regarding the AI-enabled Digital Twine in the SC context.

3.9 Research limitations

The following limitations must not be overlooked as this dissertation seeks to advance understanding of how AI-enhanced Digital Twins can serve as an innovation in the management of supply chains: First, the research employs cross-sectional quantitative data that were obtained from the internet. The quality of these data is therefore a function of the quality of the basic datasets used in the analysis (Rahman *et al.* 2024). Inconsistency in data format or differences in reporting of findings and the type or volume of data obtained could influence the stability of the discoveries made.

Secondly, due to the nature of this research, which utilizes experimental research, sometimes the results may not be generalized in the investigated supply chain contexts examined through the existing data. This can also limit the realism in reproducing empirical real-world scenarios or primary qualitative architectures that may seem to affect the practical application of AIenabled Digital Twins (Jabbar *et al.* 2021).

One is the limitations of the analysis that is mainly done on the components of Demand Forecasting and Inventory Management. Although, this approach is good in answering the research objectives it may not capture the totality of the uses of Digital Twin technologies in industries.

Finally, the study only employs secondary data which has the disadvantage of making the researcher have no control over the kind of data to be collected for the research questions at hand.

3.10 Summary

This chapter has described the research method used in this dissertation and given an approach that would be used in the investigation of the application of AI-enhanced Digital Twins in supply chain management. The research uses the positivism philosophy to enhance a major theme of objective reality and emphasizes measurable occurrences to create outcome-oriented results that are verifiable through records and numerical data (Raja Santhi and Muthuswamy, 2022). An experimental approach was selected to assess the feasibility and future of the AIintegrated digital Twin philosophy and present a solid framework to validate the hypothesis and make inferences based on evidence.

The study adopts a descriptive research design since it seeks to review information and establish trends and useful information on AI-enhanced Digital Twins. Secondary quantitative data, which can largely be found in online databases, have been used to have extensive sources of data. This procedure is especially cheaper than if one had to undertake primary data collection, while at the same time making it possible to gather huge data (Kamble *et al.* 2022). Ethical issues which include data privacy, accuracy, and proper use of secondary data were considered in a bid to come up with quality research. The chapter also provided the limitations for example the fact that the study used secondary data and the nature of the study. Lastly, the research methodology developed in the present study is an appropriate framework for examining the research issues and contributes knowledge regarding AI-enhanced Digital Twins in SCM.

IV. FINDINGS AND ANALYSIS

4.1 Introduction

This chapter sums up the results of the study performed to meet the main research objectives outlined in the earlier section of this work. The paper draws data analysis from different machine learning algorithms employed for demand forecasting from the Retail Store Inventory Dataset. Then the model performance is assessed, major trends and patterns of the data are described, and the findings are discussed in the context of enhancing the supply chain management.

The initial part of the analysis will consider the features of time and variable characteristics, and seasonality and stochastics largely derived from the EDA. After that, the chapter presents the approaches used for feature engineering and data preprocessing that formed the basis for the appropriate training of the predictive models (Sezer *et al.* 2020). Given the time series nature of this problem, this study compared the performance of six models namely: The Random Forest Regressor, Temporal Convolution Network (TCN), Long Short-Term Memory (LSTM), Prophet, XGBoost, and N-BEATS; where MAE was adopted as the key measure of Model Accuracy. All these models were applied to the above pre-processed data set and experiments were carried out to choose the best hyperparameters for the models.

4.2 Dataset Overview

The data set used for this study is the Retail Store Inventory Data Set sourced from a public domain (Kang *et al.* 2020). This business data set is a time series that satisfies inventory control meant for unit daily sales, demanded forecasts, and the data attributes. The main purpose of the dataset is for demand and much of the data is used to assist retailers decide on appropriate stocks and inventory levels to hold to make the supply chain more efficient.

Key attributes in the dataset include:

- Units Sold: A measure of the demand, in the real world, allowing to track the demand over time and its changes.
- Demand Forecast: A positive figure representing the expected amount as a basis for comparing the forecast amount with the corresponding actual amount for the validation of the forecast.
- Temporal Variables: Fixed attributes including date and season that give directions on time changes and allow for trend and seasonal analyses.

	Date	Store ID	Product ID	Category	Region	Inventory Level	Units Sold	Units Ordered	Demand Forecast	Price	Discount	Weather Condition	Holiday/Promotion	Competitor Pricing	Seasonality
0	2022-01-01	S001	P0001	Groceries	North	231	127	55	135.47	33.50	20	Rainy	0	29.69	Autumn
1	2022-01-01	S001	P0002	Toys	South	204	150	66	144.04	63.01	20	Sunny	0	66.16	Autumn
2	2022-01-01	S001	P0003	Toys	West	102	65	51	74.02	27.99	10	Sunny	1	31.32	Summer
3	2022-01-01	S001	P0004	Toys	North	469	61	164	62.18	32.72	10	Cloudy	1	34.74	Autumn
4	2022-01-01	S001	P0005	Electronics	East	166	14	135	9.26	73.64	0	Sunny	0	68.95	Summer
5	2022-01-01	S001	P0006	Groceries	South	138	128	102	139.82	76.83	10	Sunny	1	79.35	Winter
6	2022-01-01	S001	P0007	Furniture	East	359	97	167	108.92	34.16	10	Rainy	1	36.55	Winter
7	2022-01-01	S001	P0008	Clothing	North	380	312	54	329.73	97.99	5	Cloudy	0	100.09	Spring
8	2022-01-01	S001	P0009	Electronics	West	183	175	135	174.15	20.74	10	Cloudy	0	17.66	Autumn
9	2022-01-01	S001	P0010	Toys	South	108	28	196	24.47	59.99	0	Rainy	1	61.21	Winter

Figure 4.2.1: Dataset Overview

Before proceeding with the extraction and formation of the models, data pre-processing was performed for the dataset. Measures such as age were scaled to equal ranges, whereas categorized variables were encoded if needed to fit the machine learning models. Preprocessing techniques specific to each time series included data transformation using the differencing technique to turn the data into a stationary form, dealing with the temporal/sequential structure of the data, and enhancing models' capability when dealing with sequential data (Shoeibi *et al.*
2024).

Even though all tables were characterized by a high temporal frequency and numerical values, it was initially unclear whether they would best fit the research problem or not The dataset was also more prepared for comprehensive preprocessing and creative feature engineering than one might expect (Aslam *et al.* 2021). In general, the dataset was critical in the assessment of the AI-driven digital twin to optimize inventory management by presenting a solid baseline for demand forecasting. Its comprehensive nature guaranteed the qualitative research results to be sound in response to the study goals.

4.3 Exploratory Data Analysis (EDA)

4.3.1 Time Series Analysis

Monthly Resampling

The variables were transformed to have a monthly frequency to get a clearer vision of changes that have taken place over time. This approach also made it possible to accumulate daily figures and then develop monthly means that made it easier to see the long-term trends because other trends seen in the short run were likely to be a result of random variation. Using average

numbers for the measures like “Units Sold” and “Demand Forecast”, the rise and fall in inventory and sales were therefore established.

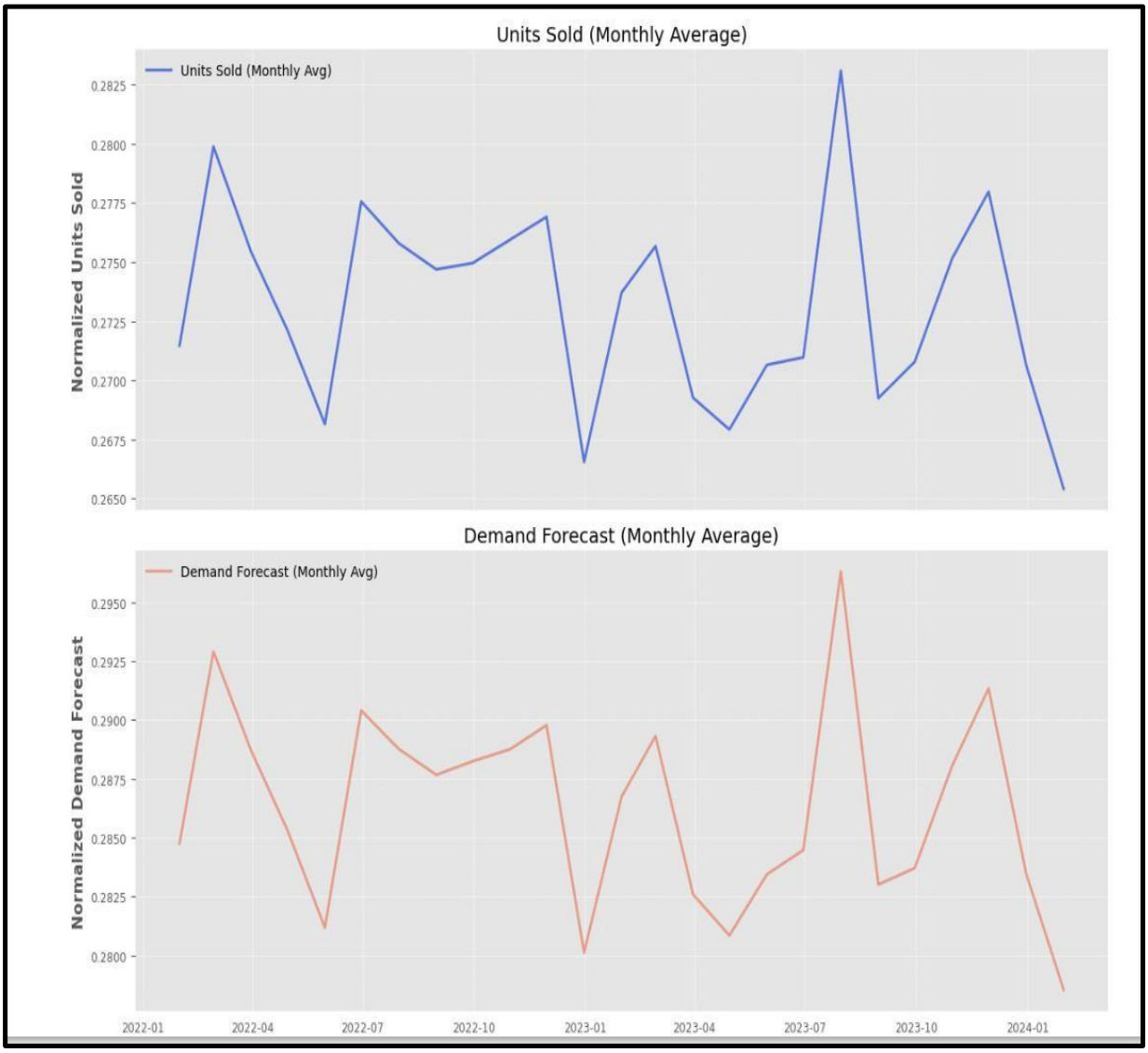

Figure 4.3.1.1: Time Series Plot

Visualizations of resampled data displayed periodicity and bursts of high or low popularity, which are associated with seasons and trends due to causes like holidays (Bharatiya, 2023). These insights were useful for identifying how demand changes with time and the subsequent choice of models capable of capturing them.

Seasonal Decomposition

Seasonal decomposition was performed to separate the time series data into its core components: trend, seasonal, and residual cycle. The breakdown also showed high cyclical fluctuation in all variables especially in ‘Units Sold’, which depicted the annual sales cycle.

The trend component pointed out that changes were steady over the long term, they may be due to large-scale market forces or perhaps growth trends.

