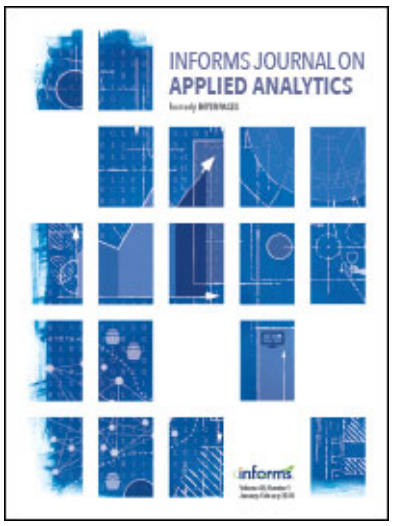


INFORMS Journal on Applied Analytics

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Faster, Smarter, Leaner: How Flipkart Optimized Its Supply Chain to Unlock Growth

Shubham Agarwal, Prateek Agrawal, Anurag Allamsetty, Adarsh Attavar, Deekshith B, Gowtham Bellala, Shobhit Bhatnagar, Hardik Choudhari, Vikas Goel, Praveen Gupta, Ananth Kachroo, Jay Kothadiya, Nagesh KM, Sai Anjani Kumar Kudupudi, Mayank Kumar, Naidu KVM, Tanu Modi, Ramkumar Moorthy, Rakesh S. Nair, Goutham Sai Panyam, Avijit Shukla, Piyush Vyas

To cite this article:
Shubham Agarwal, Prateek Agrawal, Anurag Allamsetty, Adarsh Attavar, Deekshith B, Gowtham Bellala, Shobhit Bhatnagar, Hardik Choudhari, Vikas Goel, Praveen Gupta, Ananth Kachroo, Jay Kothadiya, Nagesh KM, Sai Anjani Kumar Kudupudi, Mayank Kumar, Naidu KVM, Tanu Modi, Ramkumar Moorthy, Rakesh S. Nair, Goutham Sai Panyam, Avijit Shukla, Piyush Vyas (2026) Faster, Smarter, Leaner: How Flipkart Optimized Its Supply Chain to Unlock Growth. INFORMS Journal on Applied Analytics 56(1):42-57. https://doi.org/10.1287/inte.2025.0282

INFORMS JOURNAL ON APPLIED ANALYTICS
Vol. 56, No. 1, January–February 2026, pp. 42–57
ISSN 2644-0865 (print), ISSN 2644-0873 (online)
https://pubsonline.informs.org/journal/ijaa

THE FRANZ EDELMAN AWARD
Achievement in Operations Research

Faster, Smarter, Leaner: How Flipkart Optimized Its Supply Chain to Unlock Growth

Shubham Agarwal,[a] Prateek Agrawal,[a] Anurag Allamsetty,[a] Adarsh Attavar,[a] Deekshith B,[a] Gowtham Bellala,[a,*] Shobhit Bhatnagar,[a] Hardik Choudhari,[a] Vikas Goel,[a] Praveen Gupta,[a] Ananth Kachroo,[a] Jay Kothadiya,[a] Nagesh KM,[a] Sai Anjani Kumar Kudupudi,[a] Mayank Kumar,[a] Naidu KVM,[a] Tanu Modi,[a] Ramkumar Moorthy,[a] Rakesh S. Nair,[a] Goutham Sai Panyam,[a] Avijit Shukla,[a] Piyush Vyas[a]

[a] Flipkart Internet Private Limited, Bangalore 560103, India

*Corresponding author

Contact: shubham.agarwal@flipkart.com (SA); agrawal.prateek@flipkart.com (PA); allamsetty.anurag@flipkart.com (AlA); adarsh.attavar@flipkart.com (AdA); b.deekshith@flipkart.com (DB); b.gowtham@flipkart.com, https://orcid.org/0009-0006-7460-0190 (GB); shobhit.bhatnagar@flipkart.com (SB); hardik.choudhari@flipkart.com (HC); vikas.goel@flipkart.com (VG); gupta.praveen@flipkart.com (PG); ananth.kachroo@flipkart.com (AK); kothadiya.jay@flipkart.com (JK); nagesh.km@flipkart.com (NagK); sai.anjanikumar@flipkart.com (SAKK); mayank.k@flipkart.com (MK); naidu.kvm@flipkart.com (NaiK); tanu.modi@flipkart.com (TM); ramkumar.m@flipkart.com (RM); nair.rakesh@flipkart.com (RSN); goutham.sai@flipkart.com (GSP); avijit.shukla@flipkart.com (AS); piyush.vyas@flipkart.com (PV)

Accepted: October 15, 2025

https://doi.org/10.1287/inte.2025.0282

Abstract. The Flipkart Group, one of India's foremost digital commerce entities, serves more than 500 million registered users and offers a vast selection of more than 150 million products, connecting customers with 1.4 million sellers. To keep pace with rapid growth and the evolving ecosystem of e-commerce in India, Flipkart embarked on a transformational overhaul of its supply chain planning technology in 2021. This transformation led to the development of an advanced, fully integrated supply chain planning platform, built on machine learning and operations research techniques. The platform comprises two core layers: forecasting and optimization. The forecasting layer leverages a suite of statistical and machine learning techniques to produce multilevel demand forecasts. The optimization layer converts forecasts into actionable decisions across three key domains: inventory management, capacity planning, and network flow planning. These decisions collectively maximize delivery speed and reliability, minimizing operational costs. Flipkart has scaled this platform to automate and optimize end-to-end supply chain operations. Its impact has been profound: leading to a 10% increase in manpower utilization, a 50% reduction in unhealthy inventory, and a 50% increase in one-day deliveries.

Keywords: supply chain optimization • capacity planning • inventory planning • network flow planning • machine learning • e-commerce • Edelman award

Introduction

The Flipkart Group, which includes Flipkart, Myntra, Flipkart Wholesale, Cleartrip, and super.money, is one of India's leading digital commerce entities. Flipkart serves more than 500 million registered users and offers a vast selection of more than 150 million products, connecting customers to 1.4 million sellers. The company delivers products, including mobile phones; electronics; lifestyle products; books; home and general merchandise; groceries; perishable products; and large household appliances, such as refrigerators, washing machines, and televisions. On a typical day, Flipkart delivers more than four million shipments to customers across approximately 19,000 zip codes covering the vast geography of India, ranging from densely populated cities to far-flung rural areas with limited accessibility. On peak days, the number of deliveries can exceed 10 million.

The goal of a supply chain is to deliver products reliably at the lowest cost and at the highest speed. This is challenging given the large scale of Flipkart's business. The challenges are amplified given the complexities of the dynamic ecosystem in which Flipkart operates. Our customer base is rapidly evolving and demanding more from us. The rise of quick commerce (i.e., a rapidly growing segment of e-commerce that focuses on delivering goods to customers in a short time frame, typically within 10 to 30 minutes) and online fast fashion are recent examples. Our customer base is also

highly diverse; one set of customers is willing to pay extra for quicker deliveries, whereas another set prefers to save money even if delivery takes a full week. A one-size-fits-all supply chain no longer works. In addition, although the Indian logistics sector has made massive strides in the past few years, it is still highly unorganized and fragmented with a high degree of variability in reliability and availability across the country. This imposes additional challenges to ensuring an efficient and robust supply chain.

Prior to 2020, our planning process was quasi-manual and loosely coupled. Specifically, planning teams were consumed with building monthly plans and had limited bandwidth for achieving efficiency and improvements. Teams often worked with different objectives, resulting in a lack of coordination, introducing inefficiencies into the entire process, and limiting our ability to respond effectively to market dynamics. As the diversity and size of Flipkart's business grew, this approach hampered the company's ability to keep up with the growing complexity and started to hinder business growth.

This problem was further exacerbated in 2020 with the outbreak of the COVID pandemic. A massive jump in e-commerce penetration, coupled with frequent and unpredictable supply chain disruptions and the challenges of team members coordinating and collaborating when working remotely, tested our supply chains to their limit.

Given the rapid growth and evolution in the Indian e-commerce market, Flipkart initiated an overhaul of its supply chain planning technology in 2021 by creating a central planning platform that could enable it to succeed in a highly dynamic business environment, serve a diverse customer base, and manage a complex vendor and logistics ecosystem. Built over a period of only three years, this platform combined the strengths of machine learning (ML) and operations research (OR) to optimize capacity, inventory, and network flow planning decisions across the entire Flipkart supply chain. This platform has transformed the Flipkart supply chain and delivered significant cost savings and incremental net sales since its implementation. In addition, this platform has enabled the Flipkart supply chain to become much more agile and efficient by enabling planning teams to quickly adapt to changing business conditions.

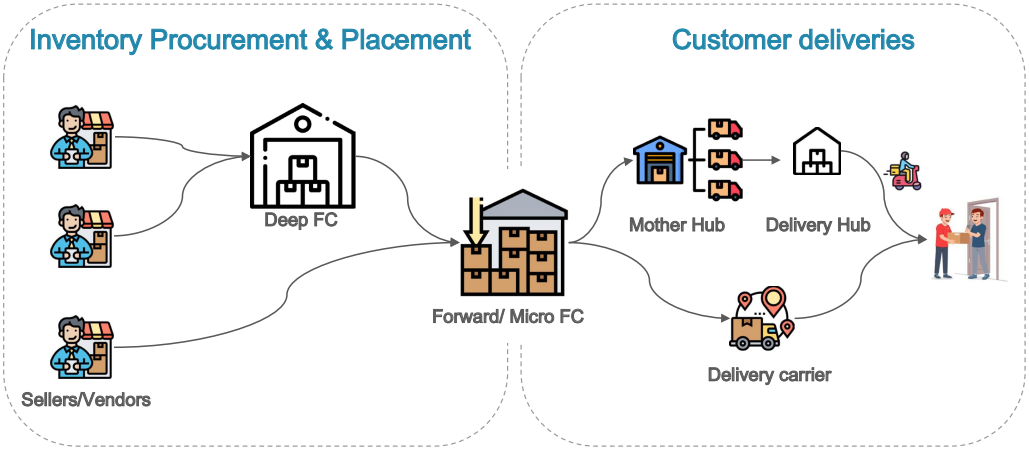
The Flipkart Supply Chain Planning Process

The Flipkart supply chain focuses on two major processes (see Figure 1): (i) inventory procurement and placement focuses on making the right products available as close to the customer as possible, and (ii) customer delivery focuses on ensuring that these products are delivered to the customer quickly at the lowest possible cost.