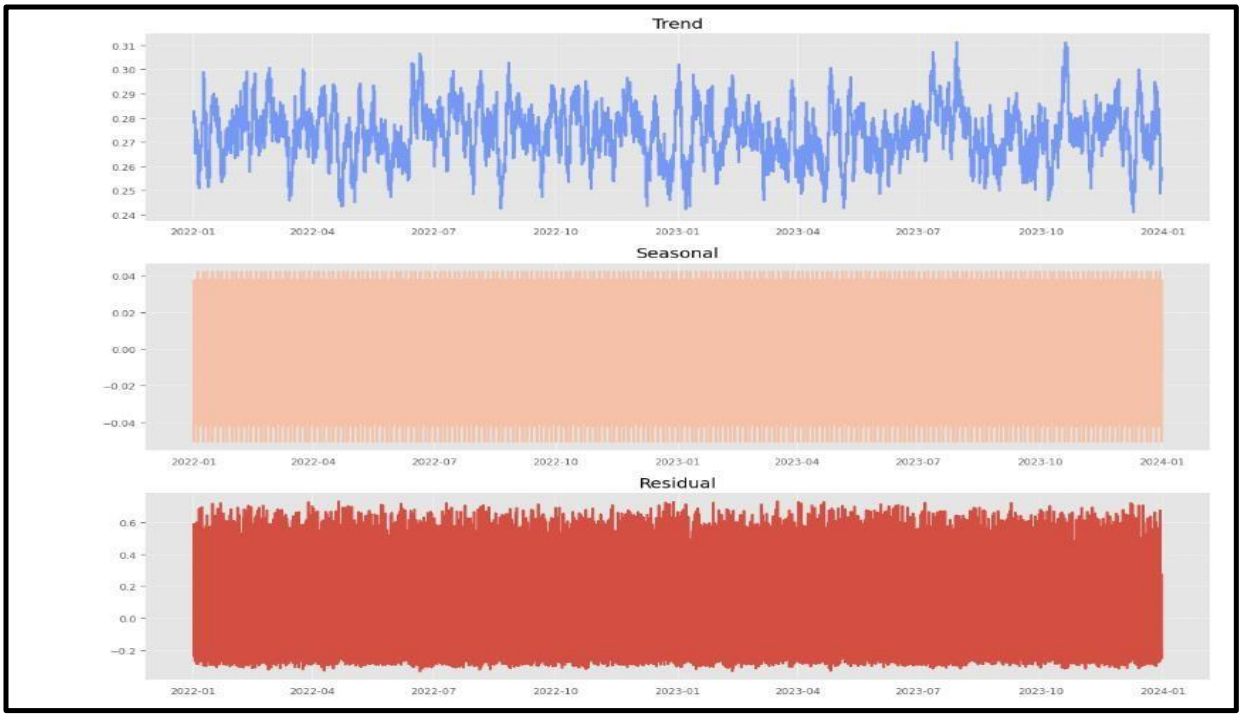

Figure 4.3.1.2: Time Series Plot Seasonal

The remaining part of the signal showed the random component displaying the variation that could not be attributed to cycles or seasonality (Zhang *et al.* 2021). This decomposition was useful to get more insight into the chronology of the underlying time series so that only the pertinent aspects of the sales data for establishing a forecast were singled out for modeling.

Autocorrelation and Partial Autocorrelation Analysis

ACF and PACF were used to look at their relationships between lagged observations expressed as time series. The results of the analysis based on the ACF plot described a decrease down to an approximate level of 0.5 at lag 1, which indicates that the observations in the variable are strongly related to the observations in the previous period.

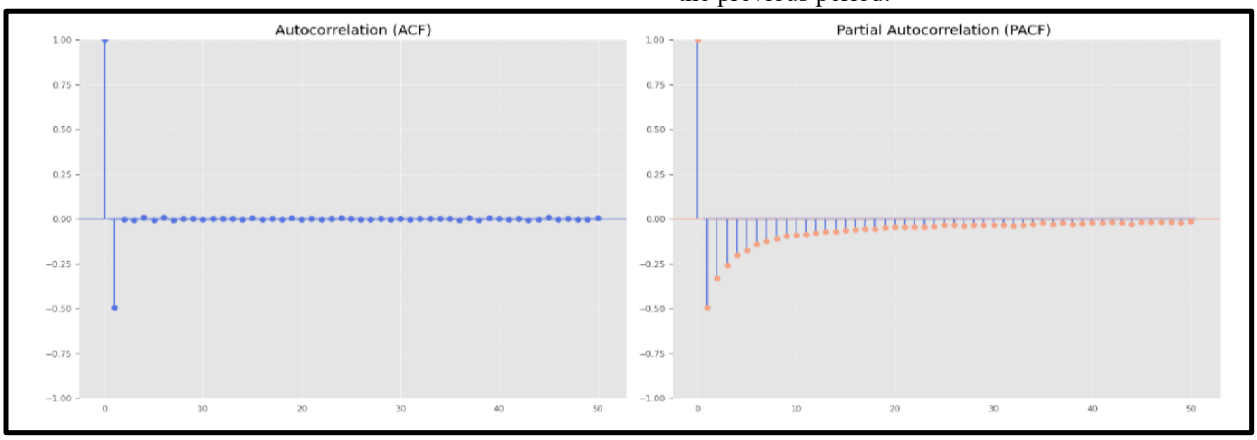

Figure 4.3.1.3: Autocorrelation and Partial Autocorrelation Analysis

The PACF was also still relatively high for the first lag, which points to an AR (1) process where the current observation is largely influenced by the first previous observation. These results justified the choice of models such as ARIMA and other time series forecasting models since the data exhibited a high degree of autocorrelation in the data (Lee and Lee,

2022). Feature engineering was also informed by this analysis to determine lags to incorporate as predictors.

4.3.2 Distribution Analysis

Histogram with Kernel Density Estimation

To dissect the shape of the distribution for "Units Sold," histograms with KDE curves overlaid were constructed. The histogram represented the frequency of exploitation of the sales data and KDE was the non-parametric technique of reparative density.

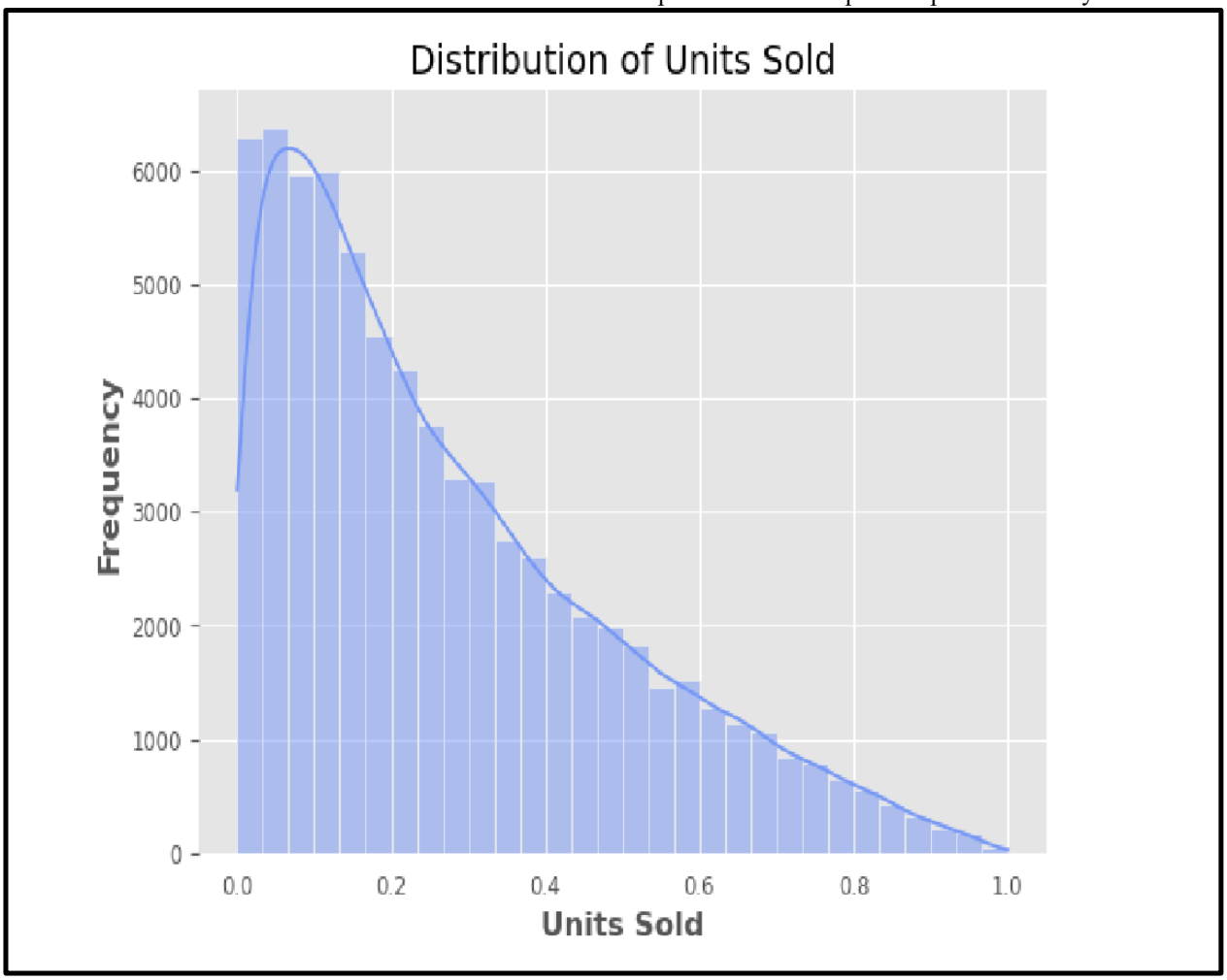

Figure 4.3.2.1: Distributions of Units Sold

It also showed that the data followed a positively skewed distribution, meaning that there were many observations close to the smaller values of the variable and only a few observations at the higher end of the range. This pattern suggests fluctuations in demand and the requirement that models must observe low-demand intervals and high-demand intermissions. It was important to understand this distribution to match the models to the given data set and reduce the likelihood of measured errors in predictions.

Boxplots

Whenever there was the need to consider the outliers in the numerical features such as "Units Sold" and "Demand Forecast," boxplots were used as a tool to help spot these. Hypothetically, outliers were identified as those values, which stood outside the Interquartile range (IQR), meaning high/low demand at certain intervals (Amjad *et al.* 2022).

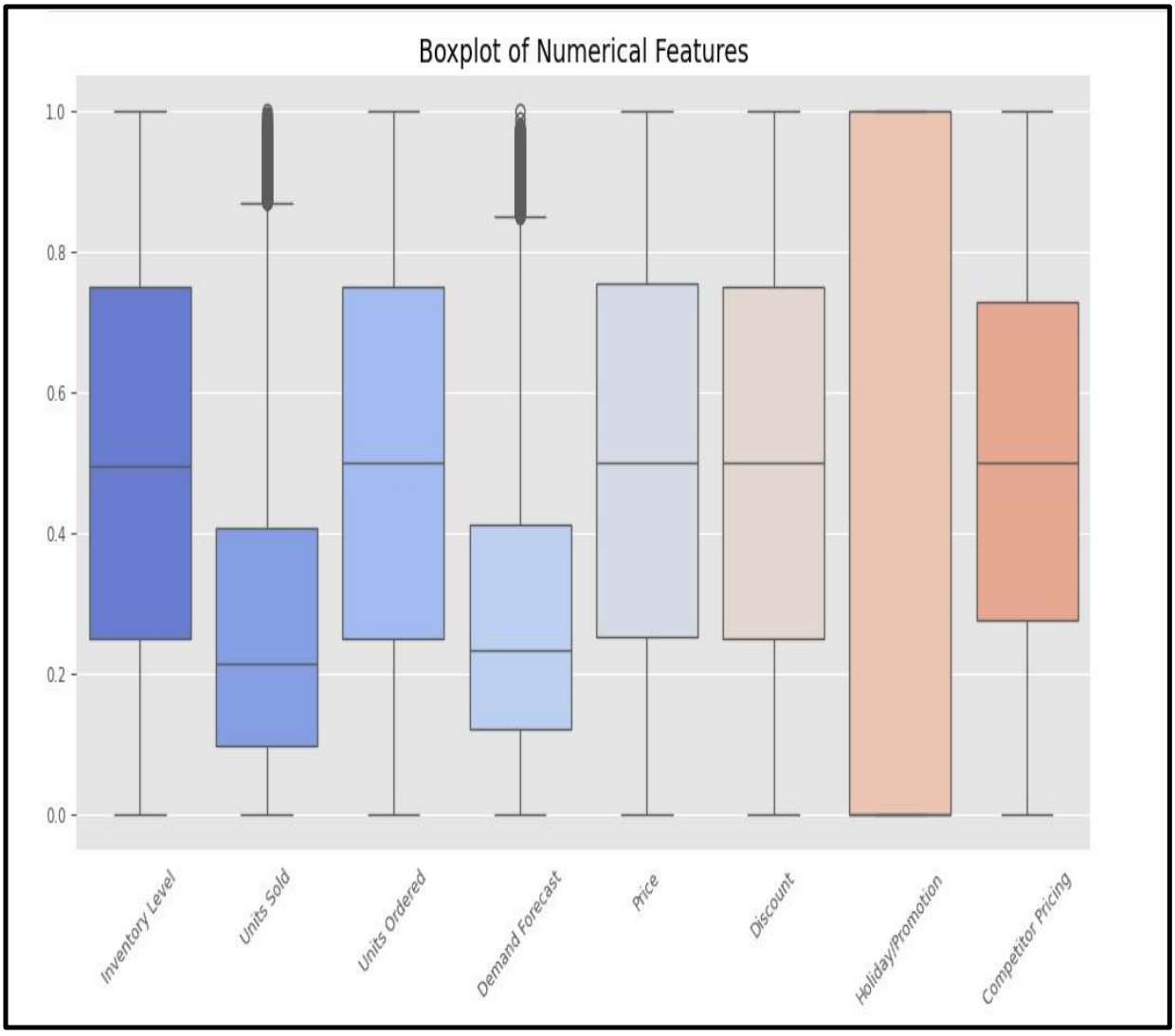

Figure 4.3.2.2: Boxplot of Numerical Features

To assess the consequences of these anomalies, the latter were analysed to identify them. Some of these had to keep to maintain the time series while others used imputation or even capping to avoid skews in prediction. The boxplots also showed the way the variability around the mean was distributed and underscored the need for accurate models that are likely to handle deviations in sales data (Xian *et al.* 2022).

4.3.3 Correlation Analysis

Heatmap

To gauge the associations between some of the temporal variables in the dataset like "Units Sold," "Demand Forecast," and so on, a correlation heatmap was created. It was clear from the heatmap that darker colours represented a higher correlation index. As evidenced by figures two and three above, there exist high positive coefficients between "Demand Forecast" and "Units Sold" confirming the accuracy of forecasted values about actual sales. Other moderate correlations were observed between temporal features and sales patterns, corroborating the integration of time variables into each of the models (Islam *et al.* 2021). Feature reduction was done using this analysis, hence we only considered attributes with high correlation to the target variable and excluded those that had poor correlation with the target variable.

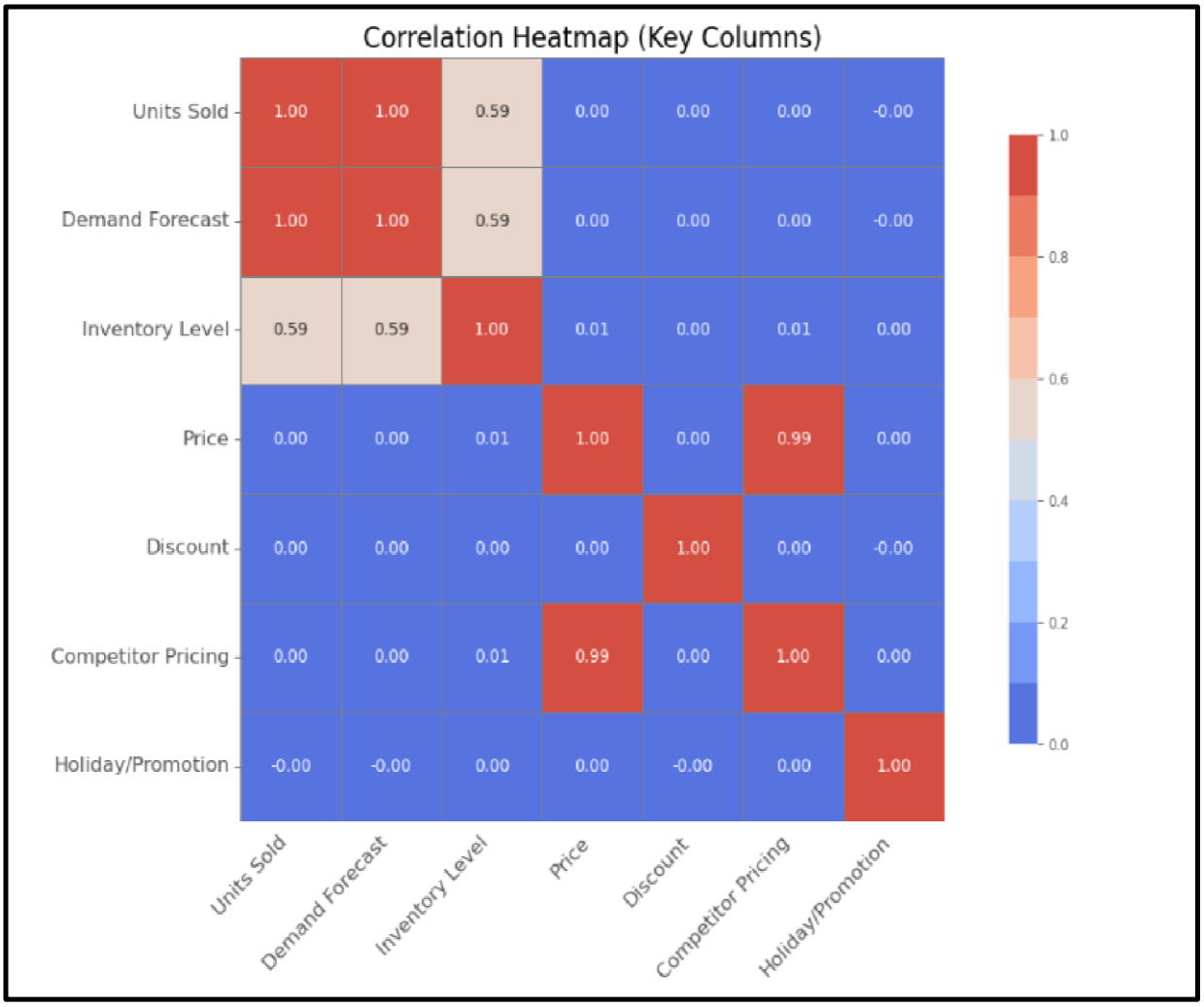

Figure 4.3.3.1: Correlation Heatmap

The heatmap also helped in dealing with multicollinearity issues, so that no variables with high correlation were retained in the model (Rahman *et al.* 2022). To identify these dependencies, the analysis primarily relied on the analysis of the most important relationships, which ultimately improved the effectiveness of the machine learning models.

4.4 Model Selection and Performance Evaluation

4.4.1 Overview of Implemented Models

Overview of Implemented Models

The objectives of demand forecasting and supply chain operations were addressed with several models including a diverse set. Every model was selected because of its high effectiveness in specific aspects of the dataset compilation process, strengths that include ensemble learning, prevalent deep learning frameworks, etc. The following models are characterized in this section together with their configurations and justification of their choice.

Random Forest Regressor

The Random Forest Regressor was chosen in this work due to its flexibility and social applicability that permits evaluating unique features consisting of both numerical and categorical data (Zheng *et al.* 2024). It has 'ensemble-based learning', whereby many decision trees will be used in the learning to encompass an over-reliance on any one decision tree that will more often make wrong assumptions during the learning process alongside giving the benefit of averaging the results. Some of the hyperparameters included in the multiclass model were tuned for the sake of improved performance.

Random Forest Regressor

```
rf_param_grid = {
    "n_estimators": [50, 100, 200],
    "max_depth": [None, 10, 20],
    "min_samples_split": [2, 5, 10],
    "min_samples_leaf": [1, 2, 4]
}
best_rf_score = float("inf")
best_rf_params = None
for params in ParameterGrid(rf_param_grid):
    print(f"Training Random Forest with params: {params}")
    rf_model = RandomForestRegressor(**params, random_state=42)
    rf_model.fit(train_rf_xgb, train['Units Sold'])
    rf_predictions = rf_model.predict(test_rf_xgb)
    rf_score = mean_absolute_error(test['Units Sold'], rf_predictions)

    if rf_score < best_rf_score:
        best_rf_score = rf_score
        best_rf_params = params

print(f"Random Forest Best Params: {best_rf_params}, MAE: {best_rf_score}")
results['Random Forest'] = {"Best Params": best_rf_params, "MAE": best_rf_score}
```

Figure 4.4.1.1: Random Forest Regressor Code

- n_estimators: To strike a balance between computational budget and prediction performance, a set of grid points spanning [50, 100, 200] was tested.
- max_depth: DTO values of [None, 10, and 20 were fixed to balance tree complexity to avoid both, underfitting and overfitting.
- min_samples_split: Validations of the range [2, 5, 10] were performed for adequate splitting of the data especially in cases with tiny samples.
- min_samples_leaf: Leaf sizes of [1, 2, 4] were used to control over-complexity while learning with the method of cross-validation to perform a generalization.

These parameter values were chosen by industry standards to allow the Random Forest Regressor to learn the possible complex patterns without falling for the train data set and thus overfitting (Alkhayat and Mehmood, 2021). It is a highly procedural type of AI and its applicability to the data made it the first kind of a choice.

XGBoost

XGBoost model was used, due to its ability to work with large datasets and ability to provide accurate prediction by gradient-boosted decision trees. Known for its fast computation and ability to prevent overfitting through regularization, XGBoost was fine-tuned with the following hyperparameters:

XGBoost Model

```
xgb_param_grid = {
    "n_estimators": [50, 100, 200],
    "max_depth": [3, 6, 10],
    "learning_rate": [0.01, 0.1, 0.2],
    "subsample": [0.8, 1.0],
    "colsample_bytree": [0.8, 1.0],
    "tree_method": ["hist"]  # Specify tree_method explicitly
}
best_xgb_score = float("inf")
best_xgb_params = None
for params in ParameterGrid(xgb_param_grid):
    print(f"Training XGBoost with params: {params}")
    xgb_model = XGBRegressor(**params, random_state=42, enable_categorical=True)
    xgb_model.fit(train_rf_xgb, train['Units Sold'])
    xgb_predictions = xgb_model.predict(test_rf_xgb)
    xgb_score = mean_absolute_error(test['Units Sold'], xgb_predictions)

    if xgb_score < best_xgb_score:
        best_xgb_score = xgb_score
        best_xgb_params = params

print(f"XGBoost Best Params: {best_xgb_params}, MAE: {best_xgb_score}")
results['XGBoost'] = {"Best Params": best_xgb_params, "MAE": best_xgb_score}
```

Figure 4.4.1.2: XGBoost Model Code

- n_estimators: The number of boosting rounds was set as [50, 100, 200], to find out how many boosting rounds are enough to increase model complexity but not performance overtrain.
- max_depth: [3 6 10] was experimented with as hyperparameters that restrict tree depth to achieve the best split between underfitting and overfitting.

learning_rate: The choice of [0.01, 0.1, 0.2] was conceived as a control on the rate of gradient update allowing it to converge gradually (Sharma *et al.* 2021).