In the inventory procurement and placement process, inventory is procured from sellers across a multiple-tier network of fulfillment centers (FCs). FCs can range from massive centers, referred to as deep FCs, with millions of cubic feet of storage to 2,000 square feet forward/micro FCs located near residential neighborhoods. Different models of inventory procurement exist from direct procurement at FCs to moving inventory from one FC to another, all with the aim of making the right inventory available as close as possible to the customer.

The customer delivery process is a complex system that spans multiple journey nodes and hops. Shipments move from the FCs to mother hubs (MHs), which sort and aggregate them based on their destinations. We off-load some of the shipments to third-party

Figure 1. (Color online) Overview of Supply Chain Processes at Flipkart



Notes. At Flipkart, inventory is procured from sellers across a multiple-tier network of FCs that consists of large FCs referred to as deep FCs and smaller FCs referred to as forward/micro FCs. The customer delivery process is a complex system in which shipments start from an FC, flow through a maze of mother hubs, and finally reach a delivery hub, from which the delivery is made to the customer by a delivery agent on a bike or in a van. In some cases, the customer delivery process may be off-loaded to third-party delivery carriers.

delivery carriers, which deliver shipments from MHs directly to customers. The remaining shipments flow through a maze of our MHs and finally reach a delivery hub (DH), which is a small facility close to a customer location. Each of these thousands of DHs serve a cluster of zip codes. Delivery agents then pick up packages at the DHs and make the final deliveries to the customers on bikes or in vans. Each leg of this journey from the FC to the customer is carefully orchestrated and planned to ensure fast delivery at the lowest cost possible.

The goal of supply chain planning is to deliver shipments to customers at the fastest speed, at the lowest cost, and with the highest reliability. At its core, planning and executing these processes is a resource allocation problem (see Figure 2). For the inventory procurement and placement process, we need to help sellers allocate their inventory in the best way possible across locations and time windows (i.e., the optimal allocation that is closest to customer demand). Hence, under inventory planning, we focus on optimizing inventory quantity, placement, replenishment, and liquidation to achieve high inventory availability and delivery speeds to the customers, improving the inventory efficiency (i.e., inventory turnover ratios) for the sellers. We provide additional details in the inventory planning section.

Similarly for the customer delivery process, we need to allocate and plan for resources, such as personnel, trucks, and delivery agents, to meet our goals. The customer delivery process consists of two main processes: capacity planning and network flow planning. In capacity planning, the key decisions include the allocation of personnel and capacities at each node of the delivery network as well as planning of the logistics resources (allocation of the load between internal and third-party carriers for deliveries). These decisions have a significant impact on delivery speeds, costs, and overall customer satisfaction.

Network flow planning focuses on optimizing the network flow to improve the delivery speed and reduce the cost per shipment that the network can deliver. Key decisions include routing, sortation planning, and FC wave scheduling.

Inventory procurement and placement, capacity planning, and network flow planning utilize inputs of forward-looking demand forecasts from a demand planning process that runs at both a category level and a product level. In this process, a category-level demand plan (e.g., the number of earphones sold nationally on each day of the month) is created and aligned across all teams in Flipkart. This plan is shared with each individual category team, which creates a product-level demand plan at a regional level (e.g., number of Apple AirPods to be sold each day in a specific state in India). These category- and product-level demand plans serve as the basis for all downstream decision making, and improving the accuracy of these plans enhances the efficiency of our downstream resource-allocation decisions.

The Central Planning Platform

In 2021, Flipkart initiated an overhaul of its supply chain planning technology by creating a central planning platform (see Figure 3) that could enable its rapid and profitable growth. This platform has two fundamental layers. The forecasting layer predicts future demand using ML and deep learning techniques. This ensures that all our decisions are based on a robust estimation of future demand. The operational efficiency layer includes a set of optimization models that

Figure 2. (Color online) Overview of the Supply Chain Planning Processes at Flipkart

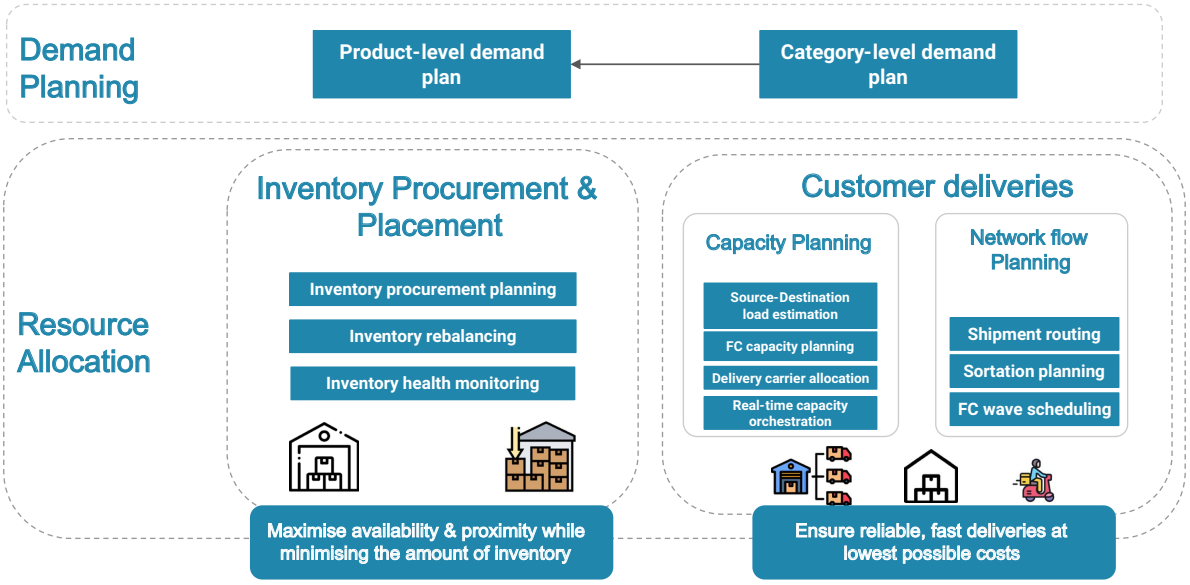
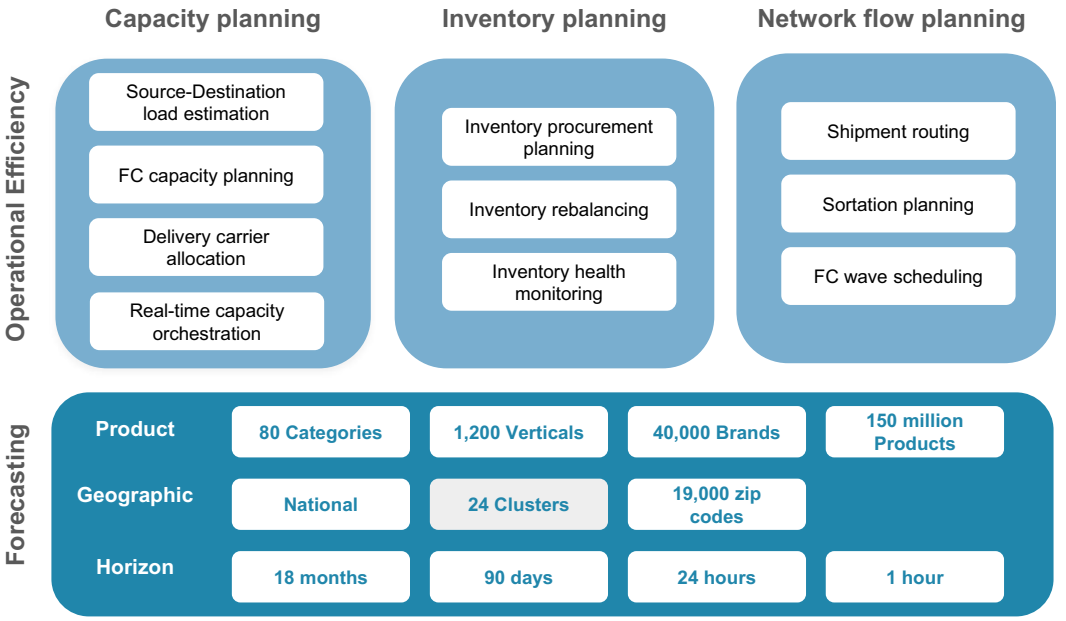


Figure 3. (Color online) Flipkart's Central Planning Platform



Notes. The platform has two fundamental layers. The forecasting layer predicts future demand using machine learning and deep learning techniques. This ensures that all our decisions are based on a robust estimation of future demand. The operational efficiency layer includes a set of optimization models that optimize decisions across the three core pillars of supply chain planning.

optimize decisions across the three core pillars of supply chain planning:

i. Capacity planning allocates resources to ensure that we have the right supply chain capacity.

ii. Inventory planning ensures that we have the right products in stock at FCs closest to the customers.

iii. Network flow planning involves the routing and scheduling of shipments for a fixed network design.