- subsample: Ratios of [0.8, 1.0] were applied to add an extra layer of model stability where training was performed on a portion of the data set.
- colsample_bytree: Versions of [0.8, 1.0] were tested to control the feature selection in the splits of trees, helping avoid the building of repetitive trees.
- tree_method: The "hist" method was chosen for its fast and easy execution across massive datasets.

N-BEATS

For this study, the chosen univariate time series model was N-BEATS because of the dependence of this algorithm on long-range dependencies (Bi *et al.* 2021). It had fully connected neural architecture hence making it easy to model the trend and seasonality patterns. The hyperparameter grid for N-BEATS was designed to align with the dataset's requirements:

NBeats Model

```
nbeats_param_grid = {
    "input_chunk_length": [30, 60],
    "output_chunk_length": [7, 14],
    "num_stacks": [2, 3],
    "num_blocks": [1, 2, 3],
    "layer_widths": [128, 256, 512]
}
best_nbeats_score = float("inf")
best_nbeats_params = None
for params in ParameterGrid(nbeats_param_grid):
    print(f"Training N-BEATS with params: {params}")
    nbeats_model = NBEATSModel(**params, random_state=42, n_epochs=50)
    nbeats_model.fit(ts_train)
    nbeats_predictions = nbeats_model.predict(len(ts_test))
    nbeats_score = mae(ts_test, nbeats_predictions)

    if nbeats_score < best_nbeats_score:
        best_nbeats_score = nbeats_score
        best_nbeats_params = params

print(f"N-BEATS Best Params: {best_nbeats_params}, MAE: {best_nbeats_score}")
results['N-BEATS'] = {"Best Params": best_nbeats_params, "MAE": best_nbeats_score}
```

Figure 4.4.1.3: NBeats Model Code

- input_chunk_length: Values of [30, 60] were tested to estimate the historical data period used in the forecasts.
- output_chunk_length: Different forecast horizons of [7, 14] were examined to meet the needs of the business.
- num_blocks: Dynamic adaptation of granularity of learned patterns involved a range of [1, 2, 3].

layer_widths: After testing various layer sizes of [128, 256, 512], it was found that such configurations allow for better learning of complex data dependencies.

Temporal Convolutional Network (TCN)

This work has been conducted within the framework of Temporal Convolution Network because the architecture of the network allows to analyse short and long term dependencies on the sequential data (Alsharef *et al.* 2022). In order to model temporal patterns necessary in the modeling of time series data TCNs employs dilated convolution.

Temporal Convolution Network Model

```
tcn_param_grid = {
    "input_chunk_length": [30, 60],
    "output_chunk_length": [7, 14],
    "num_filters": [16, 32],
    "kernel_size": [3, 5],
    "dropout": [0.1, 0.2]
}
best_tcn_score = float("inf")
best_tcn_params = None
for params in ParameterGrid(tcn_param_grid):
    print(f"Training TCN with params: {params}")
    tcn_model = TCNModel(**params, random_state=42, n_epochs=30)  # Reduced epochs
    tcn_model.fit(ts_train)
    tcn_predictions = tcn_model.predict(len(ts_test))
    tcn_score = mae(ts_test, tcn_predictions)

    if tcn_score < best_tcn_score:
        best_tcn_score = tcn_score
        best_tcn_params = params

print(f"TCN Best Params: {best_tcn_params}, MAE: {best_tcn_score}")
results['TCN'] = {"Best Params": best_tcn_params, "MAE": best_tcn_score}
```

Figure 4.4.1.4: Temporal Convolutional Network (TCN) Code

- output_chunk_length: Also, the forecast horizons of [7, 14] were chosen as more flexible for use with time series forecasting.
- kernel_size: Hypothesis 3 and 4 proposed sizes of [3, 5] were tested to effectively control information aggregation over time steps.
- Dropout: Cross entropy with regularization values of [0.1, 0.2] was adopted during learning to avoid overfitting.

Through a systematic adjustment of these parameters, TCN presented itself as a potent model for time series forecasting, especially for datasets, which include sequential information.

Long Short-Term Memory (LSTM)

LSTM Model

```
scaler = MinMaxScaler()
lstm_train_scaled = scaler.fit_transform(train["Units Sold"].values.reshape(-1, 1))
lstm_test_scaled = scaler.transform(test["Units Sold"].values.reshape(-1, 1))

X_train, y_train = [], []
window_size = 90
for i in range(window_size, len(lstm_train_scaled)):
    X_train.append(lstm_train_scaled[i - window_size:i, 0])
    y_train.append(lstm_train_scaled[i, 0])
X_train, y_train = np.array(X_train), np.array(y_train)

X_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))

lstm_model = Sequential([
    LSTM(256, activation="relu", return_sequences=True, input_shape=(X_train.shape[1], 1)),
    Dropout(0.2),
    LSTM(256, activation="relu", return_sequences=True),
    Dropout(0.2),
    LSTM(128, activation="relu"),
    Dropout(0.2),
    Dense(1)
])

optimizer = Adam(learning_rate=0.001)
lstm_model.compile(optimizer=optimizer, loss="mse")

early_stopping = EarlyStopping(monitor="loss", patience=10, restore_best_weights=True)
reduce_lr = ReduceLROnPlateau(monitor="loss", factor=0.5, patience=5, min_lr=1e-5, verbose=1)

lstm_model.fit(X_train, y_train, epochs=150, batch_size=128, verbose=1, callbacks=[early_stopping, reduce_lr])

X_test = []
for i in range(window_size, len(lstm_test_scaled)):
    X_test.append(lstm_test_scaled[i - window_size:i, 0])
X_test = np.array(X_test).reshape(-1, window_size, 1)

lstm_predictions = lstm_model.predict(X_test)
lstm_predictions = scaler.inverse_transform(lstm_predictions)

lstm_score = mean_absolute_error(test["Units Sold"].values[window_size:], lstm_predictions[:, 0])
print(f"LSTM MAE: {lstm_score}")
```

Figure 4.4.1.5: LSTM Model Code

LSTM was considered for its capability to capture sequential dependencies in a given set of data particularly when it is time series data. To generate the model, three LSTM layers were used, containing [256, 256, 128] units, including the dropout layer (0.2) to avoid the problem of overfitting and the output layer was also used. Based on this data, MinMaxScaler was used to scale it down and the data was reshaped to fit LSTM input (Sreerama *et al.* 2023).

Prophet

Prophet Model

```
prophet_train = train.reset_index().rename(columns={train.index.name: "ds", "Units Sold": "y"})
prophet_test = test.reset_index().rename(columns={test.index.name: "ds", "Units Sold": "y"})

prophet_model = Prophet(
    seasonality_mode="multiplicative",
    yearly_seasonality=True,
    weekly_seasonality=True,
    daily_seasonality=False,
    changepoint_prior_scale=0.05
)

prophet_model.add_seasonality(name="quarterly", period=90, fourier_order=5)

prophet_model.fit(prophet_train)

future = prophet_model.make_future_dataframe(periods=len(test))
forecast = prophet_model.predict(future)

prophet_score = mean_absolute_error(prophet_test["y"].values, forecast["yhat"][:len(test)])
print(f"Prophet MAE: {prophet_score}")
results['Prophet'] = {"MAE": prophet_score}
```

Figure 4.4.1.6: Prophet Model Code

(Source: Implemented in Python)

Prophet was used due to its ability to meet some objectives including simplicity to model when data has trends and seasonality respectively. The model configuration also entailed a multiplicative seasonality mode with inherent yearly and weekly seasonality. Also, a new seasonality of the quarterly time scale was introduced to incorporate intermediate cycles. The model was built based on past data and its deployment was assessed on MAE (Nimmagadda, 2021). However, as simple models, Prophet's AUC scores were not as high as those of more complicated models like XGBoost and TCN.

4.4.2 Performance Evaluation

There were used Mean Absolute Error (MAE) was used to measure the accuracy of outputs obtained from each model in regression problems. The results in terms of the MAE values are presented and compared for the four models, as well as the implications of the findings.

Random Forest Regressor

Thus, applying the Random Forest Regressor approach, the MAE score was equal to 0.015842 and indicates good accuracy within the dataset. Because the current algorithm could consider both numerical and categorical variables, it had a high precision in capturing the relationships involved (Zhang *et al.* 2021).

```
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 10, 'n_estimators': 200}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 100}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 200}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 50}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 100}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 200}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 50}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 100}
Training Random Forest with params: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 200}
Random Forest Best Params: {'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50}, MAE: 0.0158423859680002
```

We decided to go with the Random Forest model as the first model because it can handle a large amount of data, including both numerical and categorical features, making it really useful for capturing complex patterns with its ensemble learning method. This choice also assures us that overfitting will not be a problem because this method provides an average of multiple trees. The concrete ranges of the hyperparameters were based on usual tuning practices.

n_estimators=[50, 100, 200]: Controls the number of trees, providing a balance between accuracy and computational efficiency.

max_depth=[None, 10, 20]: Controls the complexity of each tree to avoid overfitting without underfitting the data.

min_samples_split=[2, 5, 10] and min_samples_leaf=[1, 2, 4] ensure that tree splits and their leaf sizes are meaningful to assure better generalization on unseen data.

Figure 4.4.2.1: Random Forest Regressor MAE

Converting the original model into an ensemble model allowed us to avoid overfitting and hyperparameter tuning enhanced the model. However, averaging the outputs could reduce the model's ability to handle large deviations in the data which would be of importance in modeling the extreme demand rates.

XGBoost

In last, XGBoost has been determined to be the most effective model with a total MAE of 0.014761. Due to its gradient-boosting decision tree model, it was able to deal with the complexities residing in the dataset (Ensafi *et al.* 2022).

```
XGBoost Best Params: {'colsample_bytree': 0.8, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 200, 'subsample': 1.0, 'tree_method': 'hist'},
MAE: 0.014760508138329404
```

We chose the XGBoost model as our second model for its ability to handle complex datasets efficiently, providing high accuracy with faster computation through gradient-boosted decision trees. This model excels in dealing with both categorical and numerical data, making it ideal for our dataset. The hyperparameter grid was carefully designed to balance model complexity and computational cost.

n_estimators [50, 100, 200]: Determines the number of boosting rounds; smaller values prevent overfitting while larger values ensure robust learning.

max_depth [3, 6, 10]: Controls tree depth, balancing underfitting for shallow trees and overfitting for deeper ones.

learning_rate [0.01, 0.1, 0.2]: Affects the step size in gradient updates; lower values allow for gradual optimization, while higher values converge faster.

subsample [0.8, 1.0]: Prevents overfitting by sampling a fraction of data for training each tree.

colsample_bytree [0.8, 1.0]: Regulates the fraction of features considered for splitting, promoting diversity in trees and reducing overfitting.

tree_method ["hist"]: Optimized for efficiency in large datasets, using a histogram-based split finding algorithm.

Figure 4.4.2.2: XGBoost MAE

Each feature was fine-tuned to control fluctuation in both accuracy and computational intensity through modifications on the learning rate, tree depth, and subsample rates. The overfitting was well handled through regularization, and this made XGBoost even perform better (Otchere *et al.* 2021). These results provide evidence that it is well-tuned for specific demand forecasting tasks, where accuracy and scalability are important.

N-BEATS

The N-BEATS model caught long-range dependencies and seasonality patterns to keep an MAE of 0.177549 (Zeroual *et al.* 2020). However, the accuracy it achieved was not as great as ensemble methods such as XGBoost and Random Forest.