These models optimize decisions ranging from monthly planning to real-time decision making and, thus, lead to overall optimization of the supply chain. The central planning platform delivers significant cost reductions and incremental net sales by enabling automation, optimization, and integration.

As we mention above, before we built this platform, supply chain planning was a manual process performed in a loosely coupled manner. Different teams were optimizing for different objectives. This platform automates the planning process, making Flipkart more agile and efficient, and ensuring that all teams are aiming for the same goal.

This platform combines the predictive power of ML with the optimization prowess of OR. This powerful combination allows us to make intelligent decisions across the entire supply chain. All our models are proprietary and utilize commercial solvers and specialized heuristics to solve problems at the massive scale at which Flipkart operates.

These models work together as an integrated system. The forecasts are mutually consistent and deeply integrated into the optimization models, and the optimization models are coupled across capacity, inventory, and network flow planning. This tight integration creates a highly optimized supply chain.

Forecasting Layer

Accurate demand forecasting is critical for optimal decision making whether it is tactical resource allocation or real-time order fulfillment. In our business, demand can experience sudden shifts because of major sales events or COVID-like events; in addition, factors such as product end-of-life planning can drive gradual changes in demand that cannot be explained by time-based factors. We need to incorporate an understanding of both sudden and gradual shifts to forecast demand accurately.

In addition, supply chain planning in our business involves a wide variety of use cases that rely on demand forecasts. Each use case requires forecasts at different granularities in terms of product, geography, and time. For example, to optimize procurement decisions, we need to forecast the daily demand of a specific product over the next few weeks in a localized region; however, to determine a marketing plan for a product category, we need to forecast the aggregate demand across the nation over a three-month period. These challenges motivated us to build an interconnected forecasting layer, which generates accurate demand estimates at different levels and time horizons across our product portfolio and use cases.

Solution Approach

Flipkart's demand forecasting journey has progressed through several stages. Before we started, business

teams used simple moving-average methods, which relied solely on historical averages.

These methods, although simple, were largely inaccurate because of a recency bias. We introduced more sophisticated statistical models such as autoregressive integrated moving average and exponential smoothing, and these enabled us to capture time-based variability more accurately. However, these methods were largely inaccurate for product categories such as grocery and cellphones, which had high demand volatility because of demand shifts driven by various factors such as seasonality, festivities, and new product launches. We achieved the next jump in accuracy by combining ML methods such as CatBoost (Prokhorenkova et al. 2018) with statistical models. Using an ensemble of decision trees, CatBoost models capture demand shifts effectively; although statistical models capture the time-based variations effectively, a combination of the two led to more accurate forecasts overall.

In 2023, we transitioned to deep learning models, specifically long short-term memory and Sequence2Sequence models (Sutskever et al. 2014). Unlike traditional ML models, these models are able to capture complex trends without explicit feature engineering. These models also allowed us to include nontabular data, such as textual product descriptions and catalog images, as features. These enhancements led to a further improvement in forecast accuracy. Finally, although these are well-known modeling techniques, optimizing these methods to achieve high accuracy across our use cases involved a significant amount of research and experimentation. Appendix A provides an overview of our study on the various deep learning architectures for time-series forecasting.

Reconciliation of Forecasts

Although we generate forecasts at various levels of granularity, these forecasts must be mutually consistent to drive optimized decision making across the supply chain. For example, the forecasts for a product across zip codes should be consistent with the national forecast for that product. This may not organically be the case because demand forecasts are generated independently. Also, planners may adjust specific forecasts based on business insights. For example, a planner may adjust the forecast for products of a specific brand to account for a promotion. This effect needs to be propagated up to the category level and down to the product level to ensure the integration of that insight into the overall forecasts.

We take a two-step approach to ensure consistency of our overall portfolio of forecasts. In step 1, we generate the best possible forecasts for all relevant time series. These forecasts include strategic inputs from planners. In the second step, we execute a reconciliation model (Van Erven and Cugliari 2015, Wickramasuriya et al. 2019), which takes related forecasts at different granularities and generates the mutually consistent final forecasts.

The above implementation changes together enabled us to achieve a 10%–50% improvement, depending on product category, in demand forecast accuracy. These improvements enable better decision making in the operational efficiency layer.

Operational Efficiency Layer

In this section, we describe the capacity and inventory planning pillars of our operational efficiency layer (we do not cover the network flow planning pillar in the interest of space). We focus on a few problems in these two pillars.

Capacity Planning

The first pillar of our operational efficiency layer is capacity planning. We run a complex supply chain that includes various resources, such as personnel, our in-house delivery trucks and drivers, and third-party delivery providers. The goal of capacity planning is to provide exactly the right number of resources at each stage in the supply chain so that we maximize delivery speed and minimize costs.

Two aspects make capacity planning challenging. The first aspect is the sheer scale of our network and our broad product portfolio. We have thousands of facilities, which are tightly coupled together, and we deliver an enormous range of products, each of which requires a different amount of handling capacity. Second, our order patterns are highly variable because of a combination of customer and seller behavior and our own interventions, such as sales events.

We solve capacity planning through a multiple-step process. We first predict how many shipments will flow from each FC to each zip code. We do this for each day and for each product category. Using these load estimates, we solve a series of optimization problems to optimize personnel and transportation capacity across the network. The final piece of the capacity puzzle involves near real-time optimization to optimize delivery speeds for select orders. We describe each of these steps in more detail in the following sections.

We note that the decisions across these modules are tightly coupled. As a result, making uniform progress across all the modules is important if we are to realize a tangible impact on the end business metrics. Overall, our models for capacity planning have led to 10% more accurate estimates for shipment volumes, an increase of more than 10% in personnel utilization, and a significant reduction in our supply chain costs.

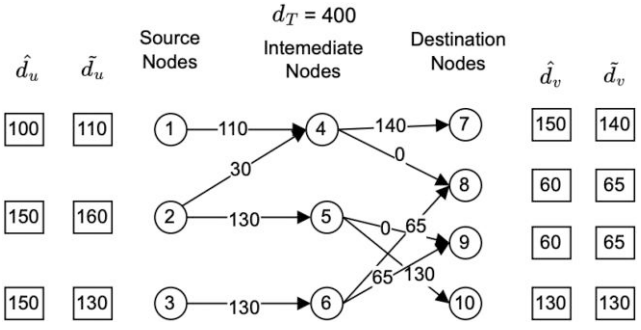
Stage 1: Source–Destination Load Estimation. The foundation of our capacity planning process is a robust load estimation framework that predicts the number of

shipments that will flow daily from each FC to each destination zip code. To build this supply–demand balance, we need to account for the demand for each product category at each zip code, the available inventory at each source, and the network structure. We also have to account for how shipments are fulfilled. Ideally, we want each order to be fulfilled from the nearest FC so that we maximize speed; however, that may not always be feasible for cost reasons.

Modeling all these features from first principles is not practical. We take a more ingenious approach. We first disaggregate the national-level forecast for each product category to FC and zip code granularity. We allow planners to refine these initial forecasts at each node based on their insights. They do this based not just on historical data but also on key insights such as expected changes to the supply chain, inventory availability, demand patterns, and company priorities around speed and cost. In our dynamic business setting, incorporating these insights leads to much more realistic and accurate predictions.

The next step is to superimpose supply chain network constraints on these node-level forecasts. We model this problem as a bipartite graph in which FCs are source nodes and zip codes are demand nodes. Edges represent shipments between the nodes. We take as input the estimated demand at each zip code, the estimated shipments from each FC, and the network constraints. The goal of the optimization model is to predict the final load at each source and destination node that minimizes deviations from the input estimates, satisfying all the constraints. We present an illustration of this supply chain network-aware source–destination load estimation in Figure 4.

Figure 4. Supply Chain Network–Constrained Source–Destination Load Estimation



Notes. The figure depicts a small network of 10 vertices, consisting of three sources, three intermediate nodes (hubs), and four destinations. The initial forecasts generated for each source node and destination node, represented by $\hat{d}_u$ and $\hat{d}_v$, are independently generated and do not incorporate (i.e., are not aware of) the supply chain constraints, such as the serviceability of a zip code by an FC. As a result, the initial forecasts indicate an imbalance. For instance, source node 3 has excess inventory because it only needs to fulfill demand for destination nodes 8 and 9, which have a combined demand of 120. The corrected forecasts, denoted by $\tilde{d}_u$ and $\tilde{d}_v$, ensure that the total demand is appropriately distributed across the network (i.e., aware of the supply chain constraints).

Given the size of our network, this is a large-scale optimization problem that is hard to solve. We tackle this problem by decomposing it into thousands of subproblems, which we solve in parallel. Refer to Naidu et al. (2022) for more details. Overall, using this approach, we align independent forecasts at the FC and zip code levels with network structure, ensuring consistency across all levels. Empirical evaluations demonstrate that this approach significantly outperforms conventional forecasting methods, yielding high accuracy by integrating granular demand patterns with structural constraints inherent in the supply chain network.

Stage 2: FC Capacity Planning. The source–destination load estimation from stage 1 gives us an estimate of the landing volumes at each facility (e.g., FC, last-mile hub). In stage 2, we optimize the personnel capacity needed at each facility to handle this load. As an example, we minimize the overall personnel cost at an FC by optimizing the cross-utilization of personnel across different functions. Each FC in Flipkart has multiple functions, such as inbound (i.e., inward) movement of inventory, interwarehouse inventory transfers (IWIT), outbound functions (i.e., picking, packing, and dispatching of customer orders), and returns. Each function in an FC hires personnel on a monthly basis. Moreover, the demand (i.e., number of customer orders) at an FC has significant day-over-day variability in a month because of sales events. However, some functions such as inbound movement of inventory happen only on specific days and have weekly or monthly targets with bursty personnel needs. Labor rules and regulations require personnel to be hired for longer periods of time. This means that any additional personnel hired to handle demand peaks results in an overall underutilization. Given this, we cross-utilize personnel across FC functions to reduce overall cost and improve personnel utilization. We formulate this as a mixed-integer programming problem that determines the monthly personnel to hire for each functional area and the daily optimal cross-deployment of these personnel.