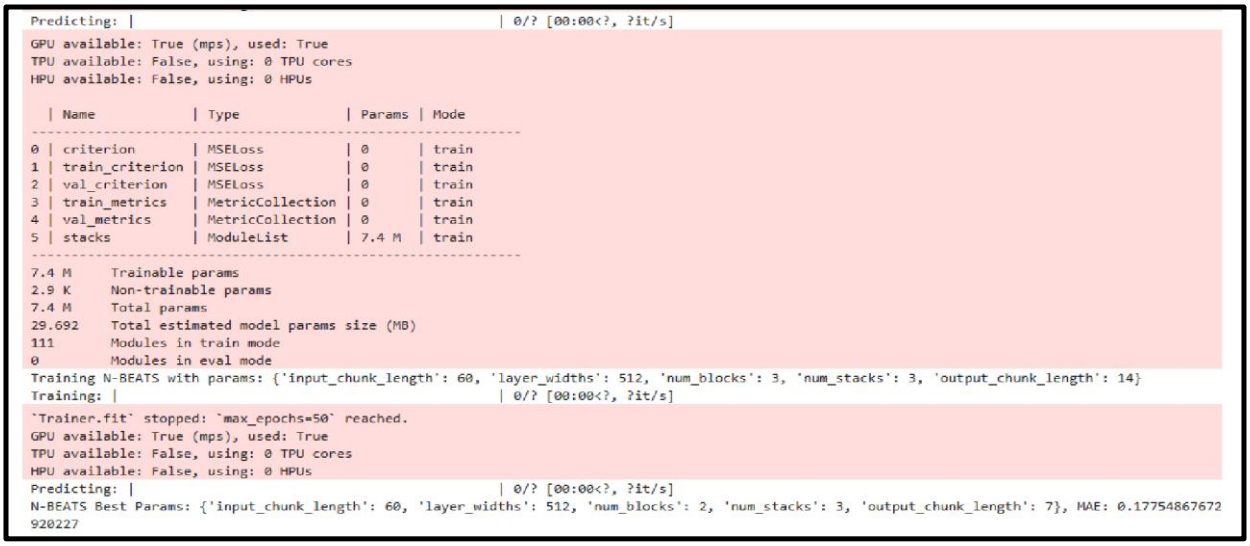

Figure 4.4.2.3: N-Beats MAE

The main drawback of the proposed model was that its completely connected neural structure may have demanded much computing power, which may hinder future extension of the model for higher dimensionality of data sources.

Temporal Convolutional Network (TCN)

TCN achieved an MAE of 0.171592 to test its ability to manage data existing in a time series. Dilated convolutions facilitated the incorporation of short-term and long-term structures of the problem inside the model (Zhang *et al.* 2021).

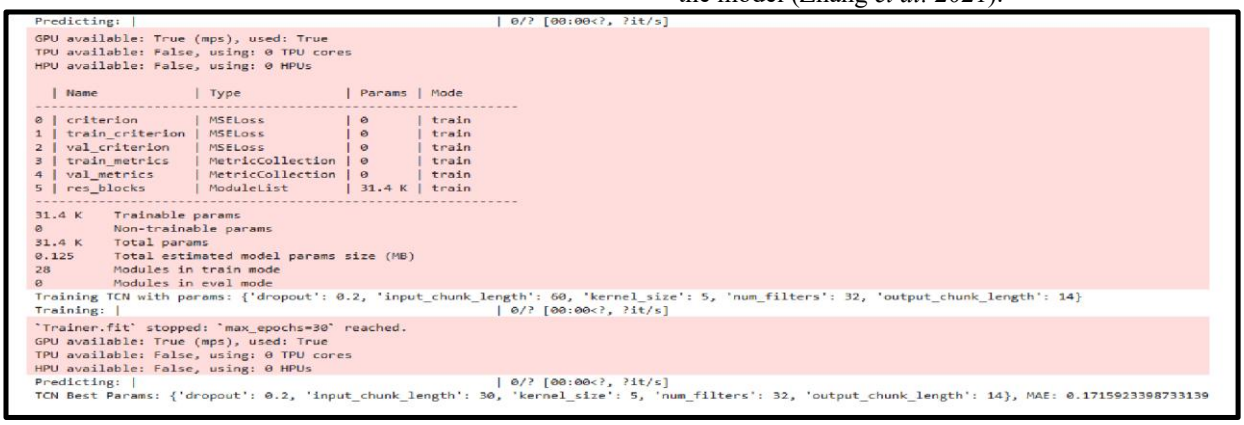

Figure 4.4.2.4: Temporal Convolutional Network (TCN) Performance Score

Its performance was then boosted with hyperparameter-tuned control adjustments of kernel size as well as dropout. Nevertheless, TCN provided slightly lower accuracy than both LSTM and XGBoost, which suggests some shortcomings in terms of variability of the used data set.

Long Short-Term Memory (LSTM)

LSTM tested an MAE of 0.167026, which indicates the accurate tuning of LSTM to manage different sequential trends in a given data set. The aspects that followed addressing the model's architecture were several LSTM layers and dropout to handle cases of overfitting (Falatouri *et al.* 2022). However, LSTM's training process was very time-consuming, and even though its performance was very good, it was slightly weaker than XGBoost.

```
Epoch 24: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
4/4 ━━━━━━━━━━━━━━━━━━━━ 2s 345ms/step - loss: 0.0695 - learning_rate: 5.0000e-04
Epoch 25/150
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 345ms/step - loss: 0.0677 - learning_rate: 2.5000e-04
Epoch 26/150
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 355ms/step - loss: 0.0675 - learning_rate: 2.5000e-04
Epoch 27/150
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 353ms/step - loss: 0.0649 - learning_rate: 2.5000e-04
Epoch 28/150
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 357ms/step - loss: 0.0719 - learning_rate: 2.5000e-04
Epoch 29/150
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 360ms/step - loss: 0.0651
Epoch 29: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
4/4 ━━━━━━━━━━━━━━━━━━━━ 1s 360ms/step - loss: 0.0656 - learning_rate: 2.5000e-04
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 138ms/step
LSTM MAE: 0.1670259001219286
```

Figure 4.4.2.5: Long Short-Term Memory (LSTM) Performance Score

The presented results show that forecasting with the suggested model is appropriate for the data with expressed temporal characteristics, at the same time they reveal the important indexing problem which can be solved with the help of efficient methods of model training on large data sets (Makumbura *et al.* 2024).

Prophet

MAE in this evaluation was also the smallest, and Prophet was below all other models with a value of 0.184475. In one sense, it fared well in capturing trends and seasonality, but its basic structure did not permit it to deal with the data.

```
01:27:09 - cmdstanpy - INFO - Chain [1] start processing
01:27:09 - cmdstanpy - INFO - Chain [1] done processing
Prophet MAE: 0.18447469795913723
```

Figure 4.4.2.6: Prophet Model Performance Score

Prophet's reliance on these predefined seasonality's appears to have limited the model's flexibility allowing larger prediction errors compared to models such as as XGBoost or LSTM.

Comparison and Analysis

The findings of the paper show that XGBoost is the best model yielding the lowest MAE of 0.014761 for demand forecasting. Due to high accuracy, time efficiency, and good performance toward the dataset's size, it is suitable for the dataset (Yuan *et al.* 2022). Random Forest and LSTM also provided good results which will assure that these models can work with more complicated data and those containing sequential information.

```
                                                  Best Params       MAE
Random Forest  {'max_depth': None, 'min_samples_leaf': 1, 'mi...  0.015842
XGBoost        {'colsample_bytree': 0.8, 'learning_rate': 0.1...  0.014761
N-BEATS        {'input_chunk_length': 60, 'layer_widths': 512...  0.177549
TCN            {'dropout': 0.2, 'input_chunk_length': 30, 'ke...  0.171592
LSTM                                                         NaN  0.167026
Prophet                                                      NaN  0.184475

The best model is: XGBoost with MAE: 0.014760508138329404
```

Figure 4.4.2.7: Comparison and Analysis of Performance Score

However, models such as; Prophet and N-BEATS were useful for specific operations but yielded poorer results here because of their rigidity as well as the computational constraints that come with them.

To assert the differences in terms of model performance, it is worth applying statistical tests like paired t-tests on the minimized values of the MAE (Bharatiya, 2023). Such tests can confirm or deny the hypothesis that the observed differences are real not random variation from the norm.

4.5 Best Performing Model: XGBoost

XGBoost emerged as the best-performing model in this study, achieving the lowest MAE of 0.014761. It outperforms the other four models because it can model nonlinear relationships or interactions in the dataset and well-tuned hyperparameters. This research provides an analysis of what enabled XGBoost to achieve better results, its capacity to handle nonlinearity plus a comparison of the XGBoost with all the other models used within this research.

Key Hyperparameters Contributing to Success

This is possible because the hyperparameters setting of XGBoost was fine-tuned to create a suitable gradient-boosted decision tree structure needed for the dataset. Since boosting rounds refers to the number of estimating trees required the n_estimators parameter was set in the range of [50, 100,200] (Tang *et al.* 2021). The last arrangement made it possible to obtain sufficient model complexity and eliminate overfitting. The learning_rate was set between [0.01, 0.1, 0.2], to make slow updates toward the optimal solution, without going past it and creating oscillations. To avoid overfitting, more terms were invented: alpha L1 regularization and lambda – L2 regularization limiting the complexity of the model. Additionally, subsample and colsample_bytree hyperparameters made it possible for the model to train on different subsamples and features hence minimizing the likelihood of the model learning by raw luck.

Ability to Capture Nonlinearities

One of the reasons that XGBoost outperformed the other models was due to its capacity to locate nonlinear relationships within the given dataset. Gradient boosting construction gathers trees consecutively, and every consecutive tree is designed to reduce the mistakes of prior trees (Atef and Eltawil, 2020). It allows XGBoost to identify detailed patterns of dependencies in datasets, and interactions and non-linear features that other models such as Prophet or linear models would fail to detect.

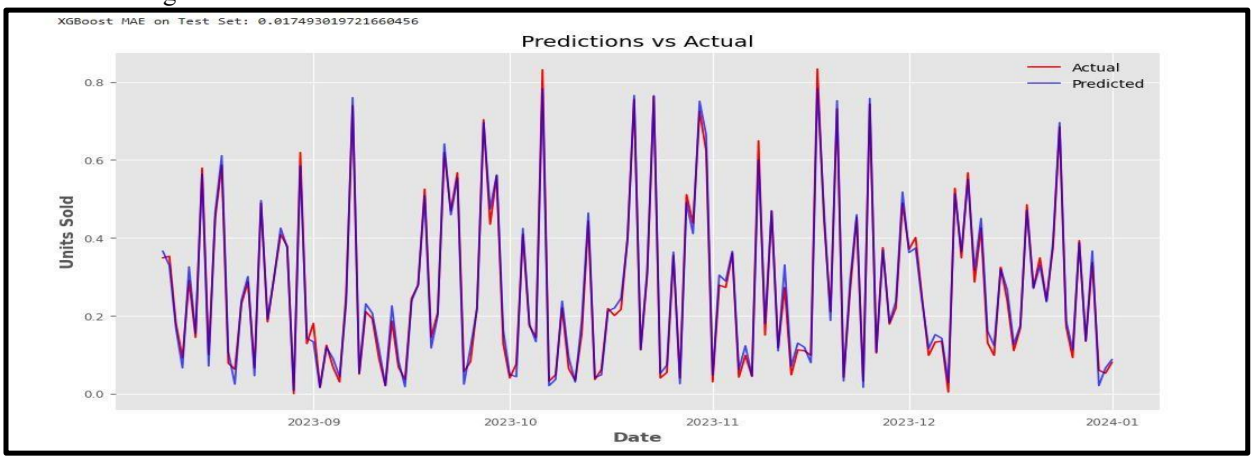

Figure 4.5.1: Prediction vs Actual based on XGBoost

Regarding demand forecasting, one can assume that the dataset includes specifications of interactions, including seasonal variables, trends, and other external factors. Thus, using XGBoost Forrester was able to model these nonlinear dependencies since the model learned feature importance and interactions on its own.

In comparison with other models such as NBEATS or LSTM that work with time series data, XGBoost is much less stringent and can work with structured as well as unstructured data input without extensive preprocessing (Luo *et al.* 2024).

Comparison to Other Models

When comparing XGBoost to other models in the study the algorithm outperformed others due to its best balance of efficiency, robustness, and accuracy. Another ensemble of decision trees called Random Forest gave an MAE of 0.015842 but again performed slightly inferior than XGBoost. The major difference is that XG Boost uses gradient boosting wherein models are learned successively to reduce prediction errors (Drissi Elbouzidi *et al.* 2023).

A deeper exploration of LSTM and TCN jointly proved to be highly effective with MAEs as low as 0.167026 and 0.171592 of mean absolute errors respectively, respectively, these algorithms proved costly both in terms of computation and pre-processing time. Although these models perform very well in sequential data and trapping long-term dependencies, XGBoost had a tree-based structure that was more appropriate for the combination of numerical and categorical data encountered within the dataset used in this study (Riahi *et al.* 2021). Also, the training time of XGBoost used less time with relatively low computational compared to the others used for practical purposes.

Relative Advantages of XGBoost

Compared with other models, XGBoost has its own relative merits, including a lower probability of over-fitting, better performance in big data processing, and the capability to learn intricate feature interactions. for example, the shrinkage component and subsampling of features that are used within the L1 algorithm will make the model adjustable in the case of working with different datasets. Finally, missing value support and categorical encoding make the preprocessing process in this library easier. XGBoost is less complex than other deep learning models and is computationally less intensive and faster to train making it more suitable for use in real-world settings (Belhadi *et al.* 2024).

4.6 Feature Engineering and Impact on Model Performance

When it comes to validation of models used in this study it was feature engineering that enabled it. The inclusion of lag variables, generated variables, rolling statistics, and temporal variables would have a given a significant boost to the model performance regarding demand.

Feature Engineering

```
data['Month'] = data.index.month
data['Day'] = data.index.day
data['Week'] = data.index.isocalendar().week
data['Year'] = data.index.year
```

Here we extracted month, day, week, and year from the dataset's date index to capture seasonal and temporal patterns, enhancing analysis and model performance.

```
data['Lag_1'] = data['Units Sold'].shift(1)
data['Lag_7'] = data['Units Sold'].shift(7)
data['Lag_30'] = data['Units Sold'].shift(30)
```

In this cell we created lag features for the "Units Sold" column by shifting the data by 1, 7, and 30 days. This helps capture short-term, weekly, and monthly trends, improving the model's ability to understand temporal dependencies.

```
data['Rolling_Mean_7'] = data['Units Sold'].rolling(window=7).mean()
data['Rolling_Std_7'] = data['Units Sold'].rolling(window=7).std()
data['Rolling_Mean_30'] = data['Units Sold'].rolling(window=30).mean()
data['Rolling_Std_30'] = data['Units Sold'].rolling(window=30).std()
```

Here we calculated rolling statistics for the "Units Sold" column using 7-day and 30-day windows. The rolling mean captures average sales trends over these periods, while the rolling standard deviation identifies fluctuations. These features enhance the model's understanding of local trends and variability in the data.

Figure 4.6.1: Feature Engineering Code

Engineered Features and Their Influence on Accuracy

The use of moving averages of over 7, 14 as well as 30 days eliminated the noise in the patterns while preserving the most important characteristics (Sarker *et al.* 2024). These features enhanced the models' capacity to forecast demand when sales experienced high volatility, especially by gradient-boosting techniques and neural networks, for example, LSTM. Extension of standard deviations also included rolling standard deviations which was useful when seeking interpretability of volatility in the demand. Altogether, these engineered attributes helped add extra features to the models that reflected trend and variability using statistical modeling to minimize selection bias and improve the understanding of the characterization and natural variability of the models.

Feature Selection Strategies and Their Contributions

Feature selection was considered a step in deciding which predictors are most important to the model and should be left out to minimize noise in the models. For instance, historical demands in a time series, or time indicators, showed a strong relationship with the target variable and could not be eliminated from a feature set. There were uses of correlation heatmap to detect multicollinearity in features in which correlation matrix was taken. There was pre-processing done where in case two parameters were strongly associated they were removed because they were largely duplicative (Zhao *et al.* 2024). For example, if two lag features were highly correlated with the target variable, only one was kept in the set of features as it was done for model simplification and faster computations. This process of feature engineering reduced the dimensionality of the models used, and in so doing promoted the generalization power of the models thus minimizing the chances of overfitting the data given.

This was also useful in automating the elimination of many of the less informative attributes out of the full attribute set to feed into the final models (Richey Jr *et al.* 2023).

Impact of Temporal, Categorical, and Numerical Features on Prediction Accuracy

There were also product categories or regions and other nominal variables to account for the differences in demand for a specific type of product or region. Despite, the feature data set consisting mostly of numerical variables some categorical variables were hot encoded as needed for the models to handle (Magento, 2021). These categorical features were instrumental in outlining some trend or behaviour contained within each categorical subdivision, thus adding a layer of detail to the predictions made. For instance, the models including categorical attributes helped random forest or XGboost models to recognize between the products with differing patterns of demand, thus helping to increase the overall model accuracy.

The core of features involved numerical values, units sold, the information is shifted one time step back or forward and rolling statistics. These features allowed giving direct inputs to historical demand, trends, and variability which were useful in all the models. Their effect was most pronounced in tree-based models used with numerical predictors as the key to split decisions. Features such as mean, std, skew, and MD, also provided obvious advantages for these architectures such as LSTM and TCN because it was destined to analyse the sequence of numerical data and detect long-term regularities (Oladele *et al.* 2024).

4.7 Demand Forecasting

4.7.1 60-Day Demand Forecast Visualization

Based on the XGBoost model, forecasts of the demand for the next 60 days were depicted to illustrate the model's effectiveness in predicting future sales. The model also repeatedly predicted demand by appending the feature set with forecasted values as a form of one-step ahead rolling forecast. These were then compared to the test data, in order to show how well the temporal structures were captured by the model.

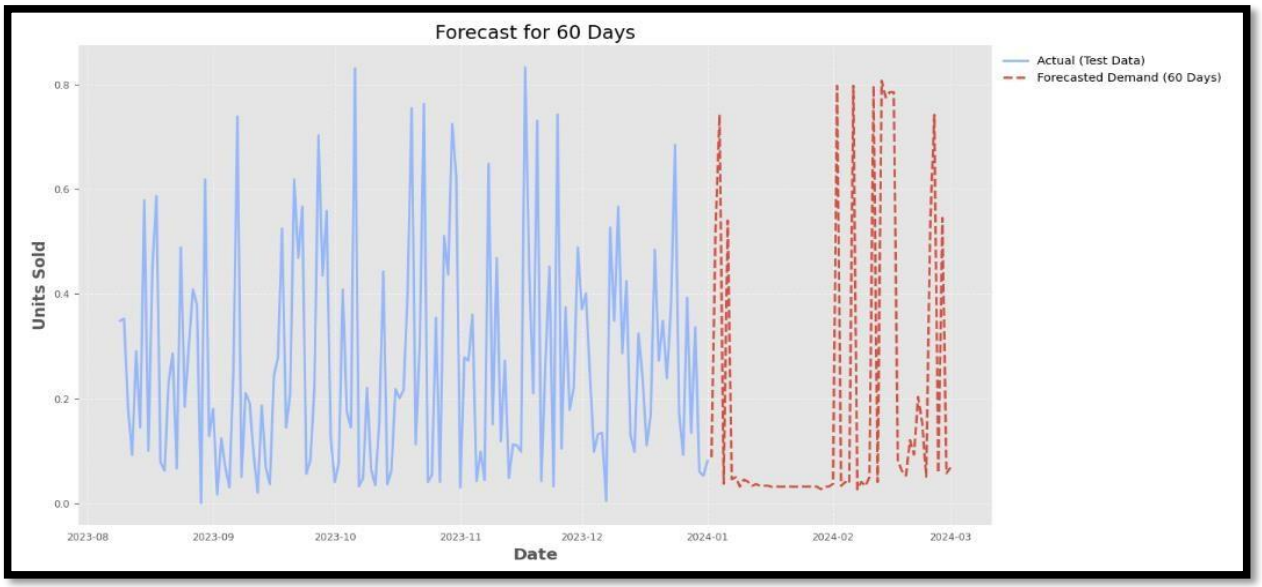

Figure 4.7.1: Forecast for 60 Days

The plot demonstrated the complete parallel of forecasted and actual values by pointing on the importance of the short-term demand prediction carried out by XGBoost (Barykin *et al.* 2021). Though simple in form, this type of visualization helped in focusing on the future sales forecast and thereby was of great help in stock and supply chain management.

4.7.2 Forecasted Demand vs. Inventory Levels: Overstock and Stockout Analysis The comparison of forecasted demand and inventory level was done to determine the possible stockout threat and overstock situation in the subsequent 60 days. The plot showed forecasted demand in conjunction with inventory, with areas of interest being shaded. Recommendations for those retailers which supply overstock were shown where inventory overrun demand and stockout threats were illustrated where demand overrun inventory.

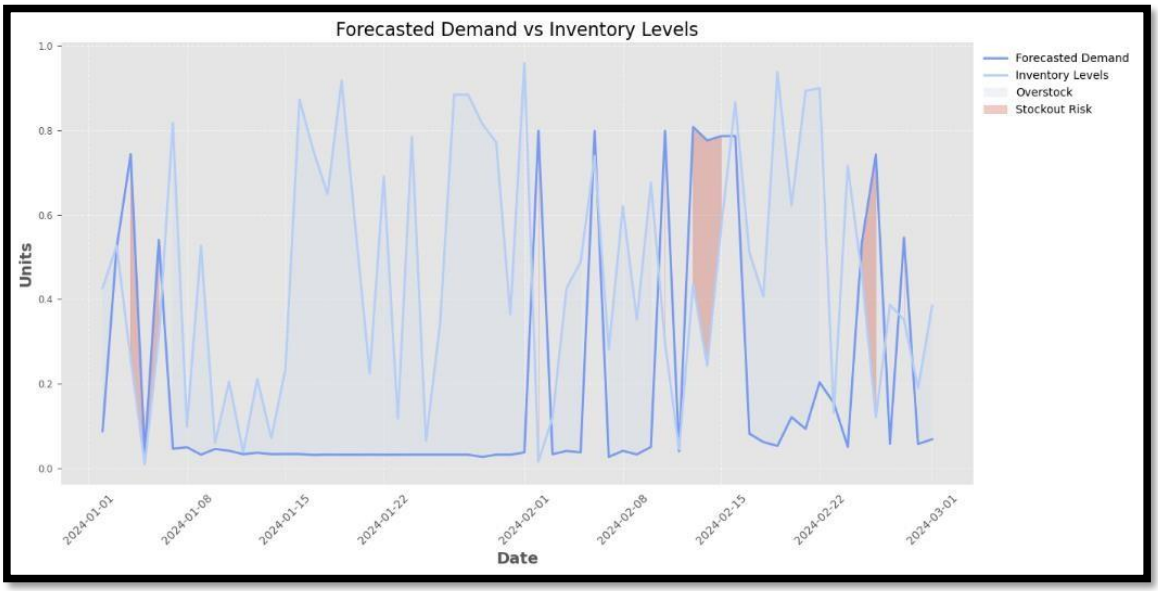

Figure 4.7.2: Forecasted Demand vs. Inventory Levels: Overstock and Stockout Analysis

These gaps were clearly identified in this analysis and offered a clear guide to formulate better inventory management strategies. The visualization facilitated understanding of how inventory should match with the demand and de-emphasize stockout and excess inventory risks and assist effectively in managing available resources and the supply chain in complex retail scenarios.