Stage 3: Delivery Carrier Allocation and Load Optimization. Given the scale of Flipkart's operations and with the expanse and diversity of India's geography, we utilize third-party delivery carriers to extend the reach and capacity of our supply chain. Delivery carriers differ in their speed, cost, and reliability. We plan the utilization of delivery carriers such that the overall speed, cost, and reliability of our supply chain is optimized. We built a multiple-criteria optimization model, combined with heuristic algorithms, to plan

allocation of shipments to various logistics delivery companies at various stages of the shipment process. This model minimizes overall supply chain costs, accounting for supply chain node-level supply–demand balance, operational constraints (e.g., carrier capacity and serviceability), contractual constraints (e.g., minimum shipments per carrier), and customer experience factors (e.g., carrier service quality).

Stage 4: Real-Time Capacity Orchestration. The last piece of the capacity planning problem involves near real-time optimization to optimize delivery speeds for select orders. The monthly capacity plan that we describe above is a direct input into our supply chain execution systems. As orders stream into our platform, our systems block capacities in real time and determine the delivery date for each order. Once the capacity on a given day has been exhausted, the remaining orders spill to the next day and result in longer delivery times.

During the year, customer demand in Indian e-commerce has dramatic peaks, which are largely driven by cultural festivities that often coincide with major sales events. As an example, the Big Billion Days festival, which is considered one of the largest online shopping festivals in India, brings more than 350 million customers onto the platform with more than 90 million orders placed over a one-week period and a peak volume of more than three million orders per hour. Similar events (at different scales) are prevalent throughout the year. These demand spikes lead to an overall platform speed degradation because of a mismatch in demand and supply capacities. This often results in poor and inconsistent customer experiences.

However, order priorities differ. For example, customers who subscribe to our paid loyalty program expect fast deliveries even during sales events. We enable this preferred customer experience by reserving network capacities to process high-priority orders (Pibernik and Yadav 2009, Samii et al. 2011, Cheng et al. 2016). The challenge is that we do not know a priori the exact demand for these priority orders. If we reserve too little capacity, we experience poor speed for priority orders. If we reserve too much, we end up with wasted capacity.

To tackle this problem, we built a stochastic programming model, which we describe in Kumar et al. (2023), that accounts for multiple demand scenarios and optimizes capacity reservation for high-priority orders (e.g., a Flipkart loyalty member, a speed-sensitive product such as diapers, which customers expect to receive within one to two days), minimizing the likelihood of underutilization. We describe it in more detail below.

Background. Our FCs process and dispatch orders in waves, and each wave is defined by an end time (also referred to as wave cutoff time) and a processing capacity as described in Kumar et al. (2024). As an example, an FC can have five waves (4–9 am, 9 am–12 pm, 12–4 pm, 4–11 pm, 11 pm–4 am), each with a different processing capacity. When a customer places an order, our execution system identifies the earliest FC wave with available capacity to assign the order for processing and dispatch. On a normal day, the capacity at each FC typically ensures that the orders can be processed and dispatched on the same day. However, during sales events, the FC demand often exceeds its capacity, resulting in significant speed degradation. Moreover, the demand during sales events is highly variable, and each event tends to have a unique demand pattern.

Solution Approach. To handle the high demand variability during events, we designed our overall solution to use a combination of a predictive module and a reactive module (see Figure 5). The predictive module is powered by a probabilistic demand forecaster along with a stochastic programming model.

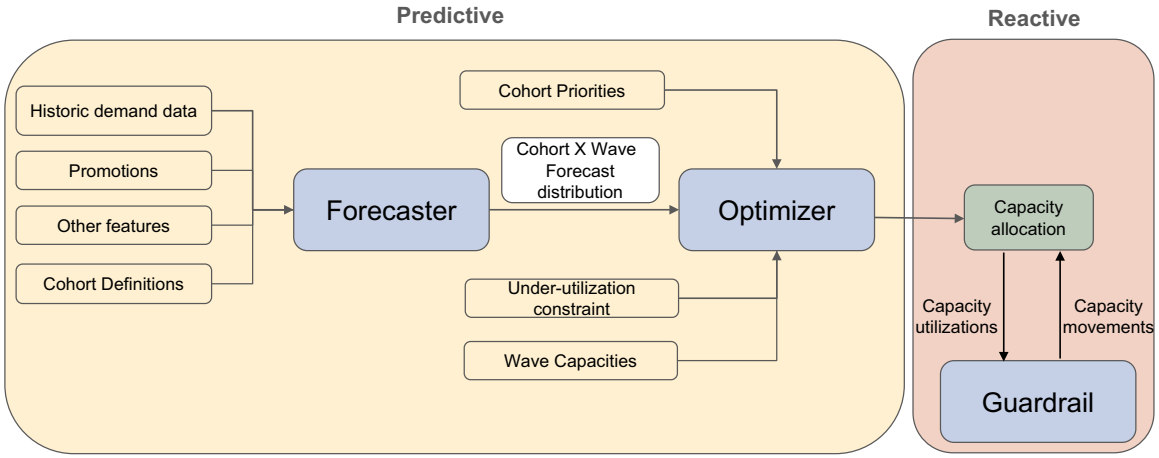
i. Forecaster: This module outputs probabilistic forecasts (i.e., a probability distribution of the future demand as shown in Figure 6) for each cohort–wave combination at an FC (where the cohorts can be high-priority orders and standard orders). The inputs to this model include historical sales patterns during various events, details on the scale and nature of those events, and other temporal features such as hour of the day.

ii. Optimizer: Given the probabilistic demand forecasts for each cohort and FC wave, we solve a stochastic optimization model to determine an optimal allocation of the wave capacity across the cohorts. This model determines the capacity distribution for each wave in the FC over the entire event duration. It is designed to minimize the spillage of the protected cohorts to the next FC wave, adhering to the overall underutilization constraint. The output of this module is the capacity to be assigned for each cohort–wave combination in the FC. Appendix B describes the optimization model and the solution.

iii. Guardrail: This is a reactive module that observes the capacity utilization across the cohorts in near real time (every half hour) and reallocates the capacity if required. This module helps handle unexpected demand fluctuations that are observed during the course of a sales event.

This solution has been in production since 2022 and has served as a critical component for speed allocation across our entire supply chain. It runs near real time across all our FCs during sales events and has helped us increase the percentage of high-priority orders that we are able to deliver within two days by more than 20%.

Figure 5. (Color online) Flipkart's Real-Time Capacity Orchestration Solution



Note. This combination of a predictive and a reactive model was critical to handling high demand variability during our sales events.

Inventory Planning

The next pillar of our operational efficiency layer is inventory planning. The goal of inventory planning is to have the right product selection, in the right quantity, and at the right location. Optimized inventory placement is key to serving customer demand with the fastest possible delivery speed, minimizing transportation and inventory costs. We need to achieve this goal, accounting for the demand and supply uncertainty across a large product range, the majority of which have low demand, and continuous product churn.

Our platform enables the following three key steps in the life cycle of inventory planning.

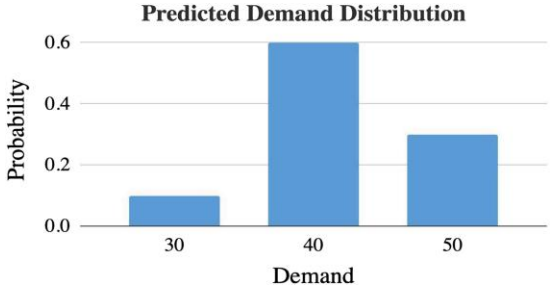
i. Inventory procurement planning: determining when and how much seller inventory to procure at each FC for each product.

ii. Inventory rebalancing (or IWIT): rebalancing the inventory across the FCs on a daily basis to match demand patterns across geographies.

iii. Inventory health monitoring: disposing of the inventory that is not selling through markdowns and returns to the sellers.

More accurate demand forecasts from our platform, together with in-house statistical and optimization models that recommend optimal inventory quantities, have improved the buying and placement decisions, thus leading to a 10% reduction in working capital for sellers on the platform and higher regional utilization (i.e., percentage of customer demand that can be served by the closest FC) across our FCs.

Figure 6. (Color online) Sample View of the Probabilistic Demand Forecast Used in the Real-Time Capacity Orchestration Solution



IWIT Optimization. In this section, we describe the inventory rebalancing problem in more detail. The goal of IWIT is to optimally rebalance the inventory across the FCs so that the products are placed closest to the demand. This enables us to deliver products faster and at a lower cost. We need to rebalance for two primary reasons:

- First, some sellers cannot deliver inventory to FCs across the country.
- Second, the procurement cycles are typically longer, and our forecasts are not perfect. As a result, the initial placement of the inventory at the time of procurement is often not optimal.

During an IWIT, trucks move the inventory between our FCs. The amount of inventory that we can move each day is limited by various operational factors such as personnel availability, storage availability, and minimum truck utilization constraints (e.g., at least 70% truck utilization). Given these constraints, prioritizing the right set of products and quantities to rebalance is important.

In the past, IWIT was planned manually using a combination of simple forecasting models and heuristic rules. However, accounting for uncertainty in demand forecasts, variability in seller deliveries, and an exceptionally large number of product and FC combinations, together with a need to honor operational constraints, made it difficult for planners to manually optimize IWIT plans.