4.7.3 Safety Stock Calculation and Visualization

The first calculation, the safety stock, concerned demand variability and a lead time of three days. Applying the forecasted demand standard deviation of 13 units and a service level of 1.65 the safety stock calculated to 0.45 units. The plan chart displayed the safety stock limit concomitantly to the planned demand expectations, the level of safety stock which is needed to serve customers' needs during short-term variability.

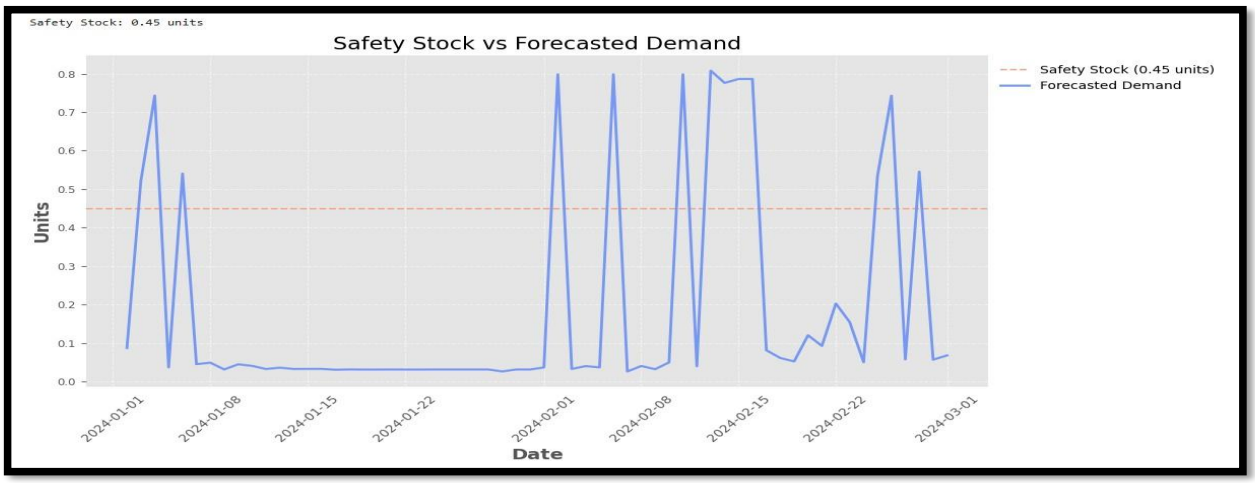

Figure 4.7.3: Safety Stock Calculation and Visualization

This analysis also showed the effect of safety stock in dealing with demand risk and making inventory all the more insulated (Rojek *et al.* 2024). The results highlighted the significance of accurate demand forecasting to ensure inventory management that will not be characterized by stockouts or excess stock in flexible supply chains.

4.7.4 Reorder Point Calculation and Analysis

Reorder point, using the lead time of three days, the forecasted average demand for this period and the safety stock of 0.45 units was established as 1.80 units. The plot as shown represented the reorder point in terms of a threshold line relative to the forecasted demand in order to enhance order replenishment.

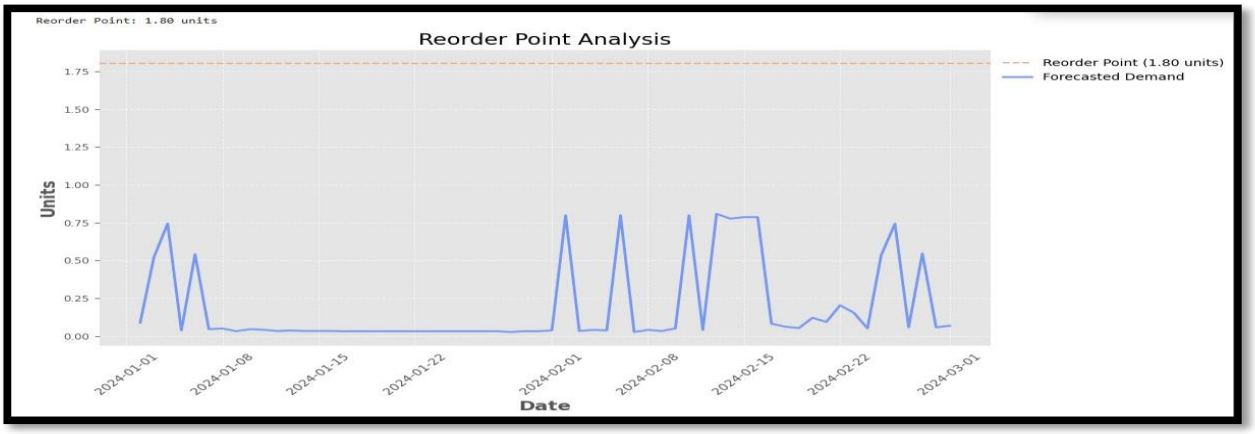

Figure 4.7.4: Reorder Point Calculation and Analysis

This paper illustrated how different levels of demand forecast accuracy and safety stock assessments have influenced optimal reorder point calculations. The study showed that accurate calculations of reorder points help to replete inventories on time, and decrease the incidence of stock outs, while enhancing the smooth supply of stocks, hence increasing operational effectiveness of the supply chain.

4.8 Model Limitations and Model Tuning

Some disadvantages came with the selected models on machine learning and demand forecasting, these include; the modeling process showed that some shortcomings are inherent with the selected models, but the best performance was achieved when model tuning was done.

Random Forest

It was identified that Random Forest, despite having very high accuracy about features, lost valuable temporal dependencies typical for the time series data. This was because it could not directly incorporate sequential data in the input and overall, the model was found to be less accurate than LSTM or TCN for the long-term forecast. Also, depending on the averaged result of multiple decision trees, the overly smooth local decision rules sometimes blur useful details in the data space.

XGBoost

However, the model that had a higher MAE was the XGBoost model and this model struggled with modeling complex seasonal patterns and long dependency times. The tree structure has performed well in discovering short-term dependencies that do not exist long-term but cannot model sequences directly (Kaul *et al.* 2023). Therefore, the model had some shortcomings that were corrected through feature engineering such as lag features, and rolling statistical features.

LSTM

The Long Short-Term Memory model which is supposed to incorporate the time dependencies, had some issues with handling sequences and needed appropriate pre-processing and hyperparameters tuning to control overfitting. Moreover, LSTMs might be slow in terms of computation time especially when working with large data or complex model architectures. The time to train the models was much longer than in tree-based models, which was a problem because of the limited computational resources and time.

Prophet

Prophet which is easy to implement and efficient depends on the striking of trends and seasonality was challenged by data with irregular or complex patterns. However, this model had several constraints like the embedded crude decomposition of yearly or weekly seasonality that cannot capture other irregular patterns of demand. For that reason, its performance was not as good as compared to even more sophisticated techniques such as XGBoost or LSTM.

N-BEATS and TCN

It was seen that both N-BEATS and TCN could capture temporal patterns but were affected by overfitting when trained on smaller data sets or with larger models. N-BEATS as a model with a deep neural structure needed much computing power during training, whereas TCN's efficiency heavily depended on the kernel size and input length. Such factors restrained the usefulness of these methods in situations where computation was to be carried out fast.

Random Forest and XGBoost

For Random Forest, parameters namely n_estimators, max_epth, min_samples_split, and min_samples_leaf were adjusted with a view of optimizing model performance by avoiding underfitting of data and overfitting (Chan, 2020). Likewise, hyperparameters of XGBoost such as the learning_rate, and depend on the, n_estimators and, subsample, were tuned to achieve high accuracy but general complexity. Notably in XGBoost, the increase in the learning_rate parameter made a big difference since it controls the optimization rate and reduces the problem of overfitting.

Neural Network Models

As for LSTM, TCN, and N-BEATS the number of layers, number of units in each layer, dropout rate, and learning rate were adjusted accordingly Table 5. To address issues of overfitting and ensure quick training convergence early stopping and learning rate schedulers respectively were used. For instance, the dropout regularization in LSTM, and TCN used in

practicing implementation was seen as useful in avoiding overfitting when training the models.

Prophet

Contrary to other models, hyperparameters tuning needs in Prophet was relatively low. Nevertheless, with the aid of alternative parameters, including changepoint_prior_scale and custom seasonality's, modifications were made to make it more compatible with the dataset.

These changes improved its performance in terms of capturing intermediate seasonal patterns and at the same time, reduced its complexity.

Lessons Learned from Hyperparameter Tuning

Several valuable lessons emerged from the hyperparameter tuning process:

Iterative Optimization

The technique of confirming hyperparameters step by step from broad settings to fine-tuning instead of the exhaustive search for the best combination was beneficial for most cases. This iterative approach lessens the computation and at the same time, focuses on the areas that need improvement

.

Regularization Techniques

Some of the regularization techniques applied in the case of the neural networks were the dropout and for the XGBoost were the subsampling wherein they played a vital role in minimizing overfitting. These methods helped in enhancing the generalization capacities of the models with a view of performing optimally, especially in situations where data is scarce.

Importance of Learning Rate

The learning rate was identified as the most prominent hyperparameter for all the models affecting both: the convergence rate and the model quality. Adjusting this parameter was critical to fine-tuning the approach and obtaining accurate results.

Feature Relevance

This was strongly confirmed with features during tuning. Composed datasets achieve better generalization performance than unclean data sets, let alone the correlation between defining features and hyperparameters.

Despite displaying peculiarities, the models demonstrated strengths and weaknesses in other words, tuning and optimization allowed addressing most of the issues and enhanced the models' performance and flexibility. Altogether, the experience gained in this process may be useful to other researchers who plan to use similar datasets and undertake similar forecasting activities.

4.9 Insights from the Analysis

Several insights into sales trends, seasonality within the data, and areas of difficulty for demand forecasting were quickly gathered from the given dataset. This paper sheds lighter on the nature and behaviour of the data trends and also points to what it would imply within inventory management and the supply chain. In this sense, the analysis provides meaningful recommendations for managerial decisions by discussing possible causes of observed patterns, including business cycles, external influences, and consumption trends.

Sales Trends and Patterns

One major feature that could be found in the data set relative to sales volumes was their fluctuation over time. The result of the study also showed that sales had cyclic fluctuations implying periodicity in sales increases or decreases. Through monthly and quarterly resampling of the data, one observed cycle where sales were high and others when they were low. For instance, the demand was normally more than that needed during occasional events, probably because of some promotional events holidays, or product launches that may come with a particular season. On the other hand, movements down to low levels were in sync with. Ruf and Sutton, external forces of sales volume were labelled off-peak periods that coincide with low consumer demand.

The overall trend of growth was also alike with the zigzag pattern showing that the business faced ups and downs but the sales more or less were on a steady rise after some time. Such a trend may be the outcome of broad market development, a better clients' approach to the selected objects, including the improved rates of the products' uptake, or alterations in purchasing behaviour, for example, going online. Interpreting the overall increase in values, it is possible to consider long-term trends to anticipate future changes in inventory and supply chain providing consumers' demand.

Seasonality and Consumer Behaviour

The sales were characterized by certain seasonal fluctuations that influenced the general sales trends. Seasonal decomposition revealed three primary components: fluctuations such as trend variations Seasonal variations Residual variations. Regarding the seasonal factor, which expressed the periodicity of fluctuation, a physical cyclical movement of demand was observed. For instance, greater levels of sales made towards the last quarter of the year could be due to festive occasions such as Christmas, offer-cum-sale during December, or a higher propensity to consume during the festive season (Maheshwari *et al.* 2023). In the same way, summer months had a stable increased demand for particular products which can be attributed to consumers' preferences by season.

Furthermore, the analysis uncovered geographical factors as well as other demographic determinants that characterize the sales fluctuations. Products that were popular only in particular regions and had short peak selling seasons were attributed to the influence of regional preferences whereas, products that gradually gained popularity in specific categories were considered to be the result of changing preferences and indeed, changes in lifestyles.