Our goal was to bring the power of prediction and optimization to make the right inventory rebalancing

decisions within a constrained environment (Agrawal et al. 2004, Meissner and Senicheva 2018, Naderi et al. 2020, Bellala and Goel 2022, Zhou and Wang 2022). Below are our key contributions:

- We formulated the IWIT problem as a large-scale optimization problem. We developed an efficient and scalable local neighborhood search heuristic that decomposes the optimization problem into several subproblems in which each subproblem can be solved to optimality. We determined through extensive numerical experiments that the developed solution achieves near optimality to the original large-scale optimization problem with an optimality gap of 3%.
- One of the key requirements of the inventory planners is interpretability of the IWIT output. To enable interpretability, we created simple routines that generate insightful and actionable reports to the business team. Specifically, we generate (i) a waterfall analysis that estimates the impact of each constraint on the IWIT quantity and (ii) a detailed report on the drop in IWIT quantity because of shortages in the budget, the personnel, and storage capacity in each FC. These reports enable the planners to address some of these constraints (e.g., acquire additional personnel) before the execution cycle.

Solution Overview. The following decisions must be made for IWIT: (i) What set of products should be rebalanced for each pair of FCs, and what are the corresponding quantities? (ii) What is the optimal truck capacity needed to enable the movement of these products?

The goal is to determine the optimal plan that maximizes in-stock (percentage of instances a product is shown as in-stock on our application) and regional utilization subject to cost and other operational constraints. See Table 1 for a sample output.

We decompose the IWIT problem and solve it in four steps. See Table 2 for a sample output from each step.

Step 1: We use the product and regional-level daily forecasts to estimate the ideal inventory quantity for each product at each FC. We further built a demand-sensing module to account for the forecast errors and the variability in seller deliveries. The demand-sensing module is a reactive module that modifies the forecasts based on the recent trend in the forecast errors. Together, this enables us to reliably estimate if we have excess or deficit inventory for each product at each FC:

$$Deficit_{ij} = max\{0,\ Ideal\ Inventory_{ij} - Current\ Inventory_{ij}\}$$
$$Excess_{ij} = max\{0,\ Current\ Inventory_{ij} - Ideal\ Inventory_{ij}\},$$
$$for\ product\ i\ at\ FC\ j$$

where the current inventory is the sum of the on-hand inventory available in the FC, in-transit IWIT inventory, and in-transit shipments from the sellers.

Step 2: Given the constrained environment, we define a utility function that assigns a priority to the deficit associated with each product and FC combination:

$$utility_{ij} = \alpha \frac{Deficit_{ij}}{Ideal\ Inventory_{ij}} * 100 + (1-\alpha)\ Sales\ Contribution_{ij}.$$

The first term in the utility function represents the percentage deficit relative to the ideal inventory level. A higher percentage denotes that we are close to being out of stock for product i at FC j. The second term represents the percentage sales of the product at FC j, for which a higher percentage denotes a high-volume and fast-moving product. The parameter α is determined by planners and is typically set to 0.5. Note that the above utility function serves as a proxy for both the in-stock and regional utilization metrics.

Step 3: We solve an optimization problem that maximizes the overall utility of the IWIT movements subject to all operational constraints (e.g., IWIT budget, FC storage capacity, IWIT inbound and outbound capacities, desired truck utilization factor). We formulate this as a mixed-integer linear programming problem. The size of this model is on the order of 10 million decision variables in our typical IWIT cycle. We developed an efficient local neighborhood search heuristic that solves this problem to near optimality (based on

Table 1. Output from Our IWIT System

Stock keeping unit	Source FC	Source slot	Destination FC	Destination slot	IWIT quantity
MarQ AC	Kolkata FC	August 1, 2025	Bangalore FC	August 5, 2025	70
MarQ AC	Kolkata FC	August 3, 2025	Bangalore FC	August 7, 2025	30
MarQ AC	Kolkata FC	August 1, 2025	Patna FC	August 3, 2025	50

Source FC	Source slot	Destination FC	Destination slot	Trucks
Kolkata FC	August 1, 2025	Bangalore FC	August 5, 2025	One big, one small
Kolkata FC	August 1, 2025	Patna FC	August 3, 2025	One big
Kolkata FC	August 3, 2025	Bangalore FC	August 7, 2025	One small

Notes. The upper table shows the IWIT quantity across different products and FC pairs. The lower table shows the truck capacity needed to enable the movement of these products.

Table 2. Sample Output from Each Step of the IWIT Solution

Step 1

Stock keeping unit	FC	Excess/deficit
MarQ AC	Kolkata FC	200
IFB washing machine	Kolkata FC	−15
MarQ AC	Patna FC	−50
MarQ AC	Bangalore FC	−100
IFB washing machine	Bangalore FC	5

Step 2

Stock keeping unit	FC	Deficit	Utility
IFB washing machine	Kolkata FC	−15	60
MarQ AC	Patna FC	−50	90
MarQ AC	Bangalore FC	−100	100

Step 3

Stock keeping unit	Source FC	Source slot	Destination FC	Destination slot	IWIT quantity
MarQ AC	Kolkata FC	August 1, 2025	Bangalore FC	August 5, 2025	70
MarQ AC	Kolkata FC	August 3, 2025	Bangalore FC	August 7, 2025	30
MarQ AC	Kolkata FC	August 1, 2025	Patna FC	August 3, 2025	50

Step 4

Source FC	Source slot	Destination FC	Destination slot	Trucks
Kolkata FC	August 1, 2025	Bangalore FC	August 5, 2025	One big, one small
Kolkata FC	August 1, 2025	Patna FC	August 3, 2025	One big
Kolkata FC	August 3, 2025	Bangalore FC	August 7, 2025	One small

numerical experiments). Appendix C describes the optimization model and the heuristic.

Step 4: The final step of our solution is to determine the optimal truck capacity needed between each pair of FCs to execute the IWIT plan from step 3. We solve this for each lane (i.e., pair of FCs) independently using an integer linear program. The objective of this model is to maximize the number of IWIT units in the lane using the fewest number of trucks.

The IWIT process runs at different frequencies—weekly for bulky product categories such as refrigerators, daily for product categories such as cellphones, and multiple times a day for groceries. The process is fully automated and rebalances more than half a million units each day across the country. Our improvements to IWIT decisions led to better inventory placement and resulted in a 50% increase in one-day deliveries and a 43% increase in two-day deliveries.

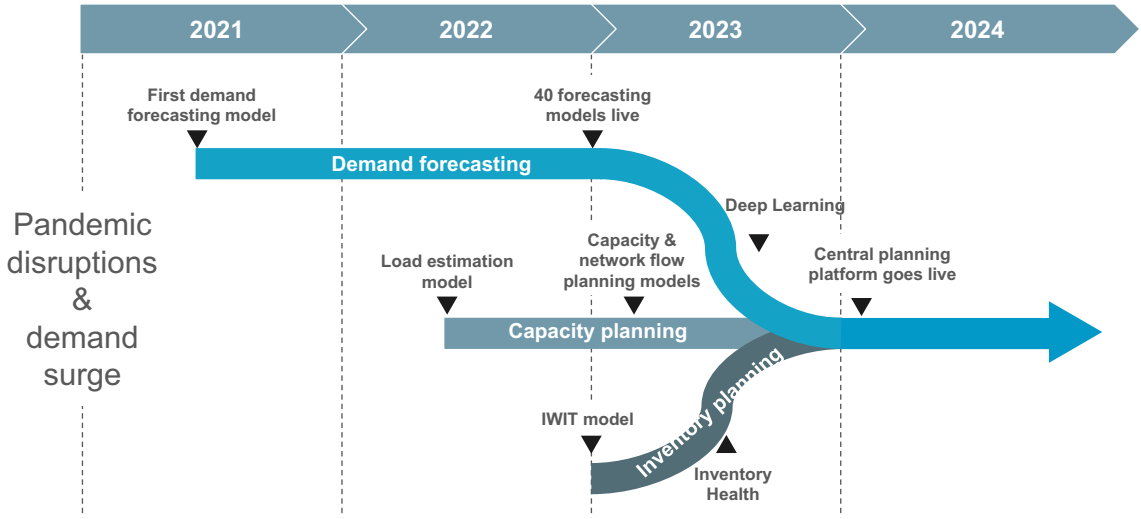
Implementation Journey

We embarked on this journey in 2021 (see Figure 7) by building demand forecasting models for individual teams who used them to plan the procurement of inventory from sellers. As we started to see success in categories such as cellphones and groceries, we started receiving interest from stakeholders of other categories to build models for them. By the end of 2022, we had built 40 unique models catering to a variety of demand forecasting needs across categories and use cases.

In parallel, as a result of supply chain disruptions and rapid demand increases, our existing capacity planning processes were not able to keep pace. We started by building the load estimation model in 2022, which helped the operations teams understand the expected loads at each part of the network and then plan personnel and other resources for the network. These initial seeds of ML-driven solutions saw great acceptance, and we were able to take this momentum and kick-start our efforts on building the more traditional OR/optimization use cases to support inventory, capacity, and network flow planning. In early 2023, we built the blueprint for a central planning system that would integrate our ML and OR efforts in a single central planning platform. We built this out iteratively and were able to go live with the first version of the complete system by early 2024.

The central planning platform is much more than just ML and OR models. It is an integrated platform that enables end-to-end solutions for planning. Apart from the ability to run the ML/OR models efficiently, the platform enables visibility of inputs, outputs, and intermediate steps of planning to all relevant stakeholders, improving transparency and trust in decision making. We have also invested in explainability of key decisions (i.e., the ability to understand and interpret

Figure 7. (Color online) Flipkart's Central Planning Platform Implementation Journey



decisions made through the platform) and built simulation tools that enable smart human overrides and intervention on the generated plans.

Management buy-in was critical for success, and we instituted and followed a rigorous process of backtesting (i.e., testing using historical data) and a user acceptability testing (UAT) period of two to three months to establish impact and build management confidence. The UATs varied by use case, ranging from A/B tests to test out new inventory placement policies, to running forecasting models in shadow mode to compare outcomes against existing techniques. To ensure continued trust, we built robust governance processes to track the accuracy and effectiveness of our models.

Impact

Our platform combines the strengths of ML and OR to optimize decisions across all three pillars of supply chain planning. Tight coupling of these models across the three pillars amplifies their impact and has enabled the platform to create more value than its individual parts provide. The platform has enabled significant cost savings and incremental net sales for Flipkart and the sellers. Some of the highlights of the impact are listed in Table 3.

Apart from the positive impact to business metrics, this platform has completely changed how Flipkart's planning and category management teams work. Previously, these teams would need to add members each year to handle the rapidly escalating size and complexity of business. With the central planning platform, the teams are now able to handle multifold size and complexity without increasing the number of team members.

Additionally, planning cycles are shorter and more frequent, allowing teams to respond faster to changing business conditions and improving the overall accuracy of planning. For example, the personnel planning cycle has decreased from a 10-day effort to a one-day effort, allowing weekly refresh of these plans.

The platform is highly modular, and thus, we are exploring its expansion to additional lines of business within Flipkart and to other Flipkart Group companies. Some of the technical contributions (e.g., stochastic programming–based real-time capacity orchestration for delivery speed protection, demand forecasting and reconciliation to achieve aligned plans) are novel and can be adapted to other contexts. As an example,

Table 3. Summary of the Business Impact of the Central Planning Platform and Its Components

Metric	Impact	Attributed to
Manpower utilization	10% increase	Capacity planning
Unhealthy inventory for sellers	50% reduction	Demand forecasting and inventory procurement planning
Overall inventory levels for sellers	10% reduction	Demand forecasting and inventory procurement planning
One-day deliveries	50% increase	Demand forecasting and Inventory rebalancing (IWIT)
Two-day deliveries for high-priority orders	20% increase	Real-time capacity orchestration
Capacity planning cycle time	90% reduction	Central planning platform

demand forecasting and reconciliation techniques are applicable in other industries; examples include retail, manufacturing, utilities (e.g., to plan generation/procurement of energy and determine the renewables mix), and cloud service providers (e.g., to plan data center capacities across geographies). Similarly, the stochastic programming–based real-time capacity orchestration for delivery speed protection is applicable in other marketplaces such as ride-sharing apps and food-delivery apps that face similar resource constraints during peak hours.

Finally, e-commerce across the world is growing and evolving rapidly. Our experience in building a supply chain planning platform to deliver business impact, scalability, and agility in a rapidly growing and dynamic context can inspire others to apply OR and ML to help their companies succeed.

Conclusion

In this paper, we describe the work done by Flipkart teams since 2021 in building a central planning platform to help automate and optimize Flipkart's vast suite of retail operations. This platform was built with a vision of transforming Flipkart's central planning and operations and enabling it to grow sustainably and succeed in a dynamic and evolving market.

The platform helped Flipkart achieve significant reductions in costs and higher delivery speeds through better inventory, capacity, and network flow planning. Beyond the impact on business metrics, the automation and optimization capabilities of this platform have had a transformational impact on the organization, enabling higher productivity and agility across our planning, operations, and business partners.

Our solution has brought state-of-the-art technologies and innovative solutions built on OR and ML, together with a collaboration and workflow management platform to produce an end-to-end platform that delivers agility, connectedness, and intelligence.

Finally, we are continuing to work with other companies (e.g., Myntra, Cleartrip) within the Flipkart Group to implement the platform for their use cases. This platform is a critical component of Flipkart's sustained success.

Acknowledgments

Authors are listed in the alphabetical order of their last name.

Appendix A. Deep Learning Models for Sales Forecasting

A.1. Model Overview

When transitioning from machine learning–based forecasting models, we evaluated a range of deep learning architectures for time series demand forecasting.

- Long short-term memory (LSTM) Seq2Seq: A baseline encoder–decoder architecture using LSTM cells to model temporal dependencies.
- Transformer Seq2Seq: A self-attention–based model using encoder–decoder architecture without recurrence (Vaswani et al. 2017).
- Temporal fusion transformer (TFT): An interpretable model that adds gating mechanisms and variable selection to transformer blocks (Lim et al. 2021).
- Time series mixer (TSMixer): A lightweight multilayer perceptron-based model that performs temporal and feature mixing without recurrence or attention (Ekambaram et al. 2023).
- Multiquantile recurrent neural network (MQRNN): A recurrent architecture with quantile loss for probabilistic multihorizon forecasting (Wen et al. 2018).
- LSTM Seq2Seq + Skip: An enhanced version of the LSTM Seq2Seq model with skip connections from decoder inputs to outputs and multihead attention between decoder states and encoder outputs.

A.2. Data Set and Evaluation Metric

We evaluated these models on a large data set consisting of three years of historical daily sales data for more than 100,000 products (SKUs) across diverse business units (home furnishings/decor/appliances, electronics, books, and general merchandise). The features used in these models include historical sales, historical inventory views, sale event calendars, festive calendars, list prices, and pricing discounts. The output forecast is generated at a granularity <SKU, daily> for a 30-day forecast horizon.

We used weighted mean absolute percentage error (wMAPE) to evaluate the performance of the model, in which the weights correspond to the percentage of sales volume of each SKU. This metric is robust to scale and provides stable business-aligned accuracy estimates:

$$\text{wMAPE} = \frac{\sum_i \sum_t |y_{it} - \hat{y}_{it}|}{\sum_i \sum_t y_{it}},$$

where y_{it} denotes the actual sales of SKU i on day t and $\hat{y}_{it}$ denotes the forecast for SKU i on day t.

A.3. Final Deployed Architecture

Based on extensive experiments, an LSTM-based sequence-to-sequence network with skip connections and attention performed the best on our data set. This design improved temporal alignment, gradient flow, and generalization across promotional and seasonal patterns. The following are details of this model:

- Encoder LSTM:

$$\mathbf{h}_t = \text{LSTM}_{\text{enc}}(\mathbf{x}_t, \mathbf{h}_{t-1}).$$

- Decoder LSTM:

$$\mathbf{s}_\tau = \text{LSTM}_{\text{dec}}(\mathbf{x}_{t+\tau}, \mathbf{s}_{\tau-1}).$$

- Multihead attention between decoder state and encoder outputs:

$$\textbf{Attention}_\tau = \text{MultiHead}(\text{Q} = \mathbf{s}_\tau, \text{K} = \mathbf{H}, \text{V} = \mathbf{H}).$$

Table A.1. The wMAPE(%) of Various Time Series Forecasting Models Across Three Business Units in Flipkart (Lower Numbers Are Better)

Model	Home	Electronics	Books and general merchandise
ARIMA + GBDT ensemble (previous baseline)	77.4	65.2	59.8
LSTM Seq2Seq	73.7	62.1	53.3
Transformer Seq2Seq	78.2	66.2	60.1
TFT	75.4	64.2	54.2
TSMixer	76.3	67.3	55.6
MQRNN	77.1	64.2	57.2
LSTM Seq2Seq + Skip (production model)	70.6	59.1	53.3

- Gated skip connections:

$$\mathbf{z}_\tau = \text{LayerNorm}(\mathbf{s}_\tau + \textbf{Attention}_\tau),$$
$$\text{GLU}(\mathbf{z}) = \sigma(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) \odot (\mathbf{W}_2\mathbf{z} + \mathbf{b}_2),$$
$$\mathbf{z}_\tau = \text{LayerNorm}(\text{GLU}(\mathbf{z}_\tau) + \text{Skip}(\mathbf{x}_{t+\tau})).$$

- Final prediction:

$$\hat{y}_{t+\tau} = \mathbf{W}_o\mathbf{z}_\tau + b_o.$$

- Quantile loss for training:

$$\mathcal{L}_q(y_t, \hat{y_t}) = \max(q(y_t - \hat{y_t}), (q-1)(y_t - \hat{y_t})),$$

where $\mathbf{x}_t$ denotes the inputs to the model, $\mathbf{h}_t$ denotes the hidden state in the encoder, $\mathbf{s}_t$ denotes the hidden state in the decoder, y_t denotes the target variable (actual sales), $\hat{y}_t$ denotes the model output (predicted sales), $\mathbf{W}_i$ and $\mathbf{b}_i$ are the model parameters, and $q \in [0,1]$ denotes the quantile value.

A.4. Model Performance

Table A.1 shows the wMAPE (percentage) of the various deep learning architectures presented in Appendix A.1 for three different business units. Though the forecast horizon is 30 days, the wMAPE is computed for the third week's demand forecast (the third week demand forecast is typically used for procurement decisions). These results demonstrate a consistently better performance (lower wMAPE %) of the LSTM Seq2Seq + Skip model over the remaining state-of-the-art deep learning architectures as well as our previous machine learning-based model (an ensemble model consisting of autoregressive integrated moving average (ARIMA) and gradient-boosted decision trees).

Appendix B. Real-Time Capacity Orchestration

We describe the optimization model used in the real-time capacity orchestration.

B.1. Sets

- $\mathcal{C}$: Set of waves, indexed by c.
- $\mathcal{P}$: Set of cohorts, indexed by p.
- $\mathcal{P}^{\mathcal{H}}$: Set of high-priority cohorts, indexed by p.
- Ω: Set of scenarios, indexed by w.