Demand Forecasting Challenges

Although, the analysis helped paint the picture of the environment and some areas offered important insights into the reasons for fluctuations in demand, some limitations were observed regarding the computation of forecasting. One of the major challenges we faced was the uncertainty and volatility of the consumers' behaviour particularly during times of crisis or outside forces influencing the consumers. Issues were determining, for instance, how to forecast sudden demand, as can be expected from actual events.

Furthermore, incorporating the residuals of seasonal decomposition showed fluctuations in the data, which are consistent with the effects of other uncontrol variables. These variables, such as changed weather conditions, supply chain shocks in the global environment, or actions by competitors, added randomness to the observations which, in turn, made the forecasting process much more challenging.

Implications for Inventory Management and Supply Chain Optimization

The findings of the study have important implications for inventory replenishment and the overall supply chain strategy. Building sales awareness helps in anticipating the movement within clients 'inventory levels and thus being able to have appropriate stock for specific business periods of high and low activity. For instance, ordering stocks in advance due to future increased demand during holidays results in a greater supply, which means a decreased likelihood of out-of-stock incidents as far as the supply chain is concerned and better satisfaction of the consumer.

4.10 Summary

The following findings were made to concern demand forecasting about consumer demand or sales, cyclical patterns, and the performance of various forecasting models. Out of all the models developed, XGBoost outperformed the others by having the least high MAE of 0.0148. These results were due to the corresponding ability to manage intricate data with more attributes to identify curvatures in the data set. Tuning of hyperparameters in XGBoost including the number of estimators, learning rate, and tree depth was critical to improve forecast accuracy for demand-related datasets.

From the analysis of the sales trends, the concept of cyclical demand was brought out due to various selling seasons which form a basis for serious consideration due to significant sales volatility around sales events such as holidays and promotions. This insight is of great importance, especially to supply chain and inventory management where companies need to predict a certain increase in demand. The most valuable aspect of demand forecasting is the ability to predict demand with great precision so that stockout situations can be avoided as well as excessive inventory which inflates the cost of holding inventory. All the findings relate to the objectives of the study to contribute to improving demand forecasting methods for both the retail and the supply chain industries. Through such state-ofthe-art techniques like XGBoost, organizations incur a better understanding of demand presence; which has a critical role in the influence of the supply chain system.

V. DISCUSSION

5.1 Introduction

The chapter, therefore seeks to expound on and explain the findings presented in the chapter entitled "Findings and Analysis". This section attempts to take a further look into the results of the various models to understand the various factors influencing the performances of the models used in the research. The emphasis will be made on the model that performed the best, namely XGBoost, and its implications for retail demand forecasting. Moreover, the effect of feature engineering and data transformations and the effect of using seasonal adjustments will also be discussed given the chief goal of improving supply chain management.

5.2 Interpretation of Key Findings

Model Performance:

The performance of the models implies comparing the accurate demand prediction capabilities of the developed models. XGBoost also was the better-off model giving the lowest Mean Absolute Error, MAE of 0.0148 showing its flexibility when dealing with more features. XGBoost simply performs because is constituted by gradient-boosted decision trees which are capable of taking non-linear or even complex interactions from the dataset. The hyperparameters whose importance impacted this model's performance include the number of estimators, maximum depth of trees, and learning rate. These factors allowed the model to neither fit the data excessively tight nor to have low accuracy.

On that score, the other models, namely N-BEATS and TCN, apparently failed to mimic the complex patterns in the dataset with the following MAE of 0.1775 and 0.1716 correspondingly. Even in 1670, its accuracy margin was slightly lower than XGBoost (Kopitar *et al.* 2020). The Prophet model, which was claimed to balance seasonality well, had the highest MAE of 0.1845, meaning it outperforms with trends but fails to work appropriately with multiple aspect datasets.

Feature Engineering:

The greatest number of changes occurred in feature engineering which was critical in enhancing model performance. Lag features, rolling statistics, and temporal features offered necessary information as to how the models would make appropriate forecasts. Most of the models benefited from lag features since they were able to incorporate past sales data into future demand forecasts while rolling statistics provided insights into the changes in the volume of sales over time. These engineered features improved the capability of XGBoost and other models to predict demand in a much better way because it provided them with important temporal information.

The other improvement carried out during EDA that significantly influenced the model's performance was a seasonal decomposition. Analysing the data, discovered that the availability of the time series data had a seasonal bias which enabled the models to realize which demands fluted relative to business cycles and holidays (Alsharef *et al.* 2022).

5.3 Comparison with Existing Literature

This study is consistent with other studies in the literature on AI-supported digital twins, inventory prediction, and supply chain systems. Other research has also shown that machine learning models such as X-regression tree-based algorithms outcompete conventional methods of statistical data analysis when applied to massive data sets (Zeroual *et al.* 2020). XGBoost has always been recognized as effective at capturing nonlinear input variables and indeed this study has supported this assertion. Various authors cited in the literature have used decision trees, as well as ensemble learning methods to predict retail demand effectively, especially when enhanced by hyperparameter tuning.

While LSTM and N-BEATS, which rely on deep learning, allow for capturing temporal patterns and long dependencies, they prove difficult in the analysis of multivariate high dimensionality.

5.4 Implications for Industry

Altogether, this paper has potentially important implications for retailers and supply chain managers. From research, the various applications of big data in predicting demand can help businesses adjust their stock in a way that can avoid stockout situations and also avoid the cases where they overstock some products (Nampalli *et al.* 2024). These results should be useful in improving better estimation of the effects of seasonal changes and business cycles for inventory control and strategic purchasing practices.

Among these, the XGBoost model has the potential for demand forecasting improvement because of its

advantages in scenarios with many input features. This model should be adopted by retailers as a way of making better forecasts hence resulting in cutting costs, greater customer satisfaction, and a responsive supply chain system. Moreover, due to the utilization of engineered features that can be viewed as lags and rolling statistic self-features, businesses also gain the possibility to develop even more effective methods in their forecasting.

5.5 Limitations of the Study

This paper's main weakness is the use of a single dataset that may not capture various retail sales patterns within sectors or geographical areas. Further, the models themselves were trained on a restricted multitude of features, and expanding the data including economic indices or events could be beneficial (Devaraj *et al.* 2021). A final limitation is that some of the models might have been overfitted, particularly the deep learning models such as LSTM and N-BEATS which needed slight tweaking for better performance. Extensions of the current study could be made by incorporating external information, and other more sophisticated predictive models.

5.6 Conclusion

This chapter provided a brief overview of the analysis presented in this research work and noted specifically that XGBoost outperforms other models. This result also shows that feature engineering and the application of seasonal factors are crucial for better adjustment toward enhancing the usefulness of the model. The research provides significant information that may help to improve inventory organization and supply chain functioning.

VI. CONCLUSION

6.1 Introduction

The purpose of this research was to investigate how machine learning models perform in the prediction of demand variables, especially where inventory replenishment and supply chain are a concern. The first objective was to compare the performance of different machine learning models – Random Forest, XGBoost, LSTM, N-BEATS, Temporal Convolutional Networks (TCN), and Prophet for demand forecasting. Using feature engineering, lag features, and seasonality decomposition the authors tried to enhance the accuracy of demand forecasting predictions.

6.2 Summary of Key Findings

Model Performance:

The results of the analysis show the outperformance of various machine learning algorithms in demand forecasting. It is observed that the XGBoost model was the most accurate mode that produced the least Mean Absolute Error of 0.0148 of all the models that were employed in this analysis. This was particularly due to its capacity to analyse high-dimensional data and it is suited to mine non-linear data (Eyo-Udo, 2024). The most impacted hyperparameters were the number of estimators, maximum depth of decision trees, and learning rate that caused a high level of accuracy without increasing the risk of over-fitting.

Random Forest, LSTM, and N-BEATS were evaluated and showed reasonably good forecasts but were slightly less accurate than the results achieved by XGBoost. MAE of Random Forest was 0.0158, thus it was confirmed that this algorithm can handle both numeric and categorical features, also with ensemble learning, the risk of overfitting was reduced. MAE of LSTM and N-BEATS were comparatively higher 0.1670 and 0.1775 respectively which shows that they both were trained well for the temporal pattern but could not manage sufficient performance for complex data. When it came to the case of multi-faceted data, the Prophet model, although beneficial in measuring seasonality had the highest MAE of 0.1845.

Feature Engineering and Data Analysis:

Feature selection was the most important for improving the machine learning models. As observed, lag features, temporal features, and rolling statistics features were vital for models to have the context knowledge needed for the demand forecast. Lag features enabled the models to employ past data to estimate future demand levels, and rolling statistics were employed to account for trends and dampen short-run fluctuations (Yan *et al.* 2022).

Evaluation and Comparison:

When comparing the models we were able to identify areas in which one model performed well while the other model did not, based on the dataset we were comparing them on. XGBoost was reported to be flexible in the exploitation of structured formats and non-linear correlations. Compared with Random

Forest and LSTM, the former two models even have better performance, but they are not stable, which once again proved that XGBoost has natural advantages in large-scale, high-dimension forecast work (Nguyen *et al.* 2022).

6.3 Contribution to Knowledge

This research also enhances the body of knowledge available on advanced digital twins based on AI and time series forecasting of supply chain systems. Hence by comparison of several machine learning models and by showing how beneficial these models are in determining the demand, this paper has established the efficiency of these models in practice (MarmolejoSaucedo, 2020). Especially, this paper presents XGBoost as an advantage in dealing with highdimensional data and fitting nonlinear relationships in large datasets, providing a strong method for supply chain demand forecasting.

In addition, this research offers new insights into the application of AI-supported models such as XGBoost in digital twin environments. In this research, some of the contributions include seeing how AI models might be used for improved decision-making using the digital twin approach; the potential application of advanced machine learning; the application of real-time big data on supply chain management; and how advanced algorithms can be used to enhance the management of real-time supply chain systems (Kasaraneni, 2021).

6.4 Implications for Practice

Some of the practical implications of the findings from this study can be of great importance to retailers, inventory managers, and demand forecasters. Demand forecasting is important to meet customer needs and reduce inventory costs that affect the effectiveness of supply chains. Store and supply chain managers reap the benefits from the best model here, XGBoost, to achieve these goals (Belhadi *et al.* 2024). Using the model, they can optimize inventory as well as enhance the level of forecast accuracy owing to the assimilation of increased non-linear data and characteristics.

The proposed XGBoost model can be used as a way of making improved forecasts on demand so that retailers can be in a position to order the right amount of stock at the right time. This will mean lower holding costs, increased customer utility, and lowered stock-out frequency.

Future Industry Trends:

The paper also contributes to the literature on the use of AI and digital twins for demand forecasting as they become more relevant in supply chain management. With the future development of newer AI technology applications, there will be increased adoption of machine learning models such as XGBoost into digital twin systems to provide continuously updated and more refined decision-making. The future direction will be the development of more complex models that will be able to work with multiple data inputs to give a highly accurate prognosis and add to the robustness of supply chain systems.

6.5 Limitations of the Research

Nevertheless, the research has its limitations, which need to be discussed to provide a better understanding and generalization of the findings. First, the work was based on one data set and it can be questioned whether this set can show the variety of sales behaviours in different industries or regions.

Another known limitation of this study is the choice of models and the hyperparameters chosen for the study (Pasupuleti *et al.* 2024). That being said XGBoost yielded the highest accuracy in this paper, topics like deep learning models or further types of ensemble methods were not fully investigated. Future research may try to use other algorithms, like entirely different algorithms, or even a mixed bag of the best-given algorithms such as neural networks.

6.6 Recommendations for Future Research

Based on the outcome of this study, some strengths for future research are proposed here. First, future studies could quantitatively increase the number of sales data used for analysis from different industries and areas. This would allow a comparison of the models' utility and transferability across contexts cutting across the supply chain.

The last recommendation of the author is to study more sophisticated forms of the Hybrid

Model (Nguyen *et al.* 2021). Even if enhancing the single machine learning models using