B.2. Parameters

- $\tilde{D}_{p,c,w}$: Forecasted demand of scenario w for cohort p in wave c.
- π_w: Probability of occurrence of scenario w.
- K^{spill}: Allowed spillage value after recovery period.
- K_c^{cap}: Total capacity of wave c.
- c^E: Last wave of the event (before recovery).
- λ: Penalty for buffer spillage.
- $\mathcal{N}^{\mathcal{C}}$: Total number of waves.

B.3. Variables

- $q_{p,c}$: Capacity allocated for cohort p in wave c.
- $s_{p,c,w}$: Spillage in wave $c <= c^E$ for cohort p in scenario w.
- $s'_{c,w}$: Spillage in wave $c > c^E$ in scenario w.
- β^{spill}: Buffer for spillage constraint.

B.4. Constraints

- Total capacity of a wave: The aggregate capacity of wave c across all cohorts for a scenario w must be equal to the total wave capacity K_c^{cap}:

$$\sum_{p \in \mathcal{P}} q_{p,c} = K_c^{\text{cap}} \qquad \forall c \in \mathcal{C}. \tag{B.1}$$

During the recovery period, no capacity reservations are made for protected cohorts, and hence, the entire capacity is allocated to the nonprotected cohort:

$$q_{p,c} = K_c^{\text{cap}} \qquad \forall p \in \mathcal{P} \setminus \mathcal{P}^{\mathcal{H}}, c \in \mathcal{C} \mid c > c^E. \tag{B.2}$$

- Spill in a wave: The spill in a wave is computed by the excess of demand over allocated capacity:

$$s_{p,1,w} = max(\tilde{D}_{p,1,w} - q_{p,1}, 0) \qquad \forall w \in \Omega, p \in \mathcal{P}, \tag{B.3}$$

$$s_{p,c,w} = max(\tilde{D}_{p,c,w} + s_{p,c-1,w} - q_{p,c}, 0) \qquad \forall w \in \Omega, p \in \mathcal{P}, c \in \mathcal{C} \mid 2 <= c <= c^E. \tag{B.4}$$

- Spill from last wave: The spill in the last wave is computed by the sum of spillages across all cohorts:

$$s'_{c,w} = \sum_p s_{p,c,w} \qquad \forall w \in \Omega, c \in \mathcal{C} \mid c > c^E. \tag{B.5}$$

The total spill from the last wave must be less than the allowed spillage value (K^{spill}):

$$s'_{c_{\text{last}},w} \le K^{\text{spill}} + \beta^{spill} \quad \forall w \in \Omega, \tag{B.6}$$

$$\sum_{w \in \Omega'} \pi_w * s'_{c_{\text{last}},w} \le K^{\text{spill}} + \beta^{spill}, \tag{B.7}$$

where $\Omega' = \{w \in \Omega : \pi_w > \pi^t\}$.

B.5. Objective Function

The objective is to maximize protection provided to high-priority cohorts or, alternatively, to minimize spillage of high-priority cohorts at a wave level:

$$\textit{Minimize} \quad \sum_{w \in \Omega} \pi_w \cdot \left(\sum_{p \in \mathcal{P}^{\mathcal{H}}} M_p^{big} \cdot \left(\sum_{c=1}^{c^E} s_{p,c,w} \right) \right) + \lambda * \beta^{spill}, \tag{B.8}$$

where $M_{p_1}^{big} > M_{p_2}^{big}$ if cohort p_1 has higher priority than p_2 and $M_{p_1}^{big}$ denotes the penalty for spilling orders in cohort p_1. The priority of the cohorts is determined by the

business. The penalty values are defined on the basis of the normalized demand of the cohorts. Let us denote W as the following sorted set:

$$W = \text{sort descending order of } \left\{ \frac{D_{p_1}}{\sum_p D_p}, \frac{D_{p_2}}{\sum_p D_p}, \cdots \right\},$$

where D_p is the demand in cohort p, then $M_{p_k}^{big} = W_k$, that is, the kth element of the set W.

B.6. Solution

To address the model complexity, we discretized the output of the probabilistic demand forecast to three bins for the immediate two FC waves and one bin for all future waves. This significantly reduced the number of potential scenarios. We solved the resulting optimization model using high-performance software for linear optimization (HiGHS 2025).

Appendix C. IWIT Optimization

We describe step 3 of our IWIT solution in more detail here.

C.1. Indices

- i: product.
- j: Source FC.
- k: Destination FC.
- s: IWIT outbound day at source.
- t: IWIT inbound day at destination.

C.2. Sets

- $\mathcal{F}$: Set of products eligible for IWIT.
- $\mathcal{W}$: Set of FCs.
- $\mathcal{C}$: Set of super categories or pallet verticals.
- $\mathcal{F}_c \subset \mathcal{F}$: Set of eligible products belonging to the super category c.
- $\mathcal{F}_v \subset \mathcal{F}$: Set of eligible products that are volumetric.
- T: IWIT planning horizon (in days).

C.3. Parameters

- $d_{i,k}$: Deficit quantity of product i at FC k.
- $e_{i,j}$: Excess quantity of product i at FC j.
- δ_i: Number of boxes per unit of product i.
- $OB_{j,s}$: Outbound capacity at FC j on day s.
- $IB_{k,t}$: Inbound capacity at FC k on day t.
- $S_{c,k,t}$: Available storage capacity for super category c at FC k on day t.
- B : IWIT budget.
- $C_{c,j,k}$: Cost per shipment for super category c from FC j to FC k.
- v_i: Volume of product i in cubic feet.
- V^s: Volume of a small truck in cubic feet.
- V^l: Volume of a large truck in cubic feet.
- N^s: Maximum number of small trucks.
- N^l: Maximum number of large trucks.
- γ^u: Maximum utilization of a truck (e.g., 95%).
- γ^l: Minimum utilization of a truck (e.g., 70%).
- MOQ: Minimum order quantity (i.e., minimum quantity required to be eligible for an IWIT).
- $w_{i,j,k}$: Utility score for fulfilling the deficit requirement of product i at FC k from FC j (as defined in the "Inter-Warehouse Inventory Transfers (IWIT)" section).

C.4. Variables

- $x_{i,j,k,s,t}$: IWIT quantity of product i from FC j on day s to FC k on day t.
- $y_{i,j,k,s,t}$: Binary variable; one if $x_{i,j,k,s,t} > 0$.
- $z_{j,k,s,t}$: Binary variable; one if $\sum_i x_{i,j,k,s,t} > 0$.

C.5. Constraints

The *Deficit* constraint ensures that the inbound quantity of a product to an FC does not exceed the deficit inventory in that FC. Similarly, the *Excess* constraint ensures that the outbound quantity of a product from an FC does not exceed the excess inventory available in that FC. The *Outbound* and *Inbound* capacity constraints limit the total daily outbound and inbound inventory in each FC to their respective capacity thresholds. The *Storage* capacity constraint ensures that the total inbound inventory within a super category adheres to the storage limits in that FC. The *IWIT Budget* constraint caps the total IWIT cost within an amortized cycle-level budget. The *Full Truck Load (FTL)* constraint requires trucks to be used at a minimum of 70% volume to avoid underutilization costs, whereas the *Maximum Truck Load (MTL)* constraint ensures enough buffer for safe loading and unloading. The *Minimum Order Quantity (MOQ)* constraint promotes bulk movements by imposing product-specific minimum shipment sizes per source–destination slot:

$$\sum_{j,s,t} x_{i,j,k,s,t} \le d_{i,k} \quad \forall i \in \mathcal{F}, k \in \mathcal{W}, \tag{Deficit}$$

$$\sum_{k,s,t} x_{i,j,k,s,t} \le e_{i,j} \quad \forall i \in \mathcal{F}, j \in \mathcal{W}, \tag{Excess}$$

$$\sum_{i,k,t} \delta_i \cdot x_{i,j,k,s,t} \le OB_{j,s} \quad \forall j \in \mathcal{W}, s \in T, \tag{Outbound Capacity}$$

$$\sum_{i,j,s} \delta_i \cdot x_{i,j,k,s,t} \le IB_{k,t} \quad \forall k \in \mathcal{W}, t \in T, \tag{Inbound Capacity}$$

$$\sum_{i \in \mathcal{F}_c, j, s, t' \le t} \delta_i \cdot x_{i,j,k,s,t'} \le S_{c,s,t} \quad \forall c \in \mathcal{C}, k \in \mathcal{W}, t \in T, \tag{Storage Capacity}$$

$$\sum_{i,j,k,s,t} C_{c,j,k} \cdot x_{i,j,k,s,t} \le B, \tag{IWIT Budget}$$

$$\sum_i v_i \cdot x_{i,j,k,s,t} \le \gamma^u \cdot (N^s \cdot V^s + N^l \cdot V^l) \cdot z_{j,k,s,t} \quad \forall j,k \in \mathcal{W}, s,t \in T, \tag{MTL}$$

$$\sum_i v_i \cdot x_{i,j,k,s,t} \ge \gamma^l \cdot V^s \cdot z_{j,k,s,t} \quad \forall j,k \in \mathcal{W}, s,t \in T, \tag{FTL}$$

$$MOQ \cdot y_{i,j,k,s,t} \le x_{i,j,k,s,t} \quad \forall i \in \mathcal{F}, j,k \in \mathcal{W}, s,t \in T. \tag{MOQ}$$

C.6. Objective Function

The objective is to maximize the total utility score:

$$\sum_{i,j,k,s,t} w_{i,j,k} \cdot x_{i,j,k,s,t}.$$

C.7. Solution

We used a local neighborhood search heuristic to efficiently solve this optimization problem with open-source,

mixed-integer programming solvers such as the COIN-OR Branch and Cut Solver. The algorithm begins by generating an initial feasible solution by rounding the solution to the relaxation of linear programming, followed by filtering based on MOQ, FTL, and MTL constraints. It then applies a series of local search heuristics that iteratively improve the solution by solving subproblems with fixed variable subsets. Each subproblem corresponds to a specific outbound slot, inbound slot, source, or destination and is small enough to be solved quickly to optimality.

References

Agrawal V, Xiuli Chao X, Sridhar Seshadri S (2004) Dynamic balancing of inventory in supply chains. *Eur. J. Oper. Res.* 159(2): 296–317.

Bellala G, Goel V (2022) Inventory placement optimization in e-commerce: The need for speed. *Informs Annual Conf.* (Institute for Operations Research and the Management Sciences, Catonsville, MD).

Cheng Y, Li B, Jiang Y (2016) Optimal choices for the e-tailer with inventory rationing, hybrid channel strategies, and service level constraint under multiperiod environments. *Math. Problems Engrg.* 2016(1):1–12.

Ekambaram V, Jati A, Nguyen N, Sinthong P, Kalagnanam J (2023) Tsmixer: Lightweight MLP-mixer model for multivariate time series forecasting. *Proc. 29th ACM SIGKDD Conf. Knowledge Discovery Data Mining* (Association for Computing Machinery, New York), 459–469.

HiGHS (2025) HiGHS—High performance software for linear optimization. Accessed August 28, 2025, https://highs.dev.

Kumar A, Bellala G, Bhat R, Desai S (2023) Optimal wave design in fulfilment centers—The key to faster speed. *Informs Annual Conf.* (Institute for Operations Research and the Management Sciences, Catonsville, MD).

Kumar A, Bandaru A, Bellala G, Goel V, Goswami A, Bhat R, Desai S (2024) PICASA—Optimal capacity allocation in supply chain for better speed offering. *Informs Annual Conf.* (Institute for Operations Research and the Management Sciences, Catonsville, MD).

Lim B, Arık S, Loeff N, Pfister T (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Internat. J. Forecasting* 37(4):1748–1764.

Meissner J, Senicheva OV (2018) Approximate dynamic programming for lateral transshipment problems in multi-location inventory systems. *Eur. J. Oper. Res.* 265(1):49–64.

Naderi S, Kilic K, Dasci A (2020) A deterministic model for the transshipment problem of a fast fashion retailer under capacity constraints. *Internat. J. Production Econom.* 227:107687.

Naidu KVM, Gupta P, Gujjula V (2022) Network aware forecasting for eCommerce supply planning. *Proc. 31st ACM Internat. Conf. Inform. Knowledge Management* (Association for Computing Machinery, New York), 1532–1541.

Pibernik R, Yadav P (2009) Inventory reservation and real-time order promising in a make-to-stock system. *OR Spectrum* 31(1): 281–307.

Prokhorenkova L, Gusev G, Vorobev A, Dorogush AV, Gulin A (2018) CatBoost: Unbiased boosting with categorical features. Accessed September 10, 2025, https://proceedings.neurips.cc/paper_files/paper/2018/file/14491b756b3a51daac41c24863285549-Paper.pdf.

Samii A-B, Pibernik R, Yadav P (2011) An inventory reservation problem with nesting and fill rate-based performance measures. *Internat. J. Production Econom.* 133(1):393–402.

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Preprint, submitted September 10, https://arxiv.org/abs/1409.3215.

Van Erven T, Cugliari J (2015) Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts. Antoniadis A, Poggi JM, Brossat X, eds. *Modeling and Stochastic Learning for Forecasting in High Dimensions* (Springer International Publishing, Basel, Switzerland), 297–317.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez LK, Polosukhin I (2017) Attention is all you need. Accessed September 10, 2025, https://dl.acm.org/doi/pdf/10.5555/3295222.3295349.

Wen R, Torkkola K, Narayanaswamy B, Madeka D (2018) A multi-horizon quantile recurrent forecaster. Preprint, submitted November 29, 2017, https://arxiv.org/abs/1711.11053.

Wickramasuriya SL, Athanasopoulos G, Hyndman RJ (2019) Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization. *J. Amer. Statist. Assoc.* 114(526): 804–819.

Zhou Z, Wang X (2022) Replenishment and transshipment in periodic-review systems with a fixed order cost. *Eur. J. Oper. Res.* 307(3):1240–1247.

Shubham Agarwal is a data scientist and has been at Flipkart since 2019. His professional experience spans developing optimization and machine learning products in the retail domain. He holds a bachelors degree in electronics and communication engineering with honors from the International Institute of Information Technology, Hyderabad.

Prateek Agrawal is the director of product for supply chain and inventory planning at Flipkart, where he joined in 2020. His prior work experience focused on strategy, business, and product in companies such as McKinsey, Myntra, and Blinkit. He holds an MBA from the Indian Institute of Management Calcutta and a BTech in production and industrial engineering from the Indian Institute of Technology Roorkee.

Anurag Allamsetty is a senior product manager at Flipkart since 2020. His professional experience spans large-scale solutions for automation, reducing supply chain cost and improving delivery speed to customers. He holds an MBA from the Indian Institute of Management Lucknow and a MTech + BTech in mechanical engineering from the Indian Institute of Technology (BHU) Varanasi.

Adarsh Attavar is a senior product manager at Flipkart, where he has been driving solutions across supply chain and demand platforms since 2020. He holds a BTech in mechanical engineering from the Indian Institute of Technology, Bombay.

Deekshith B is a data scientist and has been at Flipkart since 2021. His professional experience spans developing optimization and machine learning products in the retail and aerospace domains. He holds a masters degree from the Indian Institute of Science, Bangalore.

Gowtham Bellala is a principal data scientist with Flipkart since 2019, leading scientific work on the application of artificial intelligence, machine learning, and operations research in supply chain design, planning, and optimization. His prior work experience includes Hewlett Packard Labs and C3.ai in the United States. He holds a PhD in machine learning from the University of Michigan, Ann Arbor, and a BTech in electrical engineering from the Indian Institute of Technology, Madras.

Shobhit Bhatnagar is a data scientist and has been at Flipkart since 2022. His professional experience spans developing optimization products in the retail and energy domains. He holds a masters degree in industrial engineering and operations research from the Indian Institute of Technology, Bombay.

Hardik Choudhari is a product manager at Flipkart since 2021. His work experience spans building growth and supply chain products along with forecasting and planning platforms. He holds a bachelors degree in chemical engineering from the Birla Institute of Technology and Science, Goa.

Vikas Goel is a senior director of data sciences and has been at Flipkart since 2021. Prior to this, he held various technical and leadership roles with ExxonMobil in the United States and in India. He

holds a PhD from Carnegie Mellon University with specialization in operations research and a BTech in chemical engineering from the Indian Institute of Technology, Delhi.

Ananth Kachroo is a product manager at Flipkart driving custom automation and machine learning–based intelligence solutions in the demand and inventory planning domains. He has previously worked as a product manager in the business-to-consumer technology space and as a data scientist in targeted advertising. He holds a chemical engineering degree from Birla Institute of Technology and Science, Pilani.

Jay Kothadiya is a data scientist and has been at Flipkart since 2022. His professional experience spans developing optimization and machine learning products in the retail domain. He holds a masters degree from the Indian Institute of Technology, Kanpur.

Sai Anjani Kumar Kudupudi is a senior data scientist and has been at Flipkart since 2022. His professional experience spans developing optimization and machine learning products in the energy, retail, and airline domains. He holds a dual degree (BTech and MTech) in industrial engineering from the Indian Institute of Technology, Kharagpur.

Mayank Kumar is a senior data scientist and has been at Flipkart since 2017. His professional experience spans developing optimization and machine learning products in the retail domain. He holds a bachelors degree in electronics and communication engineering from the Indian Institute of Technology, Guwahati.

Naidu KVM was a senior principal data scientist with Flipkart from May 2018 to October 2024. His prior work experience includes Google, Amazon, Target, and Walmart. He holds an ME in computer science and automation from the Indian Institute of Science, Bangalore, and a BTech in computer science and engineering from the Indian Institute of Technology, Guwahati.

Tanu Modi is a product manager at Flipkart since 2022, building automation, increasing efficiency, and scaling systems across various planning products: demand, supply, and inventory planning. She holds a BTech in chemical engineering from the Indian Institute of Technology, Kanpur.

Ramkumar Moorthy is a product manager at Flipkart since 2015 and holds a BTech from Anna University, Chennai, and an MBA from the Indian Institute of Management Ahmedabad. He has spent the last 10 years building and scaling multiple consumer and software as a service/cloud products, some of which were first of their kind in the Indian e-commerce and supply chain space. Currently, he is leading the Flipkart retention products, Flipkart Plus and Flipkart Black, India's largest loyalty programs.

Rakesh S. Nair is a product manager and has been at Flipkart since 2022. He holds a bachelors degree in computer science from the National Institute of Technology Calicut and an MBA from the Indian Institute of Management Indore. He started as an research and development engineer and then turned to product management in which spent the past 15 years developing products across fintech, smart/green mobility and supply chain domains.

Goutham Sai Panyam is a data scientist and has been at Flipkart since 2021. His professional experience spans developing optimization, machine learning, and generative artificial intelligence products in the retail industry. He holds a bachelors degree in computer science from the Indian Institute of Technology, Kanpur.

Piyush Vyas is a senior data scientist and has been at Flipkart since 2020. His professional experience spans developing optimization and machine learning products in retail, banking, services, logistics, and high-frequency trading domains. He holds an integrated degree (BS + MS) in mathematics and scientific computing from the Indian Institute of Technology, Kanpur.